Chapter 8
Basic Cryptography

YORK: Then, York, be still awhile, till time do serve:
Watch thou and wake when others be asleep,
To pry into the secrets of the state;
—*The Second Part of King Henry the Sixth*, I, i, 249–260.

Cryptography is a deep mathematical subject. Because this book focuses on system security, we view cryptography as a supporting tool. Viewed in this context, the reader needs only a brief overview of the major points of cryptography relevant to that use. This chapter provides such an overview.

Cryptographic protocols provide a cornerstone for secure communication. These protocols are built on ideas presented in this chapter and are discussed at length later on.

8.1 What Is Cryptography?

The word *cryptography* comes from two Greek words meaning "secret writing" and is the art and science of concealing meaning. *Cryptanalysis* is the breaking of codes. The basic component of cryptography is a cryptosystem.

Definition 8–1. A *cryptosystem* is a 5-tuple $(\mathcal{E}, \mathcal{D}, \mathcal{M}, \mathcal{K}, C)$, where $\mathcal{M}$ is the set of *plaintexts*, $\mathcal{K}$ the set of *keys*, C is the set of *ciphertexts*, $\mathcal{E}$: $\mathcal{M} \times \mathcal{K} \rightarrow C$ is the set of *enciphering functions*, and $\mathcal{D}$: $C \times \mathcal{K} \rightarrow \mathcal{M}$ is the set of *deciphering functions*.

EXAMPLE: The Caesar cipher is the widely known cipher in which letters are shifted. For example, if the key is 3, the letter A becomes D, B becomes E, and so forth, ending with Z becoming C. So the word "HELLO" is enciphered as "KHOOR." Informally, this cipher is a cryptosystem with:

$\mathcal{M}$ = { all sequences of Roman letters }

$$\mathcal{K} = \{\, i \mid i \text{ an integer such that } 0 \le i \le 25 \,\}$$
$$\mathcal{E} = \{\, E_k \mid k \in \mathcal{K} \text{ and for all } m \in \mathcal{M},\ E_k(m) = (m + k) \bmod 26 \,\}$$

Representing each letter by its position in the alphabet (with A in position 0), "HELLO" is 7 4 11 11 14; if $k = 3$, the ciphertext is 10 7 14 14 17, or "KHOOR."

$$\mathcal{D} = \{\, D_k \mid k \in \mathcal{K} \text{ and for all } c \in \mathcal{C},\ D_k(c) = (26 + c - k) \bmod 26 \,\}$$

Each D_k simply inverts the corresponding E_k.

$$\mathcal{C} = \mathcal{M}$$

because $\mathcal{E}$ is clearly a set of onto functions.

The goal of cryptography is to keep enciphered information secret. Assume that an *adversary* wishes to break a ciphertext. Standard cryptographic practice is to assume that she knows the algorithm used to encipher the plaintext, but not the specific cryptographic key (in other words, she knows $\mathcal{D}$ and $\mathcal{E}$). She may use three types of attacks:

1. In a *ciphertext only* attack, the adversary has only the ciphertext. Her goal is to find the corresponding plaintext. If possible, she may try to find the key, too.
2. In a *known plaintext* attack, the adversary has the ciphertext and the plaintext that was enciphered. Her goal is to find the key that was used.
3. In a *chosen plaintext* attack, the adversary may ask that specific plaintexts be enciphered. She is given the corresponding ciphertexts. Her goal is to find the key that was used.

A good cryptosystem protects against all three types of attacks.

Attacks use both mathematics and statistics. The statistical methods make assumptions about the statistics of the plaintext language and examine the ciphertext to correlate its properties with those assumptions. Those assumptions are collectively called a *model* of the language. Figure 8–1 presents a character-based, or 1-gram, model of English text; others are 2-gram models (reflecting frequencies of pairs of letters), Markov models, and word models. In what follows, we use the 1-gram model and assume that the characters are chosen independently of one another.

8.2 Classical Cryptosystems

Classical cryptosystems (also called *single-key* or *symmetric* cryptosystems) are cryptosystems that use the same key for encipherment and decipherment. In these systems, for all $E_k \in \mathcal{C}$ and $k \in \mathcal{K}$, there is a $D_k \in \mathcal{D}$ such that $D_k = E_k^{-1}$.

a	0.080	h	0.060	n	0.070	t	0.090
b	0.015	i	0.065	o	0.080	u	0.030
c	0.030	j	0.005	p	0.020	v	0.010
d	0.040	k	0.005	q	0.002	w	0.015
e	0.130	l	0.035	r	0.065	x	0.005
f	0.020	m	0.030	s	0.060	y	0.020
g	0.015					z	0.002

Figure 8–1 Table of character frequencies in the English language, from Denning [242], Figure 2.3, p. 65.

EXAMPLE: The Caesar cipher discussed earlier had a key of 3, so the enciphering function was E_3. To decipher "KHOOR," we used the same key in the decipherment function D_3. Hence, the Caesar cipher is a classical cipher.

There are two basic types of classical ciphers: *transposition* ciphers and *substitution* ciphers.

8.2.1 Transposition Ciphers

A *transposition cipher* rearranges the characters in the plaintext to form the ciphertext. The letters are not changed.

EXAMPLE: The *rail fence* cipher is composed by writing the plaintext in two rows, proceeding down, then across, and reading the ciphertext across, then down. For example, the plaintext "HELLO, WORLD" would be written as:

```
HLOOL
ELWRD
```

resulting in the ciphertext "HLOOLELWRD."

Mathematically, the key to a transposition cipher is a permutation function. Because the permutation does not alter the frequency of plaintext characters, a transposition cipher can be detected by comparing character frequencies with a model of the language. If, for example, character frequencies for 1-grams match those of a model of English, but 2-gram frequencies do not match the model, then the text is probably a transposition cipher.

Attacking a transposition cipher requires rearrangement of the letters of the ciphertext. This process, called *anagramming*, uses tables of n-gram frequencies to identify common n-grams. The cryptanalyst arranges the letters in such a way that the

characters in the ciphertext form some n-grams with highest frequency. This process is repeated, using different n-grams, until the transposition pattern is found.

EXAMPLE: Consider the ciphertext "HLOOLELWRD." According to a Konheim's digram table [527], the digram "HE" occurs with frequency 0.0305[1] in English. Of the other possible digrams beginning with "H," the frequency of "HO" is the next highest, at 0.0043, and the digrams "HL," "HW," "HR," and "HD" have frequencies of less than 0.0010. Furthermore, the frequency of "WH" is 0.0026, and the digrams "EH," "LH," "OH," "RH," and "DH" occur with frequencies of 0.0002 or less. This suggests that "E" follows "H." We arrange the letters so that each letter in the first block of five letters (from "H" up to but not including the "E") is adjacent to the corresponding letter in the second block of five letters, as follows.

HE
LL
OW
OR
LD

Reading the letters across and down produces "HELLOWORLD." Note that the shape of the arrangement is different from that in the previous example. However, the two arrangements are equivalent, leading to the correct solution.

8.2.2 Substitution Ciphers

A *substitution cipher* changes characters in the plaintext to produce the ciphertext.

EXAMPLE: The Caesar cipher discussed earlier had a key of 3, altering each letter in the plaintext by mapping it into the letter three characters later in the alphabet (and circling back to the beginning of the alphabet if needed). This is a substitution cipher.

A Caesar cipher is susceptible to a statistical ciphertext-only attack.

EXAMPLE: Consider the ciphertext "KHOOR ZRUOG." We first compute the frequency of each letter in the ciphertext:

G 0.1 H 0.1 K 0.1 O 0.3 R 0.2 U 0.1 Z 0.1

[1] This means that in Konheim's sample, 3.05% of the digrams were "HE."

i	$\phi(i)$	i	$\phi(i)$	i	$\phi(i)$	i	$\phi(i)$
0	0.0482	7	0.0442	13	0.0520	19	0.0315
1	0.0364	8	0.0202	**14**	**0.0535**	20	0.0302
2	0.0410	9	0.0267	15	0.0226	21	0.0517
3	**0.0575**	**10**	**0.0635**	16	0.0322	22	0.0380
4	0.0252	11	0.0262	17	0.0392	23	0.0370
5	0.0190	12	0.0325	18	0.0299	24	0.0316
6	**0.0660**					25	0.0430

Figure 8–2 The value of $\phi(i)$ for $0 \le i \le 25$ using the model in Figure 8–1.

We now apply the character-based model. Let $\phi(i)$ be the correlation of the frequency of each letter in the ciphertext with the character frequencies in English (see Figure 8–1). Let $f(c)$ be the frequency of character c (expressed as a fraction). The formula for this correlation for this ciphertext (with all arithmetic being mod 26) is

$$\phi(i) = \Sigma_{0 \le c \le 25} f(c)p(c - i) = 0.1p(6 - i) + 0.1p(7 - i) + 0.1p(10 - i) + 0.3p(14 - i) + 0.2p(17 - i) + 0.1p(20 - i) + 0.1p(25 - i)$$

This correlation should be a maximum when the key k translates the ciphertext into English. Figure 8–2 shows the values of this function for the values of i. Trying the most likely key first, we obtain as plaintext "EBIIL TLOIA" when $i = 6$, "AXEEH PHKEW" when $i = 10$, "HELLO WORLD" when $i = 3$, and "WTAAD LDGAS" when $i = 14$.

The example above emphasizes the statistical nature of this attack. The statistics indicated that the key was most likely 6, when in fact the correct key was 3. So the attacker must test the results. The statistics simply reduce the number of trials in most cases. Only three trials were needed, as opposed to 13 (the expected number of trials if the keys were simply tried in order).

EXAMPLE: Using Konheim's model of single-character frequencies [527], the most likely keys (in order) are $i = 6$, $i = 10$, $i = 14$, and $i = 3$. Konheim's frequencies are different than Denning's, and this accounts for the change in the third most probable key.

8.2.2.1 Vigenère Cipher

A longer key might obscure the statistics. The Vigenère cipher chooses a sequence of keys, represented by a string. The key letters are applied to successive plaintext

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

Figure 8–3 The Vigenère tableau.

characters, and when the end of the key is reached, the key starts over. The length of the key is called the *period* of the cipher. Figure 8–3 shows a *tableau*, or table, to implement this cipher efficiently. Because this requires several different key letters, this type of cipher is called *polyalphabetic*.

EXAMPLE: The first line of a limerick is enciphered using the key "BENCH," as follows.

Key	B ENCHBENC HBENC HBENCH BENCHBENCH
Plaintext	A LIMERICK PACKS LAUGHS ANATOMICAL
Ciphertext	B PVOLSMPM WBGXU SBYTJZ BRNVVNMPCS

The *index of coincidence* measures the differences in the frequencies of the letters in the ciphertext. It is defined as the probability that two randomly chosen letters from the ciphertext will be the same. Let F_c be the frequency of cipher character c, and let N be the length of the ciphertext. It can be shown (see Exercise 7) that the index of coincidence IC is $IC = \frac{1}{N(N-1)} \sum_{i=0}^{25} F_i(F_i - 1)$. Figure 8–4 shows the expected values of IC for several periods. The lower the index of coincidence, the less variation in the characters of the ciphertext and (from our model of English) the longer the period of the cipher.

For many years, the Vigenère cipher was considered unbreakable. Then a Prussian cavalry officer named Kasiski noticed that repetitions occur when characters of the key appear over the same characters in the ciphertext. The number of characters between the repetitions is a multiple of the period.

EXAMPLE: Let the message be THE BOY HAS THE BAG and let the key be VIG. Then:

Key	VIGVIGVIGVIGVIG
Plaintext	THEBOYHASTHEBAG
Ciphertext	OPKWWECIYOPKWIM

In the ciphertext, the string OPK appears twice. Both are caused by the key sequence VIG enciphering the same ciphertext, THE. The ciphertext repetitions are nine characters apart. Hence, 9 is a multiple of the period (which is 3 here).

We examine the ciphertext for multiple repetitions and tabulate their length and the number of characters between successive repetitions. The period is likely to

Period	1	2	3	4	5	10	Large
Expected IC	0.066	0.052	0.047	0.045	0.044	0.041	0.038

Figure 8–4 Indices of coincidences for different periods. From Denning [242], Table 2.2, p. 78.

be a factor of the number of characters between these repetitions. From the repetitions, we establish the probable period, using the index of coincidence to check our deduction. We then tabulate the characters for each key letter separately and solve each as a Caesar cipher.

EXAMPLE: Consider the Vigenère cipher

```
ADQYS  MIUSB  OXKKT  MIBHK  IZOOO  EQOOG  IFBAG  KAUMF
VVTAA  CIDTW  MOCIO  EQOOG  BMBFV  ZGGWP  CIEKQ  HSNEW
VECNE  DLAAV  RWKXS  VNSVP  HCEUT  QOIOF  MEGJS  WTPCH
AJMOC  HIUIX
```

Could this be a Caesar cipher (which is a Vigenère cipher with a key length of 1)? We find that the index of coincidence is 0.043, which indicates a key of length 5 or more. So we assume that the key is of length greater than 1, and apply the Kasiski method. Repetitions of two letters or more are as follows.

Letters	Start	End	Gap length	Factors of gap length
MI	5	15	10	2, 5
OO	22	27	5	5
OEQOOG	24	54	30	2, 3, 5
FV	39	63	24	2, 2, 2, 3
AA	43	87	44	2, 2, 11
MOC	50	122	72	2, 2, 2, 3, 3
QO	56	105	49	7, 7
PC	69	117	48	2, 2, 2, 2, 3
NE	77	83	6	2, 3
SV	94	97	3	3
CH	118	124	6	2, 3

The longest repetition is six characters long; this is unlikely to be a coincidence. The gap between the repetitions is 30. The next longest repetition, MOC, is three characters long and has a gap of 72. The greatest common divisor of 30 and 72 is 6. Of the 11 repetitions, six have gaps with a factor of 6. The only factors that occur more in the gaps are 2 (in eight gaps) and 3 (in seven gaps). As a first guess, let us try 6.

To verify that this is reasonable, we compute the index of coincidence for each alphabet. We first arrange the message into six columns.

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

Each column represents one alphabet. The indices of coincidence are as follows.

Alphabet #1:	IC = 0.069	Alphabet #4:	IC = 0.056
Alphabet #2:	IC = 0.078	Alphabet #5:	IC = 0.124
Alphabet #3:	IC = 0.078	Alphabet #6:	IC = 0.043

All indices of coincidence indicate a single alphabet except for the ICs associated with alphabets #4 (period between 1 and 2) and #6 (period between 5 and 10). Given the statistical nature of the measure, we will assume that these are skewed by the distribution of characters and proceed on the assumption that there are six alphabets, and hence a key of length 6.

Counting characters in each column (alphabet) yields:

Column	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	3	1	0	0	4	0	1	1	3	0	1	0	0	1	3	0	0	1	1	2	0	0	0	0	0	0
#2	1	0	0	2	2	2	1	0	0	1	3	0	1	0	0	0	0	0	1	0	4	0	4	0	0	0
#3	1	2	0	0	0	0	0	0	2	0	1	1	4	0	0	0	4	0	1	3	0	2	1	0	0	0
#4	2	1	1	0	2	2	0	1	0	0	0	0	1	0	4	3	1	0	0	0	0	0	0	2	1	1
#5	1	0	5	0	0	0	2	1	2	0	0	0	0	0	5	0	0	0	3	0	0	2	0	0	0	0
#6	0	1	1	1	0	0	2	2	3	1	1	0	1	2	1	0	0	0	0	0	0	3	0	1	0	1

An unshifted alphabet has the following characteristics (L meaning low frequency, M meaning moderate frequency, and H meaning high frequency).

H M M M H M M H H M M M M H H M L H H H M L L L L L

We now compare the frequency counts in the six alphabets above with the frequency count of the unshifted alphabet. The first alphabet matches the characteristics of the unshifted alphabet (note the values for A, E, and I in particular). Given the gap between B and I, the third alphabet seems to be shifted with I mapping to A. A similar gap occurs in the sixth alphabet between O and V, suggesting that V maps to A. Substituting into the ciphertext (bold letters are plaintext) produces

ADI**Y**S **RIUK**B **O**C**K**K**L** MI**GH**K **A**ZO**TO** **EIOOL** **IF**T**A**G **PAUE**F
VATAS CI**I**T**W** **EOCNO** **EIOOL** **BMTF**V **EGGOP** C**NE**K**I** HS**SE**W
NECSE **D**D**AAA** **RW**C**X**S **ANSN**P **HHEUL** **QONOF** **EEGOS** **WLPCM**
AJEOC **MIUA**X

In the last line, the group **AJE** suggests the word **ARE**. Taking this as a hypothesis, the second alphabet maps A into S. Substituting back produces

ALI**Y**S **RICK**B **O**C**KSL** MI**GHS** **A**ZO**TO** **MIOOL** **INTAG** **PACE**F
VATIS CI**I**T**E** **EOCNO** **MIOOL** **BUTF**V **EGOOP** C**NESI** HS**SEE**
NECSE **LDAAA** **RECX**S **ANANP** **HHECL** **QONON** **EEGOS** **ELPCM**
AREOC **MICA**X

The last block suggests **MICAL**, because AL is a common ending for adjectives. This means that the fourth alphabet maps O into A, and the cipher becomes

```
ALIMS  RICKP  OCKSL  AIGHS  ANOTO  MICOL  INTOG  PACET
VATIS  QIITE  ECCNO  MICOL  BUTTV  EGOOD  CNESI  VSSEE
NSCSE  LDOAA  RECLS  ANAND  HHECL  EONON  ESGOS  ELDCM
ARECC  MICAL
```

In English, a Q is always followed by a U, so the I in the second group of the second line must map to U. The fifth alphabet maps M to A. The cipher is solved:

```
ALIME  RICKP  ACKSL  AUGHS  ANATO  MICAL  INTOS  PACET
HATIS  QUITE  ECONO  MICAL  BUTTH  EGOOD  ONESI  VESEE
NSOSE  LDOMA  RECLE  ANAND  THECL  EANON  ESSOS  ELDOM
ARECO  MICAL
```

With proper spacing and punctuation, we have

```
A LIMERICK PACKS LAUGHS ANATOMICAL
INTO SPACE THAT IS QUITE ECONOMICAL
   BUT THE GOOD ONES I'VE SEEN
   SO SELDOM ARE CLEAN,
AND THE CLEAN ONES SO SELDOM ARE COMICAL.
```

The key is ASIMOV.

It is worth noting that the Vigenère cipher is easy to break by hand. However, the principles of attack hold for more complex ciphers that can be implemented only by computer. A good example is the encipherments that several older versions of WordPerfect used [75, 78]. These allowed a user to encipher a file with a password. Unfortunately, certain fields in the enciphered file contained information internal to WordPerfect, and these fields could be predicted. This allowed an attacker to derive the password used to encipher the file, and from that the plaintext file itself.

8.2.2.2 One-Time Pad

The one-time pad is a variant of the Vigenère cipher. The technique is the same. The key string is chosen at random, and is at least as long as the message, so it does not repeat. Technically, it is a threshold scheme [815], and is provably impossible to break [115]. The implementation issues of the pad, including random generation of the key and key distribution, do not concern us here (although a later chapter will touch on them).

8.2.3 Data Encryption Standard

The Data Encryption Standard (DES) [662] was designed to encipher sensitive but nonclassified data. It is bit-oriented, unlike the other ciphers we have seen. It uses both transposition and substitution and for that reason is sometimes referred to as a *product cipher*. Its input, output, and key are each 64 bits long. The sets of 64 bits are referred to as *blocks*.

The cipher consists of 16 *rounds*, or iterations. Each round uses a separate key of 48 bits. These *round keys* are generated from the key block by dropping the parity bits (reducing the effective key size to 56 bits), permuting the bits, and extracting 48 bits. A different set of 48 bits is extracted for each of the 16 rounds (see Figure 8–5). If the order in which the round keys is used is reversed, the input is deciphered.

The rounds are executed sequentially, the input of one round being the output of the previous round. The right half of the input, and the round key, are run through a function f that produces 32 bits of output; that output is then xor'ed into the left half, and the resulting left and right halves are swapped (see Figure 8–6).

The function f provides the strength of the DES. The right half of the input (32 bits) is expanded to 48 bits, and this is xor'ed with the round key. The resulting 48 bits are split into eight sets of six bits each, and each set is put through a substitution

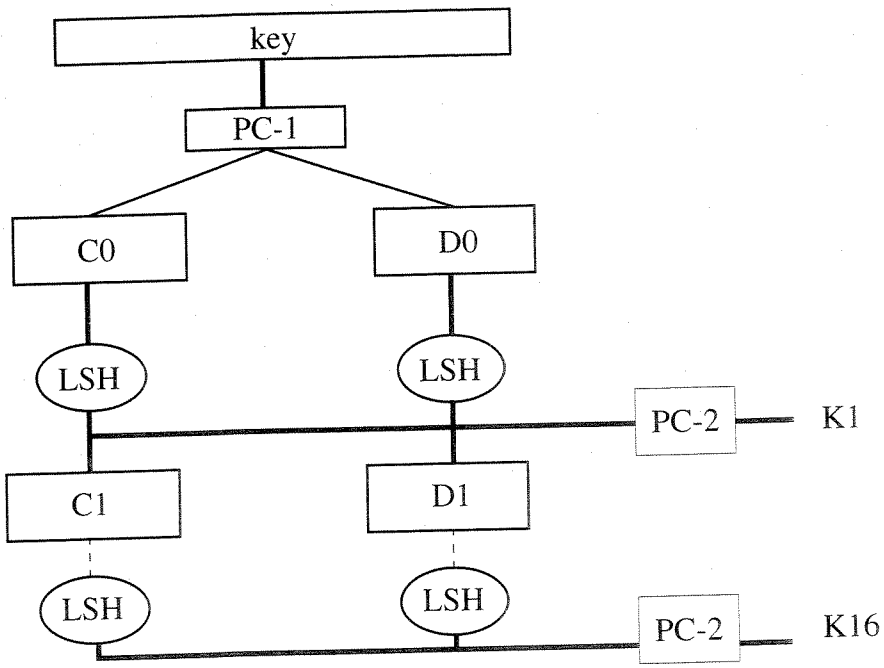

Figure 8–5 DES key schedule generation. PC-1 and PC-2 are permutation tables; LSH is a table of left shifts (rotations).

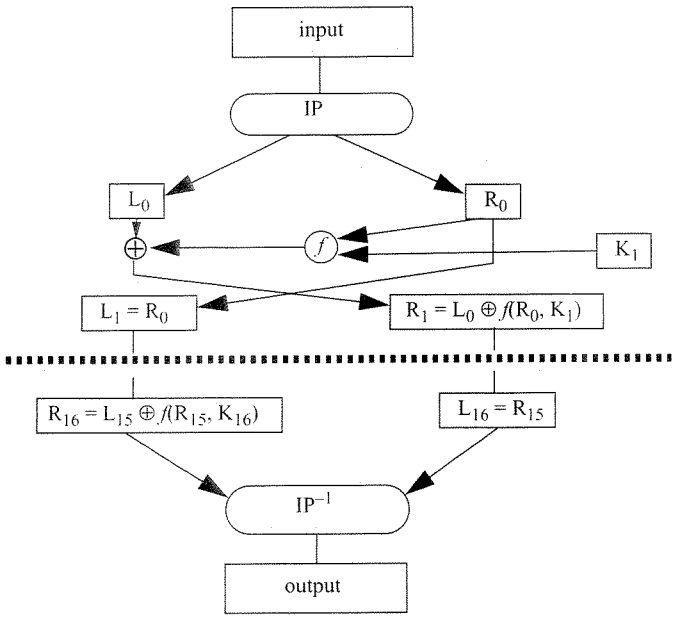

Figure 8–6 DES message encipherment and decipherment.

table called the *S-box*. Each S-box produces four bits of output. They are catenated into a single 32-bit quantity, which is permuted. The resulting 32 bits constitute the output of the f function (see Figure 8–7).

When the DES was first announced, it was criticized as too weak. First, Diffie and Hellman [268] argued that a key length of 56 bits was simply too short, and they designed a machine that could break a DES-enciphered message in a matter of days. Although their machine was beyond the technology of the time, they estimated that it could soon be built for about $20,000,000. Second, the reasons for many of the decisions in the design of the DES—most notably, those involving the S-boxes—were classified. Many speculated that the classification hid "trapdoors," or ways to invert the cipher without knowing the key.

Some properties of the DES were worrisome. First, it had four weak keys (keys that were their own inverses) and 12 semiweak keys (keys whose inverses were other keys). Second, let $\overline{k}$, $\overline{m}$, and $\overline{c}$ be the complement of the key k, the plaintext m, and the ciphertext c, respectively. Let $\mathrm{DES}_k(m)$ be the encipherment of plaintext m under key k. Then the *complementation property* states that

$$\mathrm{DES}_k(m) = c \Rightarrow \mathrm{DES}_{\overline{k}}(\overline{m}) = \overline{c}$$

Third, some of the S-boxes exhibited irregular properties. The distribution of odd and even numbers was nonrandom, raising concerns that the DES did not randomize the input sufficiently. Several output bits of the fourth S-box seemed to depend on some

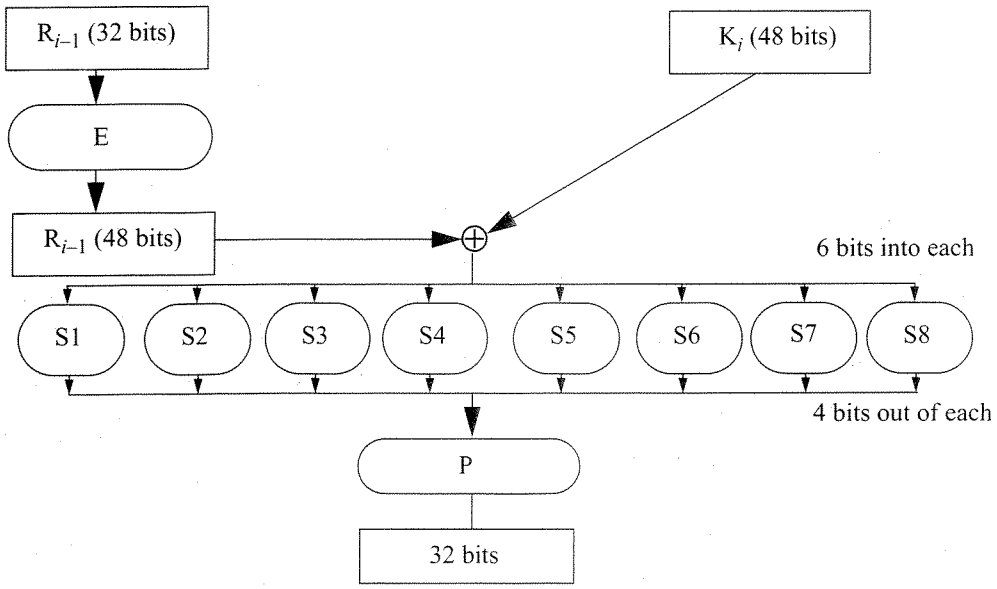

Figure 8–7 The *f* function.

of the output bits of the third S-box. This again suggested that there was a structure to the S-boxes, and because some of the design decisions underlying the S-boxes were unknown, the reasons for the structure were unknown. The structure made hardware implementation of the DES simpler [907]. It distributed the dependence of each output bit on each input bit rapidly, so that after five rounds each output bit depended on every key and input bit [625]. It could have been needed to prevent the cipher from being broken easily. It also could enable a trapdoor to allow the cipher to be broken easily. There was considerable speculation that the NSA had weakened the algorithm, although a congressional investigation did not reflect this [59].

In 1990, a breakthrough in cryptanalysis answered many of these questions. Biham and Shamir applied a technique called *differential cryptanalysis* to the DES [90, 91, 92]. This technique required them to generate 2^{47} pairs of chosen plaintext and ciphertext, considerably fewer than the trial-and-error approach others had used. During the development of this technique, they found several properties of the DES that appeared to answer some of the questions that had been raised.

First, for a known plaintext attack, differential cryptanalysis requires 2^{56} plaintext and ciphertext pairs for a 15-round version of the DES. For the full 16 rounds, 2^{58} known plaintext and ciphertext pairs are needed, which is more than sufficient for a trial-and-error approach. (Matsui subsequently improved this using a variant attack called linear cryptanalysis [596]; this attack requires 2^{43} known plaintext and ciphertext pairs on the average.) Second, small changes in the S-boxes weakened the cipher (so that the required number of chosen plaintext and ciphertext pairs was reduced). Third, making every bit of the round keys independent (for an

effective key length of $16 \times 48 = 768$ bits) did not make the DES resistant to differential cryptanalysis, which suggests that the designers of the DES knew about differential analysis. Coppersmith later confirmed this [209].

The DES is used in several modes [663]. Using it directly is called electronic code book (ECB) mode, and is very rare. Modes in which it can be used to generate a pseudo-one-time pad are cipher feed back (CFB) mode (see Section 10.2.1.2) and output feed back (OFB) mode (see Section 10.2.1.1). Its most common modes of use are cipher block chaining (CBC) mode (see Section 10.2.2), encrypt-decrypt-encrypt (EDE) mode, and triple DES mode (the EDE and triple DES modes are described in Section 10.2.2.1).

The CBC mode is an iterative mode in which a block of ciphertext depends not only on its input but also on the preceding ciphertext block. In addition to a 64-bit key, it requires a 64-bit initialization vector. Figure 8–8 shows this mode. It has the *self-healing property*. This property says that if one block of ciphertext is altered, the error propagates for at most two blocks. Figure 8–9 shows how a corrupted block affects others.

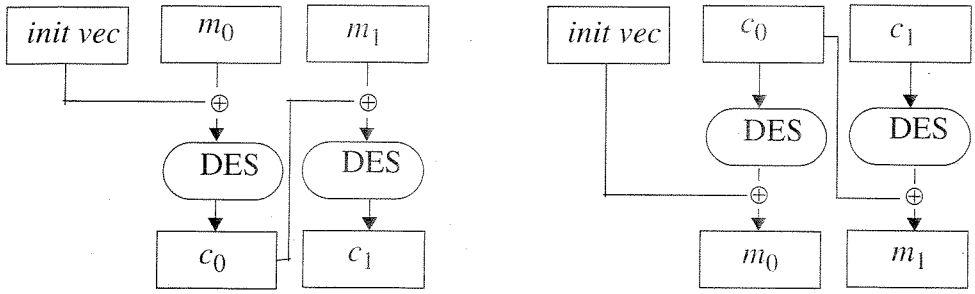

Figure 8–8 Cipher block chaining mode. The left diagram shows encipherment; each ciphertext is "fed back" into the cipher stream. The right diagram shows decipherment.

Incorrect ciphertext:	ef7c4cb2b4ce6f3b	f6266e3a97af0e2c
	746ab9a6308f4256	33e60b451b09603d
Corresponding plaintext:	*efca61e19f4836f1*	3231333336353837
	3231343336353837	3231343336353837
The real plaintext:	3231343336353837	3231343336353837
	3231343336353837	3231343336353837

Figure 8–9 Example of the self-healing property. The ciphertext at the top was stored incorrectly (the italicized 4c should be 4b). Its decipherment is shown next, with the incorrect octets italicized. The plaintext used to create the ciphertext is shown at the bottom.

The EDE mode is used by many financial institutions. It requires two 64-bit keys k and k'. The ciphertext c corresponding to some data m is $c = DES_k(DES_{k'}^{-1}(DES_k(m)))$. Triple DES uses three keys k, k', and k'', and the second step is an encipherment, not a decipherment: $c = DES_k(DES_{k'}(DES_{k''}(m)))$.

In 1998, a design for a computer system and software that could break any DES-enciphered message in a few days was published [358]. This design complemented several challenges to break specific DES messages. Those challenges had been solved using computers distributed throughout the Internet. By 1999, it was clear that the DES no longer provided the same level of security as it had 10 years earlier, and the search was on for a new, stronger cipher (to be called the Advanced Encryption Standard, or AES) to fill the needs that the DES no longer filled.

The DES is one of the most important classical cryptosystems in the history of cryptography. It provided the impetus for many advances in the field and laid the theoretical and practical groundwork for many other ciphers. While analyzing it, researchers developed differential and linear cryptanalysis. Cryptographers developed other ciphers to avoid real, or perceived, weaknesses; cryptanalysts broke many of these ciphers and found weaknesses in others. Many of the features of the DES are used in other ciphers. Hence, even though it is nearing the end of its useful lifetime, it is well worth understanding.

In late 2001, the National Institute of Standards and Technology announced the selection of Rijndael as the Advanced Encryption Standard [672], the successor to the DES. Like the DES, the AES is a product cipher. Unlike the DES, the AES can use keys of 128, 192, or 256 bits and operates on blocks of 128 bits. It was specifically designed to withstand the attacks to which the DES showed weaknesses [228]. Time will show how rapidly it supplants the DES, but the lessons learned from the DES have borne fruit.

8.2.4 Other Classical Ciphers

Several algorithms have been proposed to overcome the weaknesses in the DES. NewDES (which, despite its name, is not a variant of DES but a new algorithm) has a block size of 64 bits and a key length of 120 bits [803]. However, it can be broken using an attack similar to differential cryptanalysis [796]. FEAL is another block cipher, with a block size of 64 bits and a key size of 64 bits [642, 822]. FEAL-4 (FEAL with four rounds) and FEAL-8 (FEAL with eight rounds) fell to differential cryptanalysis with 20 [658] and 10,000 [357] chosen plaintexts, respectively. Biham and Shamir broke FEAL-N, which uses N rounds, for $N < 32$ by differential cryptanalysis more quickly than by trial-and-error [91]. It was proposed that the key be lengthened to 128 bits, but the 128-bit key proved as easy to break as FEAL-N with the original 64-bit key. REDOC-II [226] has an 80-bit block and a 160-bit key. It has 10 rounds, and although a single round was successfully cryptanalyzed [89], the use of 10 rounds appears to withstand differential cryptanalysis.

LOKI89 [137], proposed as an alternative to the DES, was vulnerable to differential cryptanalysis [89]. Its successor, LOKI91 [138], uses a 64-bit key and a 64-bit block size. Differential cryptanalysis fails to break this cipher [516]. Khufu [623] has a block size of 64 bits and a key size of 512 bits. When used with 24 or 32 rounds, it resists chosen plaintext attacks. Its S-boxes are computed from the keys. Khafre [623], similar in design to Khufu, uses fixed S-boxes, but it has been broken [89].

IDEA is an eight-round cipher that uses 64-bit blocks and 128-bit keys [541]. It uses three operations: exclusive or's, addition modulo 2^{16}, and multiplication modulo $2^{16} + 1$. It appears to withstand known attacks but is too new for any definitive statement to be made about its security [796]. It is used in noncommercial software—notably, in the electronic mail program PGP [965]—but is patented and requires licensing for use in commercial software.

8.3 Public Key Cryptography

In 1976, Diffie and Hellman [267] proposed a new type of cryptography that distinguished between encipherment and decipherment keys.[2] One of the keys would be publicly known; the other would be kept private by its owner. Classical cryptography requires the sender and recipient to share a common key. Public key cryptography does not. If the encipherment key is public, to send a secret message simply encipher the message with the recipient's public key. Then send it. The recipient can decipher it using his private key. (Chapter 9, "Key Management," discusses how to make public keys available to others.)

Because one key is public, and its complementary key must remain secret, a public key cryptosystem must meet the following three conditions.

1. It must be computationally easy to encipher or decipher a message given the appropriate key.
2. It must be computationally infeasible to derive the private key from the public key.
3. It must be computationally infeasible to determine the private key from a chosen plaintext attack.

The RSA cipher provides both secrecy and authentication.

[2] James Ellis, a cryptographer working for the British government's Communications-Electronics Security Group, said "he showed proof of concept in a January 1970 CESG report titled 'The Possibility of Secure Non-Secret Digital Encryption.'" Two of his colleagues found practical implementations. This work remained classified until 1997 ([244], p. 299).

8.3.1 RSA

RSA [756] is an exponentiation cipher. Choose two large prime numbers p and q, and let $n = pq$. The *totient* $\phi(n)$ of n is the number of numbers less than n with no factors in common with n.[3]

EXAMPLE: Let $n = 10$. The numbers that are less than 10 and are relatively prime to (have no factors in common with) n are 1, 3, 7, and 9. Hence, $\phi(10) = 4$. Similarly, if $n = 21$, the numbers that are relatively prime to n are 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, and 20. So $\phi(21) = 12$.

Choose an integer $e < n$ that is relatively prime to $\phi(n)$. Find a second integer d such that $ed \bmod \phi(n) = 1$. The public key is (e, n), and the private key is d.

Let m be a message. Then:

$$c = m^e \bmod n$$

and

$$m = c^d \bmod n$$

EXAMPLE: Let $p = 7$ and $q = 11$. Then $n = 77$ and $\phi(n) = 60$. Alice chooses $e = 17$, so her private key is $d = 53$. In this cryptosystem, each plaintext character is represented by a number between 00 (A) and 25 (Z); 26 represents a blank. Bob wants to send Alice the message "HELLO WORLD." Using the representation above, the plaintext is 07 04 11 11 14 26 22 14 17 11 03. Using Alice's public key, the ciphertext is

$$07^{17} \bmod 77 = 28$$
$$04^{17} \bmod 77 = 16$$
$$11^{17} \bmod 77 = 44$$
$$\ldots$$
$$03^{17} \bmod 77 = 75$$

or 28 16 44 44 42 38 22 42 19 44 75.

In addition to confidentiality, RSA can provide data and origin authentication. If Alice enciphers her message using her private key, anyone can read it, but if anyone alters it, the (altered) ciphertext cannot be deciphered correctly.

[3] Our examples will use small numbers for pedagogical purposes. Actual RSA primes should be at least 512 bits each, giving a modulus of at least 1,024 bits. In practice, RSA is combined with cryptographic hash functions to prevent rearrangement of blocks (see Section 10.1.2).

EXAMPLE: Suppose Alice wishes to send Bob the message "HELLO WORLD" in such a way that Bob will be sure that Alice sent it. She enciphers the message with her private key and sends it to Bob. As indicated above, the plaintext is represented as 07 04 11 11 14 26 22 14 17 11 03. Using Alice's private key, the ciphertext is

$$07^{53} \bmod 77 = 35$$
$$04^{53} \bmod 77 = 09$$
$$11^{53} \bmod 77 = 44$$
$$\ldots$$
$$03^{53} \bmod 77 = 05$$

or 35 09 44 44 93 12 24 94 04 05. In addition to origin authenticity, Bob can be sure that no letters were altered.

Providing both confidentiality and authentication requires enciphering with the sender's private key and the recipient's public key.

EXAMPLE: Suppose Alice wishes to send Bob the message "HELLO WORLD" in confidence and authenticated. Again, assume that Alice's private key is 53. Take Bob's public key to be 37 (making his private key 13). The plaintext is represented as 07 04 11 11 14 26 22 14 17 11 03. The encipherment is

$$(07^{53} \bmod 77)^{37} \bmod 77 = 07$$
$$(04^{53} \bmod 77)^{37} \bmod 77 = 37$$
$$(11^{53} \bmod 77)^{37} \bmod 77 = 44$$
$$\ldots$$
$$(03^{53} \bmod 77)^{37} \bmod 77 = 47$$

or 07 37 44 44 14 59 22 14 61 44 47.

The recipient uses the recipient's private key to decipher the message and the sender's public key to authenticate it.

EXAMPLE: Bob receives the ciphertext above, 07 37 44 44 14 59 22 14 61 44 47. The decipherment is

$$(07^{13} \bmod 77)^{17} \bmod 77 = 07$$
$$(37^{13} \bmod 77)^{17} \bmod 77 = 04$$
$$(44^{13} \bmod 77)^{17} \bmod 77 = 11$$
$$\ldots$$
$$(47^{13} \bmod 77)^{17} \bmod 77 = 03$$

or 07 04 11 11 14 26 22 14 17 11 03. This corresponds to the message "HELLO WORLD" from the preceding example.

The use of a public key system provides a technical type of nonrepudiation of origin. The message is deciphered using Alice's public key. Because the public key is the inverse of the private key, only the private key could have enciphered the message. Because Alice is the only one who knows this private key, only she could have enciphered the message. The underlying assumption is that Alice's private key has not been compromised, and that the public key bearing her name really does belong to her.

In practice, no one would use blocks of the size presented here. The issue is that, even if n is very large, if one character per block is enciphered, RSA can be broken using the techniques used to break classical substitution ciphers (see Sections 8.2.2 and 10.1.3). Furthermore, although no individual block can be altered without detection (because the attacker presumably does not have access to the private key), an attacker can rearrange blocks and change the meaning of the message.

EXAMPLE: A general sends a message to headquarters asking if the attack is on. Headquarters replies with the message "ON" enciphered using an RSA cipher with a 1,024-bit modulus, but each letter is enciphered separately. An attacker intercepts the message and swaps the order of the blocks. When the general deciphers the message, it will read "NO," the opposite of the original plaintext.

Moreover, if the attacker knows that headquarters will send one of two messages (here, "NO" or "ON"), the attacker can use a technique called "forward search" or "precomputation" to break the cipher (see Section 10.1.1). For this reason, plaintext is usually padded with random data to make up a block. This can eliminate the problem of forward searching, because the set of possible plaintexts becomes too large to precompute feasibly.

A different general sends the same request as in the example above. Again, headquarters replies with the message "ON" enciphered using an RSA cipher with a 1,024-bit modulus. Each letter is enciphered separately, but the first six bits of each block contain the number of the block, the next eight bits contain the character, and the remaining 1,010 bits contain random data. If the attacker rearranges the blocks, the general will detect that block 2 arrived before block 1 (as a result of the number in the first six bits) and rearrange them. The attacker also cannot precompute the blocks to determine which contains "O," because she would have to compute 2^{1010} blocks, which is computationally infeasible.

8.4 Cryptographic Checksums

Alice wants to send Bob a message of n bits. She wants Bob to be able to verify that the message he receives is the same one that was sent. So she applies a mathematical function, called a checksum function, to generate a smaller set of k bits from the original n bits. This smaller set is called the *checksum* or *message digest*. Alice then sends Bob both the message and the checksum. When Bob gets the message, he

recomputes the checksum and compares it with the one Alice sent. If they match, he assumes that the message has not been changed.

EXAMPLE: The parity bit in the ASCII representation is often used as a single-bit checksum. If *odd parity* is used, the sum of the 1-bits in the ASCII representation of the character, and the parity bit, is odd. Assume that Alice sends Bob the letter "A." In ASCII, the representation of "A" using odd parity is $p0111101$ in binary, where p represents the parity bit. Because five bits are set, the parity bit is 0 for odd parity.

When Bob gets the message 00111101, he counts the 1-bits in the message. Because this number is odd, Bob knows that the message has arrived unchanged.

Definition 8–2. A *cryptographic checksum function* (also called a *strong hash function* or a *strong one-way function*) $h: A \rightarrow B$ is a function that has the following properties.

1. For any $x \in A$, $h(x)$ is easy to compute.
2. For any $y \in B$, it is computationally infeasible to find $x \in A$ such that $h(x) = y$.
3. It is computationally infeasible to find $x, x' \in A$, such that $x \neq x'$ and $h(x) = h(x')$. (Such a pair is called a *collision*.)

The third requirement is often stated as:

4. Given any $x \in A$, it is computationally infeasible to find another $x' \in A$ such that $x \neq x'$ and $h(x') = h(x)$.

However, properties 3 and 4 are subtlely different. It is considerably harder to find an x' meeting the conditions in property 4 than it is to find a pair x and x' meeting the conditions in property 3. To explain why, we need to examine some basics of cryptographic checksum functions.

Given that the checksum contains fewer bits than the message, several messages must produce the same checksum. The best checksum functions have the same number of messages produce each checksum. Furthermore, given any message, the checksum it produces can be determined only by computing the checksum. Such a checksum function acts as a random function.

The size of the output of the cryptographic checksum is an important consideration owing to a mathematical principle called the *pigeonhole principle*.

Definition 8–3. The *pigeonhole principle* states that if there are n containers for $n + 1$ objects, at least one container will hold two objects. To understand its application here, consider a cryptographic checksum function that computes hashes of three bits and a set of files each of which contains five bits. This yields $2^3 = 8$ possible hashes for $2^5 = 32$ files. Hence, at least four different files correspond to the same hash.

Now assume that a cryptographic checksum function computes hashes of 128 bits. The probability of finding a message corresponding to a given hash is 2^{-128}, but the probability of finding two messages with the same hash (that is, with the value of neither message being constrained) is 2^{-64} (see Exercise 20).

> **Definition 8–4.** A *keyed* cryptographic checksum function requires a cryptographic key as part of the computation. A *keyless* cryptographic checksum does not.

EXAMPLE: The DES in CBC mode can be used as a message authentication code if 64 bits or fewer are required. The message is enciphered, and the last n bits of the last output are the cryptographic hash. Because the DES requires a cryptographic key, this checksum function (called DES-MAC) is a keyed cryptographic checksum function. Because the DES is vulnerable to attack, so is this checksum technique. Furthermore, because the hash is at most 64 bits, finding two inputs that produce the same output would require 2^{32} messages.

Examples of keyless hash functions include MD2 [489]; MD4 [753]; MD5 [754]; the Secure Hash Algorithm (SHA-1) which produces 160-bit checksums [664, 663]; Snefru (either 128-bit or 256-bit checksums) [622]; and HAVAL, which produces checksums of 128, 160, 192, 224, and 256 bits [963]. Of these, Snefru is vulnerable to differential cryptanalysis if four rounds or fewer are used [92], so Merkle recommends using at least eight passes. Dobbertin devised a method of generating collisions in MD4 [274]; a similar method also works against MD5 but is slower [273].

8.4.1 HMAC

HMAC is a generic term for an algorithm that uses a keyless hash function and a cryptographic key to produce a keyed hash function [531]. This mechanism enables Alice to validate that data Bob sent to her is unchanged in transit. Without the key, anyone could change the data and recompute the message authentication code, and Alice would be none the wiser.

The need for HMAC arose because keyed hash functions are derived from cryptographic algorithms. Many countries restrict the import and export of software that implements such algorithms. They do not restrict software implementing keyless hash functions, because such functions cannot be used to conceal information. Hence, HMAC builds on a keyless hash function using a cryptographic key to create a keyed hash function.

Let h be a keyless hash function that hashes data in blocks of b bytes to produce a hash l bytes long. Let k be a cryptographic key. We assume that the length of k is no greater than b; if it is, use h to hash it to produce a new key of length b. Let k' be the key k padded with bytes containing 0 to make b bytes. Let *ipad* be a sequence of bytes containing the bits 00110110 and repeated b times; let *opad* be a similar sequence with the bits 01011100. The HMAC-h function with key k for message m is

$$\text{HMAC-}h(k, m) = h(k' \oplus opad \,||\, h(k' \oplus ipad \,||\, m))$$

where $\oplus$ is exclusive or and $||$ is concatenation.

Bellare, Canetti, and Krawczyk [65] analyze the security of HMAC and conclude that the strength of HMAC depends on the strength of the hash function h. Various HMAC functions are used in Internet security protocols (see Chapter 10).

8.5 Summary

For our purposes, three aspects of cryptography require study. Classical cryptography uses a single key shared by all involved. Public key cryptography uses two keys, one shared and the other private. Both types of cryptosystems can provide secrecy and origin authentication (although classical cryptography requires a trusted third party to provide both). Cryptographic hash functions may or may not use a secret key and provide data authentication.

All cryptosystems are based on substitution (of some quantity for another) and permutation (scrambling of some quantity). Cryptanalysis, the breaking of ciphers, uses statistical approaches (such as the Kasiski method and differential cryptanalysis) and mathematical approaches (such as attacks on the RSA method). As techniques of cryptanalysis improve, our understanding of encipherment methods also improves and ciphers become harder to break. The same holds for cryptographic checksum functions. However, as computing power increases, key length must also increase. A 56-bit key was deemed secure by many in 1976; it is clearly not secure now.

8.6 Further Reading

Cryptography is a vast, rich subject. Kahn's book *The Codebreakers* [482, 485] is required reading for anyone interested in this field. Kahn has written other excellent historical books on codebreaking during World War II [483, 484]. Helen Fouché Gaines presents techniques for cryptanalysis of many classical ciphers using traditional, pencil-and-paper analysis [343]. Sinkov applies basic mathematics to many of these classical ciphers [836]. Schneier describes many old, and new, algorithms in a clear, easy-to-understand manner [796]; his book is excellent for implementers. The underpinnings of these algorithms, and others, lie in statistics and mathematics. For classical cryptography, Konheim's book [527] is superb once the reader has mastered his notation. Unlike other books, it focuses on cryptanalysis of classical ciphers using statistical attacks. Meyer and Matyas [626] and Biham and Shamir [92] discuss the strengths and weaknesses of the DES. Seberry and Pieprzyk [805] and Simmons [834] discuss modern cryptography and its applications. Koblitz [521], Coutinho [215], and Goldreich [365] discuss modern mathematics, cryptographic theory, and