right pair, then we test each possible candidate subkey (L_1, L_2) and increment an appropriate counter if a certain x-or is observed. The steps include computing an exclusive-or with candidate subkeys and applying the inverse S-box (as was done in Algorithm 3.2), followed by computation of the relevant xor-value.

A differential attack based on a differential trail having propagation ratio equal to ϵ will often be successful if the number of tuples (x, x^*, y, y^*), which we denote by T, is approximately $c\,\epsilon^{-1}$, for a "small" constant c. We implemented the attack described in Algorithm 3.3, and found that the attack was often successful if we took T between 50 and 100. In this example, $\epsilon^{-1} \approx 38$.

3.5 The Data Encryption Standard

On May 15, 1973, the National Bureau of Standards (now the *National Institute of Standards and Technology*, or *NIST*) published a solicitation for cryptosystems in the Federal Register. This led ultimately to the adoption of the *Data Encryption Standard*, or *DES*, which became the most widely used cryptosystem in the world. *DES* was developed at IBM, as a modification of an earlier system known as *Lucifer*. *DES* was first published in the Federal Register of March 17, 1975. After a considerable amount of public discussion, *DES* was adopted as a standard for "unclassified" applications on January 15, 1977. It was initially expected that DES would only be used as a standard for 10–15 years; however, it proved to be much more durable. *DES* was reviewed approximately every five years after its adoption. Its last renewal was in January 1999; by that time, development of a replacement, the *Advanced Encryption Standard*, had already begun (see Section 3.6).

3.5.1 Description of DES

A complete description of the *Data Encryption Standard* is given in the *Federal Information Processing Standards* (*FIPS*) Publication 46, dated January 15, 1977. *DES* is a special type of iterated cipher called a *Feistel cipher*. We describe the basic form of a Feistel cipher now, using the terminology from Section 3.1. In a Feistel cipher, each state u^i is divided into two halves of equal length, say L^i and R^i. The round function g has the following form: $g(L^{i-1}, R^{i-1}, K^i) = (L^i, R^i)$, where

$$L^i = R^{i-1}$$
$$R^i = L^{i-1} \oplus f(R^{i-1}, K^i).$$

We observe that the function f does not need to satisfy any type of injectivity property. This is because a Feistel-type round function is always invertible, given

the round key:

$$L^{i-1} = R^i \oplus f(L^i, K^i)$$
$$R^{i-1} = L^i.$$

DES is a 16-round Feistel cipher having block length 64: it encrypts a plaintext bitstring x (of length 64) using a 56-bit key, K, obtaining a ciphertext bitstring (of length 64). Prior to the 16 rounds of encryption, there is a fixed *initial permutation* **IP** that is applied to the plaintext. We denote

$$\mathbf{IP}(x) = L^0 R^0.$$

After the 16 rounds of encryption, the inverse permutation $\mathbf{IP}^{-1}$ is applied to the bitstring $R^{16}L^{16}$, yielding the ciphertext y. That is,

$$y = \mathbf{IP}^{-1}(R^{16}L^{16})$$

(note that L^{16} and R^{16} are swapped before $\mathbf{IP}^{-1}$ is applied). The application of **IP** and $\mathbf{IP}^{-1}$ has no cryptographic significance, and is often ignored when the security of *DES* is discussed. One round of *DES* encryption is depicted in Figure 3.6.

Each L^i and R^i is 32 bits in length. The function

$$f : \{0,1\}^{32} \times \{0,1\}^{48} \to \{0,1\}^{32}$$

takes as input a 32-bit string (the right half of the current state) and a round key. The key schedule, $(K^1, K^2, \ldots, K^{16})$, consists of 48-bit round keys that are derived from the 56-bit key, K. Each K^i is a certain permuted selection of bits from K.

The f function is shown in Figure 3.7. Basically, it consists of a substitution (using an S-box) followed by a (fixed) permutation, denoted **P**. Suppose we denote the first argument of f by A, and the second argument by J. Then, in order to compute $f(A, J)$, the following steps are executed.

1. A is "expanded" to a bitstring of length 48 according to a fixed *expansion function* **E**. $\mathbf{E}(A)$ consists of the 32 bits from A, permuted in a certain way, with 16 of the bits appearing twice.
2. Compute $\mathbf{E}(A) \oplus J$ and write the result as the concatenation of eight 6-bit strings $B = B_1B_2B_3B_4B_5B_6B_7B_8$.
3. The next step uses eight S-boxes, denoted $S_1, \ldots, S_8$. Each S-box

   $$S_i : \{0,1\}^6 \to \{0,1\}^4$$

   maps six bits to four bits, and is traditionally depicted as a 4×16 array whose entries come from the integers $0, \ldots, 15$. Given a bitstring of length six, say

   $$B_j = b_1b_2b_3b_4b_5b_6,$$

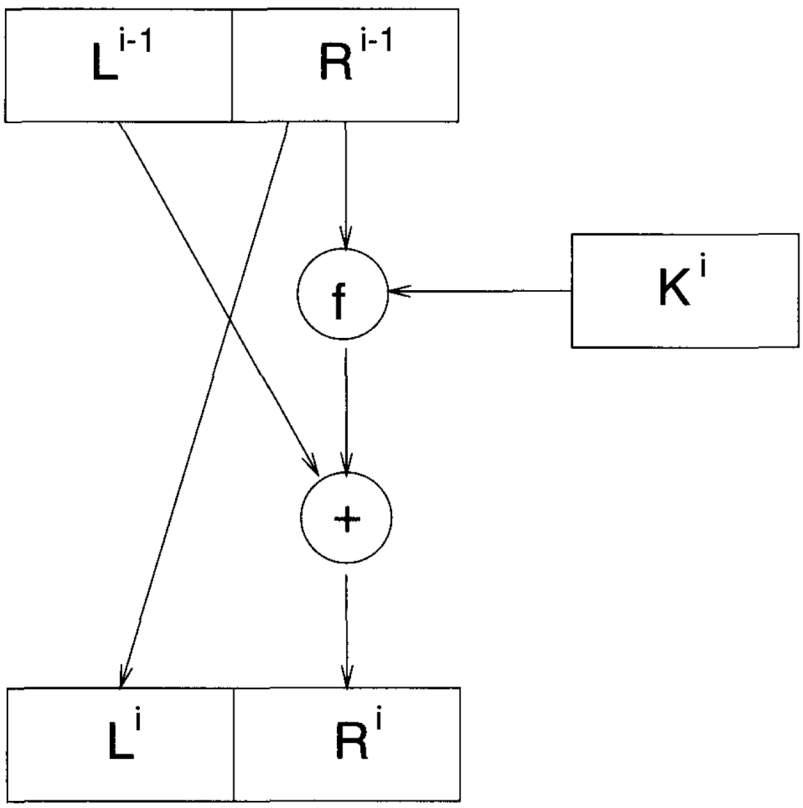

FIGURE 3.6
One round of DES encryption

we compute $S_j(B_j)$ as follows. The two bits b_1b_6 determine the binary representation of a row r of S_j (where $0 \leq r \leq 3$), and the four bits $b_2b_3b_4b_5$ determine the binary representation of a column c of S_j ($0 \leq c \leq 15$). Then $S_j(B_j)$ is defined to be the entry $S_j(r, c)$, written in binary as a bitstring of length four. In this fashion, we compute $C_j = S_j(B_j)$, $1 \leq j \leq 8$.

4. The bitstring

$$C = C_1C_2C_3C_4C_5C_6C_7C_8$$

of length 32 is permuted according to the permutation $\mathbf{P}$. The resulting bitstring $\mathbf{P}(C)$ is defined to be $f(A, J)$.

For future reference, the eight *DES* S-boxes are now presented:

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

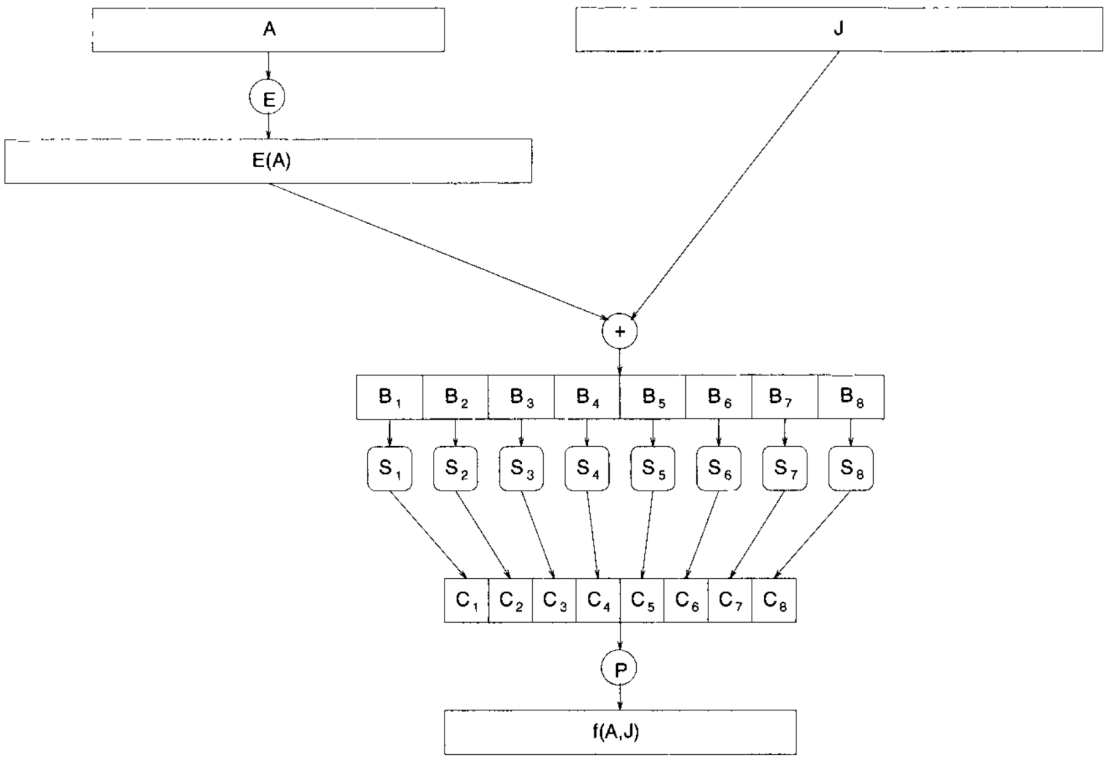

FIGURE 3.7
The DES f function

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Example 3.4 We show how to compute a sample output of an S-box, using the traditional presentation described above. Consider the S-box S_1, and suppose that the input is the binary 6-tuple 101000. The first and last bits are 10, which is the binary representation of the integer 2. The middle four bits are 0100, which is the binary representation of the integer 4. Row 2 of S_1 is the third row (because the rows are numbered $0, 1, 2, 3$); similarly, column 4 is the fifth column. The entry in row 2, column 4 of S_1 is 13, which is 1101 in binary. Therefore 1101 is the output of the S-box S_1, given the input 101000. □

The *DES* S-boxes are not permutations, of course, because the number of possible inputs (64) exceeds the number of possible outputs (16). However, it can be verified that each row of each of the eight S-boxes is a permutation of the integers $0, \ldots, 15$. This property is one of several design criteria that were required of the S-boxes in order to prevent certain types of cryptanalytic attacks.

The expansion function **E** is specified by the following table:

E bit-selection table					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Given a bitstring of length 32, say $A = (a_1, a_2, \ldots, a_{32})$, $\mathbf{E}(A)$ is the following bitstring of length 48:

$$\mathbf{E}(A) = (a_{32}, a_1, a_2, a_3, a_4, a_5, a_4, \ldots, a_{31}, a_{32}, a_1).$$

The permutation **P** is as follows:

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Denote the bitstring $C = (c_1, c_2, \ldots, c_{32})$. Then the permuted bitstring $\mathbf{P}(C)$ is as follows:

$$\mathbf{P}(C) = (c_{16}, c_7, c_{20}, c_{21}, c_{29}, \ldots, c_{11}, c_4, c_{25}).$$

3.5.2 Analysis of DES

When *DES* was proposed as a standard, there was considerable criticism. One objection to *DES* concerned the S-boxes. All computations in *DES*, with the sole exception of the S-boxes, are linear, i.e., computing the exclusive-or of two outputs is the same as forming the exclusive-or of two inputs and then computing the output. The S-boxes, being the non-linear components of the cryptosystem, are vital to its security. (We saw in Chapter 1 how linear cryptosystems, such as the *Hill Cipher*, could easily be cryptanalyzed by a known plaintext attack.) At the time that *DES* was proposed, several people suggested that its S-boxes might contain hidden "trapdoors" which would allow the National Security Agency to easily decrypt messages while claiming falsely that *DES* is "secure." It is, of

course, impossible to disprove such a speculation, but no evidence ever came to light that indicated that trapdoors in *DES* do, in fact, exist.

Actually, it was eventually revealed that the *DES* S-boxes were designed to prevent certain types of attacks. When Biham and Shamir invented the technique of differential cryptanalysis (which we discussed in Section 3.4) in the early 1990s, it was acknowledged that the purpose of certain unpublished design criteria of the S-boxes was to make differential cryptanalysis of *DES* infeasible. Differential cryptanalysis was known to IBM researchers at the time that *DES* was being developed, but it was kept secret for almost 20 years, until Biham and Shamir independently discovered the attack.

The most pertinent criticism of *DES* is that the size of the keyspace, 2^{56}, is too small to be really secure. The IBM *Lucifer* cryptosystem, a predecessor of *DES*, had a 128-bit key. The original proposal for *DES* had a 64-bit key, but this was later reduced to a 56-bit key. IBM claimed that the reason for this reduction was that it was necessary to include eight parity-check bits in the key, meaning that 64 bits of storage could only contain a 56-bit key.

Even in the 1970s, it was argued that a special-purpose machine could be built to carry out a known plaintext attack, which would essentially perform an exhaustive search for the key. That is, given a 64-bit plaintext x and corresponding ciphertext y, every possible key would be tested until a key K is found such that $e_K(x) = y$ (note that there may be more than one such key K). As early as 1977, Diffie and Hellman suggested that one could build a VLSI chip which could test 10^6 keys per second. A machine with 10^6 chips could search the entire key space in about a day. They estimated that such a machine could be built, at that time, for about $20,000,000.

Later, at the CRYPTO '93 Rump Session, Michael Wiener gave a very detailed design of a *DES* key search machine. The machine is based on a key search chip which is pipelined, so that 16 encryptions take place simultaneously. This chip would test 5×10^7 keys per second, and could have been built using 1993 technology for $10.50 per chip. A frame consisting of 5760 chips could be built for $100,000. This would allow a *DES* key to be found in about 1.5 days on average. A machine using ten frames would cost $1,000,000, but would reduce the average search time to about 3.5 hours.

Wiener's machine was never built, but a key search machine costing $250,000 was built in 1998 by the Electronic Frontier Foundation. This computer, called "DES Cracker," contained 1536 chips and could search 88 billion keys per second. It won RSA Laboratory's "DES Challenge II-2" by successfully finding a *DES* key in 56 hours in July 1998. In January 1999, RSA Laboratory's "DES Challenge III" was solved by the DES Cracker working in conjunction with a worldwide network (of 100,000 computers) known as `distributed.net`. This co-operative effort found a *DES* key in 22 hours, 15 minutes, testing over 245 billion keys per second.

Other than exhaustive key search, the two most important cryptanalytic attacks on *DES* are differential cryptanalysis and linear cryptanalysis. (For SPNs, these

attacks were described in Sections 3.4 and 3.3, respectively.) In the case of *DES*, linear cryptanalysis is the more efficient of the two attacks, and an actual implementation of linear cryptanalysis was carried out in 1994 by its inventor, Matsui. This linear cryptanalysis of *DES* is a known-plaintext attack using 2^{43} plaintext-ciphertext pairs, all of which are encrypted using the same (unknown) key. It took 40 days to generate the 2^{43} pairs, and it took 10 days to actually find the key. This cryptanalysis did not have a practical impact on the security of *DES*, however, due to the extremely large number of plaintext-ciphertext pairs that are required to mount the attack: it is unlikely in practice that an adversary will be able to accumulate such a large number of plaintext-ciphertext pairs that are all encrypted using the same key.

3.6 The Advanced Encryption Standard

On January 2, 1997, NIST began the process of choosing a replacement for *DES*. The replacement would be called the *Advanced Encryption Standard*, or *AES*. A formal call for algorithms was made on September 12, 1997. It was required that the *AES* have a block length of 128 bits, and support key lengths of 128, 192 and 256 bits. It was also necessary that the *AES* should be available worldwide on a royalty-free basis.

Submissions were due on June 15, 1998. Of the 21 submitted cryptosystems, 15 met all the necessary criteria and were accepted as *AES* candidates. NIST announced the 15 *AES* candidates at the "First AES Candidate Conference" on August 20, 1998. A "Second AES Candidate Conference" was held in March 1999. Then, in August 1999, five of the candidates were chosen by NIST as finalists: *MARS*, *RC6*, *Rijndael*, *Serpent* and *Twofish*.

The "Third AES Candidate Conference" was held in April 2000. On October 2, 2000, *Rijndael* was selected to be the *Advanced Encryption Standard*. On February 28, 2001, NIST announced that a draft Federal Information Processing Standard for the *AES* was available for public review and comment. *AES* was adopted as a standard on November 26, 2001, and it was published as FIPS 197 in the Federal Register on December 4, 2001.

The selection process for the *AES* was notable for its openness and its international flavor. The three candidate conferences, as well as official solicitations for public comments, provided ample opportunity for feedback and public discussion and analysis of the candidates, and the process was viewed very favorably by everyone involved. The "international" aspect of *AES* is demonstrated by the variety of countries represented by the authors of the 15 candidate ciphers: Australia, Belgium, Canada, Costa Rica, France, Germany, Israel, Japan, Korea, Norway, the United Kingdom and the USA. *Rijndael*, which was ultimately selected as the *AES*, was invented by two Belgian researchers, Daemen and Rijmen.