i	$\varphi(i)$	*i*	$\varphi(i)$	*i*	$\varphi(i)$	*i*	$\varphi(i)$
0	0.0469	7	0.0461	13	0.0505	19	0.0312
1	0.0393	8	0.0194	14	0.0561	20	0.0287
2	0.0396	9	0.0286	15	0.0215	21	0.0526
3	0.0586	10	0.0631	16	0.0306	22	0.0398
4	0.0259	11	0.0280	17	0.0386	23	0.0338
5	0.0165	12	0.0318	18	0.0317	24	0.0320
6	0.0676					25	0.0443

Figure 10–2 The value of $\varphi(i)$ for $0 \leq i \leq 25$ using the model in Figure 10–1.

EXAMPLE: Using Konheim's model of single-character frequencies [1092, p. 16], 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.

A variant of the shift cipher, called an *affine cipher*, uses a multiplier in addition to the shift. Exercise 4 examines this cipher.

10.2.2.1 Vigenère Cipher

The shift cipher maps every character into another character in one alphabet. Such a cipher is a *monoalphabetic* cipher. As noted above, it preserves the statistics of the underlying message, which a cryptanalyst can use to decipher the message.

A *polyalphabetic* cipher uses multiple alphabets to generate the ciphertest, thereby obscuring the statistics. The Vigenère cipher is such a cryptosystem. In it, the key is a sequence of letters. The key letters are applied to successive plaintext 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 10–3 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
```

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

	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 10–3 The Vigenère tableau.

number of characters between the repetitions is a multiple of the period. From this observation, he developed an effective attack.

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 "OPKW" appears twice. Both are caused by the key sequence "VIGV" enciphering the same ciphertext, "THEB." The ciphertext repetitions are nine characters apart. As Figure 10–4 shows, the lower this value, the less variation in the characters of the ciphertext and, from our models of English, the longer the period of the cipher.

Period	Expected IC	Period	Expected IC	Period	Expected IC
1	0.0660	7	0.0420	50	0.0386
2	0.0520	8	0.0415	60	0.0385
3	0.0473	9	0.0411	70	0.0384
4	0.0450	10	0.0408	80	0.0384
5	0.0436	20	0.3940	90	0.0383
6	0.0427	30	0.0389	99	0.0383
		40	0.0387		

Figure 10–4 Indices of coincidences for different periods.

The first step in the Kasiski method is to determine the length of the key. The *index of coincidence* (IC) measures the differences in the frequencies of the letters in the ciphertext. It is defined as the probability that two letters randomly chosen from the ciphertext will be the same. The lower this value, the less variation in the characters of the ciphertext and, from our models of English, the longer the period of the cipher.

Let F_c be the frequency of cipher character c, and let N be the length of the ciphertext. Then the index of coincidence IC can be shown to be (see Exercise 6)

$$IC = \frac{1}{N(N-1)} \sum_{i=0}^{25} F_i(F_i - 1)$$

We examine the ciphertext for multiple repetitions and tabulate their length and the number of characters between successive repetitions. The period is likely to 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 shift 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 shift cipher (which is a Vigenère cipher with a key length of 1)? We find that the index of coincidence is 0.0433, which indicates a key of around length 5. So we assume that the key is of length greater than 1, and apply the Kasiski method. Repetitions of length 2 are likely coincidental, so we look for repetitions of length 3 or more:

Letters	**Start**	**End**	**Gap length**	**Gap length factors**
OEQOOG	24	54	30	2, 3, 5
MOC	50	122	72	2, 2, 2, 3, 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. So 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 rows, one for each alphabet:

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

We then compute the indices of coincidence for these alphabets:

Alphabet #1: IC = 0.0692	Alphabet #4: IC = 0.0562
Alphabet #2: IC = 0.0779	Alphabet #5: IC = 0.1238
Alphabet #3: IC = 0.0779	Alphabet #6: IC = 0.0429

All indices of coincidence indicate a single alphabet except for the indices of coincidence associated with alphabets #4 (period between 1 and 2) and #6 (period between 5 and 6). 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 6 alphabets, and hence a key of length 6.

Counting characters in each column (alphabet) yields

Row	*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 (lowercase letters are plaintext) produces

```
aDiYS riUkB OckKl MIghK aZOto EiOOl iFtAG paUeF
VatAs CIitW eOCno EiOOl bMtFV egGoP CneKi HSseW
nECse DdAAa rWcXS anSnP HheUl QOnoF eEGos WlPCm
aJeOC miUaX
```

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

```
aliYS rickB Ocksl MIghs aZOto miOOl intAG paceF
Vatis CIite eOCno miOOl butFV egooP Cnesi HSsee
nECse ldAAa recXS ananP Hhecl QOnon eEGos elPCm
areOC micaX
```

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, capitalization, 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."

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 [171, 173]. 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.

10.2.2.2 One-Time Pad

Repetitions provide a means for the cryptanalyst to attack the Vigènere cipher. The *one-time pad* is a variant of the Vigenère cipher with a key that is at least as long as the message and is chosen at random, so it does not repeat. Technically, it is a threshold scheme (see Section 16.3.2), and is provably impossible to break [240] (see also Section C.3.3, "Perfect Secrecy").

The weakness of the one-time pad is that the key must never be used more than once.

EXAMPLE: In 1943, the U.S. Army's Signal Intelligence Service began to examine messages sent from Soviet agents in the United States to Moscow. These messages were encoded using a complex cipher that was based on a one-time pad, which in this context was a set of pages of random number groups. This in theory made the messages unbreakable. But sometimes the manufacturers of these pads reused pages. Taking advantage of this duplication, cryptanalysts in the Signal Intelligence Service and, later, the U.S. National Security Agency, were able to decipher many of the messages sent between 1943 and 1980, providing insight into Soviet espionage of that time.

10.2.3 Data Encryption Standard

The Data Encryption Standard (DES) [2146] is one of the most important symmetric cryptosystems in the history of cryptography. It provided the impetus for

With proper spacing, capitalization, 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."

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 [171, 173]. 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.

10.2.2.2 One-Time Pad

Repetitions provide a means for the cryptanalyst to attack the Vigènere cipher. The *one-time pad* is a variant of the Vigenère cipher with a key that is at least as long as the message and is chosen at random, so it does not repeat. Technically, it is a threshold scheme (see Section 16.3.2), and is provably impossible to break [240] (see also Section C.3.3, "Perfect Secrecy").

The weakness of the one-time pad is that the key must never be used more than once.

EXAMPLE: In 1943, the U.S. Army's Signal Intelligence Service began to examine messages sent from Soviet agents in the United States to Moscow. These messages were encoded using a complex cipher that was based on a one-time pad, which in this context was a set of pages of random number groups. This in theory made the messages unbreakable. But sometimes the manufacturers of these pads reused pages. Taking advantage of this duplication, cryptanalysts in the Signal Intelligence Service and, later, the U.S. National Security Agency, were able to decipher many of the messages sent between 1943 and 1980, providing insight into Soviet espionage of that time.

10.2.3 Data Encryption Standard

The Data Encryption Standard (DES) [2146] is one of the most important symmetric 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 used infrequently, it is well worth understanding.

In 1973, the U.S. National Bureau of Standards (NBS)[1] invited the submission of proposals for a cryptographic system, in an effort to develop a commercial standard that could also be used for unclassified government communications. The requirements included that the algorithm be made public, available to all to use freely, efficient, and economic to implement. They received no suitable proposals. In 1974, the NBS issued another invitation. At the time, IBM was developing a cryptosystem for use in the commercial world [1903]. IBM submitted this algorithm, LUCIFER [1791], to the NBS, which requested the U.S. National Security Agency's help in evaluating the algorithm. It modified the algorithm in several ways, published the modified algorithm, and held two workshops to evaluate the cryptosystem. The modified cryptosystem was adopted as a standard in 1976 [1684].

10.2.3.1 Structure

The DES 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*. Thus, $\mathcal{M}$, $\mathcal{K}$, and $\mathcal{C}$ are sets of all combinations of 64 bits, $\mathcal{E}$ the DES encryption algorithm, and $\mathcal{D}$ the DES decryption algorithm.

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. 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.

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 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.

Section F.1 describes the algorithm in detail, and presents the tables involved.

[1] The name was later changed to the National Institute of Standards and Technology (NIST).

10.2.3.2 Analysis of the DES

When the DES was first announced, it was criticized as too weak. First, Diffie and Hellman [565] 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 4 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 $DES_k(m)$ be the encipherment of plaintext m under key k. Then the *complementation property* states that

$$DES_k(m) = c \Rightarrow DES_{\overline{k}}(\overline{m}) = \overline{c}$$

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 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 [1904]. 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 [1327]. 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 [140].

In 1990, a breakthrough in cryptanalysis answered many of these questions. Biham and Shamir applied a technique called *differential cryptanalysis* to the DES [204, 206, 207]. 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, the initial version of 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* [1261, 1262]; this attack requires 2^{43} known plaintext and ciphertext pairs on average.) Second, small changes in the S-boxes weakened the cipher, reducing the required number of chosen plaintext and ciphertext pairs. 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 [459].

10.2.3.3 DES and Modes

The DES is used in several modes [2147]. Using it directly is called *electronic codebook* (ECB) mode, and is very rare. Modes in which it can be used to generate a pseudo-one-time pad are *cipher feedback* (CFB) mode (see Section 12.2.1.2) and *output feedback* (OFB) mode (see Section 12.2.1.1). Its most common modes of use are *cipher block chaining* (CBC) mode (see Section 12.2.2), *encrypt-decrypt-encrypt* (EDE) mode, and *triple DES* mode (the EDE and triple DES modes are described in Section 12.2.2.1).

10.2.3.4 Retirement of the DES

In 1998, a design for a computer system and software that could break any DES-enciphered message in a few days was published [625]. 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 fill the needs that the DES no longer filled. In 2001, the Advanced Encryption Standard was announced (see Section 10.2.5), and in 2005, NIST officially withdrew the DES [138]. Triple DES mode remains the only approved implementation [127].

10.2.4 Other Modern Symmetric Ciphers

Several algorithms were proposed to overcome the weaknesses found 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 [1703]. However, it can be broken using an attack similar to differential cryptanalysis [1023]. FEAL has a block size of 64 bits and a key size of 64 bits [1364, 1735]. FEAL-4 (FEAL with 4 rounds) and FEAL-8 (FEAL with 8 rounds) fell to differential cryptanalysis with 20 [1404] and 10,000 [770] 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 [206]. 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 [485] has an 80-bit block and a 160-bit key. It has 10 rounds, and although a single round was successfully cryptanalyzed [205], the use of 10 rounds appears to withstand differential cryptanalysis.

LOKI89 [304], proposed as an alternative to the DES, was vulnerable to differential cryptanalysis [206]. Its successor, LOKI91 [302], uses a 64-bit key and a 64-bit block size. Linear cryptanalysis fails to break this cipher [1883].

LOKI97 [303] uses a 128-bit block size and a 256-bit key schedule, but is believed to be vulnerable to both linear and differential cryptanalysis [1075]. Khufu [1323] 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 [1323], similar in design to Khufu, uses fixed S-boxes, but it has been broken [206].

IDEA is an 8-round cipher that uses 64-bit blocks and 128-bit keys [1124]. It uses three operations: exclusive or, addition modulo 2^{16}, and multiplication modulo $2^{16} + 1$. It appears to withstand known attacks [881, 1125] but variants with fewer than the full 8 rounds have been broken [203, 532]. It is used in commercial software—notably, in the electronic mail program PGP (and not the GNU software GPG) [1213]—but is patented and requires licensing for use in commercial software.

Schneier developed Blowfish [1683] as an alternative to the DES, unencumbered by patents. It appears to be secure against linear cryptanalysis [1418], but has been superseded by Twofish [1690, 1691], a finalist for the Advanced Encryption Standard (AES), the successor to the DES (see Section 10.2.5). Other ciphers that were finalists for the AES were Serpent [199], RC6 [1597], and MARS [328]. These were extensively analyzed as part of that competition [201, 365,950,1021,1022,1737].

10.2.5 Advanced Encryption Standard

In 1997, the U.S. National Institute of Standards and Technology (NIST) announced a competition to select the successor to the DES. Like the DES, the chosen algorithm had to be available for royalty-free use. Unlike the DES, it was to encipher 128 bit blocks and use keys of 128, 192, and 256 bits. Initially, 21 cryptosystems were submitted. The developers presented the cryptosysytems in two workshops, and then selected Twofish, Serpent, RC6, MARS, and Rijndael. After a third workshop, NIST announced that Rijndael was selected to be the Advanced Encryption Standard [2116].

10.2.5.1 Structure

Like the DES, the AES is a bit-oriented product cipher. Unlike the DES, the AES can use keys of 128, 192, or 256 bits and operates on 128 bits of input, producing 128 bits of output. The number of rounds in the AES depends upon the key length—10 rounds if the key is 128 bits, 12 rounds if the key is 192 bits, and 14 rounds if the key is 256 bits. Thus, $\mathcal{M}$ and $\mathcal{C}$ are sets of all combinations of 128 bits and $\mathcal{K}$ is the set of all combinations of 128, 192, or 256 bits, depending on the key length chosen. $\mathcal{E}$ is the AES encryption algorithm for the key length selected, and $\mathcal{D}$ the corresponding AES decryption algorithm.

The AES maintains a state array that initially consists of the input. Each round transforms the state array, and the contents of the array at the end of the last round is the output.

Associated with each round is a round key. If the AES is n rounds, there will be n round keys. The original key is divided into 4-byte words.[2] The *RotWord* transformation rotates the word by 1 byte; the *SubWord* transformation changes the bytes by applying an S-box. The result is xor'ed with a bit string, and then with the corresponding word of the previous round (or the initial key, if this is the first round key). Each round key consists of 4, 6, or or 8 words depending on the length of the original key.

To begin the encryption, the transformation *AddRoundKey* combines the supplied key with the state array. Next come a series of rounds, each of which (except the last) consists of four operations. First, the *SubBytes* transformation substitutes new values for each byte in the state array using an S-box. Then, the *ShiftRows* transformation cyclically shifts rows. The *MixColumns* transformation alters each column independently, and then the *AddRoundKey* transformation xors the state with the round key. The last round omits the *MixColumns* transformation. The contents of the resulting state array is the output.

Decryption is accomplished in a similar fashion. The round key schedule is reversed, and three of the four transformations are changed. In each round, the *InvShiftRows* transformation, which is the inverse of the *ShiftRows* transformation used in encryption, shifts the rows of the state array. The *InvSubBytes* transformation reverses the *SubBytes* transformation using an S-box that is the inverse of the one associated with *SubBytes*. Then the *AddRoundKey* transformation xors in the appropriate round key, and the *InvMixColumns* transformation, again the inverse of the *MixColumns* transformation, reverses the *MixColumns* transformation. The final round omits the *InvMixColumns* transformation.

An alternate expression of the decryption algorithm notes that *InvShiftRows* and *InvSubBytes* commute with respect to (functional) composition, and that *InvMixColumns* is linear with respect to the column input. Given these, the *Equivalent Inverse Cipher* algorithm exchanges the order of the *InvShiftRows* and *InvSubBytes* transformations, applies *InvMixColumns* to all round keys except the initial key and the final round key, and then exchanges the order of the *InvMixColumns* and *AddRoundKey* transformations. This provides a more efficient structure for decryption, paralleling the structure of encryption.

Section F.2 describes the algorithm in detail, including the tables and transformations involved.

10.2.5.2 Analysis of the AES

The designers constructed the AES to withstand the attacks to which the DES showed weakness [490]. As with the DES, the selection of the values in the S-box is critical. Unlike the DES, the developers described the design principles underlying the choice of S-box. The first is nonlinearity, so the output of the transformation is not a linear function of the input. The second is algebraic complexity, so the inverse of each byte is obtained, and this is remapped with an affine transformation. The result is that no input to the S-box is ever mapped either to itself or to its bitwise complement.

[2] A "byte" in this context is 8 bits, regardless of the underlying architecture.