Back to TOC

3.3 DES: THE DATA ENCRYPTION STANDARD

- Adopted by NIST in 1977.

- Based on a cipher (Lucifer) developed earlier by IBM for Lloyd's of London for cash transfer.

- DES uses the Feistel cipher structure with 16 rounds of processing.

- DES uses a 56-bit encryption key. (The key size was apparently dictated by the memory and processing constraints imposed by a single-chip implementation of the algorithm for DES.) The key itself is specified with 8 bytes, but one bit of each byte is used as a parity check.

- **DES encryption was broken in 1999 by Electronics Frontiers Foundation (EFF, `www.eff.org`).** This resulted in NIST issuing a new directive that year that required organizations to use **Triple DES**, that is, three consecutive applications of **DES**. (That DES was found to be not as strong as originally believed also prompted NIST to initiate the development of new standards for data encryption. The result is **AES** that we will discuss later.)

- **Triple DES** continues to enjoy wide usage in commercial applications even today. To understand Triple DES, you must first understand the basic **DES** encryption.

- As mentioned, DES uses the Feistel structure with 16 rounds.

- **What is specific to DES is the implementation of the F function in the algorithm and how the round keys are derived from the main encryption key.**

- As will be explained in Section 3.3.5, the round keys are generated from the main key by a sequence of permutations. Each round key is 48 bits in length.

Back to TOC

3.3.1 One Round of Processing in DEA

- The algorithmic implementation of DES is known as **DEA** for **Data Encryption Algorithm.**

- Figure 4 shows a single round of processing in DEA. The dotted rectangle constitutes the F function.

- The 32-bit right half of the 64-bit input data block is expanded by into a 48-bit block. This is referred to as the **expansion permutation** step, or the **E-step**.

- The above-mentioned E-step entails the following:
  - first divide the 32-bit block into eight 4-bit words
  - attach an additional bit on the left to each 4-bit word that is the last bit of the previous 4-bit word
  - attach an additional bit to the right of each 4-bit word that is the beginning bit of the next 4-bit word.

  Note that what gets prefixed to the first 4-bit block is the last bit of the last 4-bit block. By the same token, what gets appended to the last 4-bit block is the first bit of the first 4-bit block. The reason for why we expand each 4-bit block into a 6-bit block in the manner explained will become clear shortly.

- The 56-bit key is divided into two halves, each half shifted separately, and the combined 56-bit key **permuted/contracted** to yield a 48-bit **round** key. How this is done will be explained later.

- The 48 bits of the expanded output produced by the E-step are XORed with the round key. This is referred to as **key mixing**.

- The output produced by the previous step is broken into eight six-bit words. Each six-bit word goes through a substitution step; its replacement is a 4-bit word. The substitution is carried out with an **S-box**, as explained in greater detail in Section 3.3.2. [The name "S-Box" stands for "Substitution Box".]

- So after all the substitutions, we again end up with a 32-bit word.

- The 32-bits of the previous step then go through a P-box based permutation, as shown in Figure 4.

- What comes out of the P-box is then XORed with the left half of the 64-bit block that we started out with. The output of this XORing operation gives us the right half block for the next round.

- Note that the goal of the substitution step implemented by the **S-box** is to introduce **diffusion** in the generation of the output from the input. *Diffusion means that a change in any plaintext bit must propagate out to as many ciphertext bits as possible.*

- The strategy used for creating the different round keys from the main key is meant to introduce **confusion** into the encryption process. *Confusion in this context means that the relationship between the encryption key and the ciphertext must be as complex as possible.* Another way of describing confusion would be that each bit of the key must affect as many bits as possible of the output ciphertext block.

- Diffusion and confusion are the two cornerstones of block cipher design.

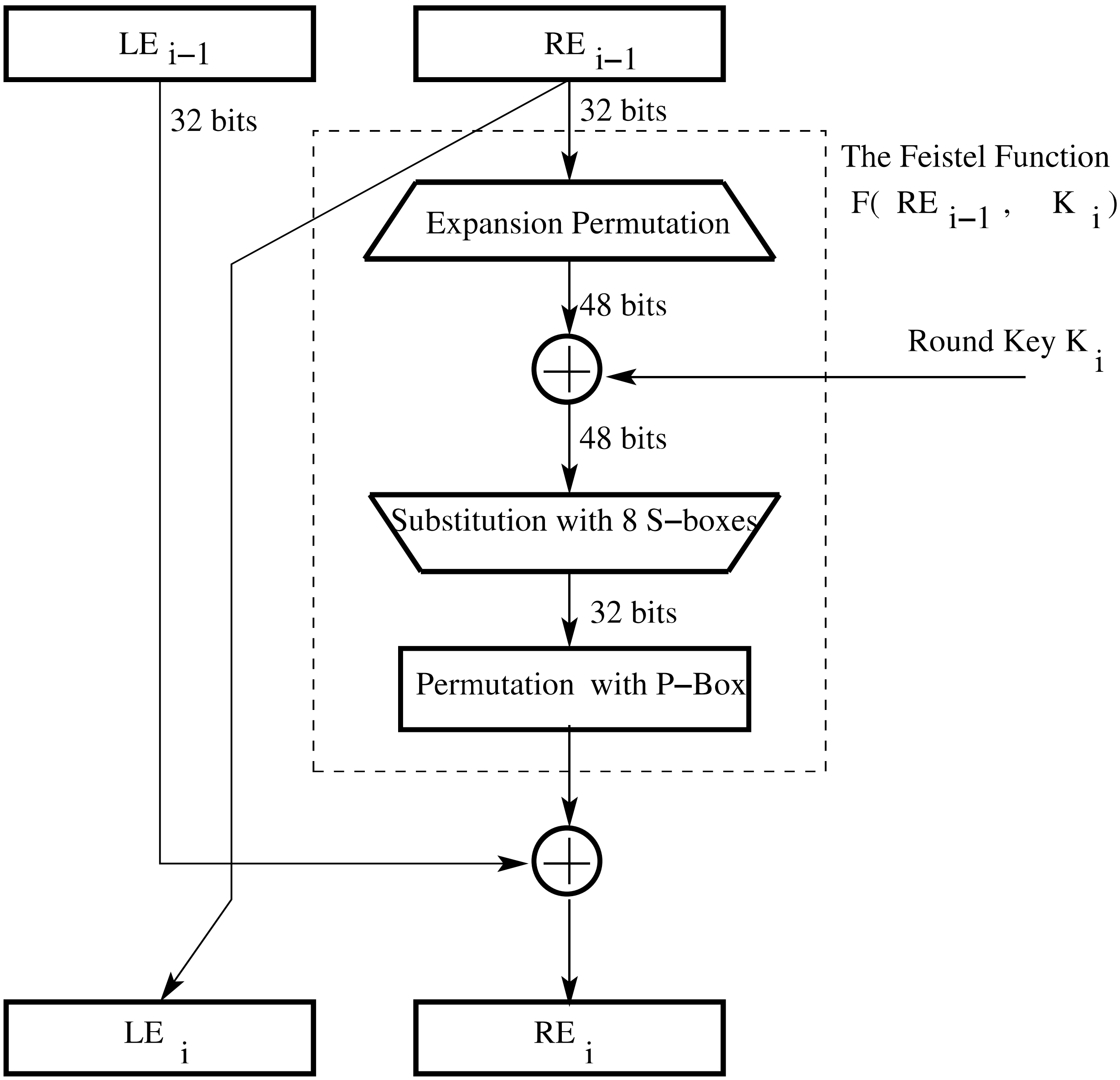

Figure 4: *One round of processing in DES.* (*This figure is from Lecture 3 of "Lecture Notes on Computer and Network Security" by Avi Kak*)