Data Encryption Standard, DES

Exclusive-or encryption is unsafe when there is a pattern to the data that was encrypted (e.g. the redundancy in natural English), but removing all patterns can make it safe. Or safer. Permutation encryption is also unsafe when there is a pattern to the data, but removing all patterns makes it safer. Both Exclusive-or and permutations tend to remove patterns in data. So each could be used to make the other safer. The general idea behind DES is that it performs 16 layers of exclusive-oring and 16 layers of permutations interleaved with each other. It might be possible to trace the changes to a bit back through a couple of layers (and it is), but not through all 32. That's the idea anyway.

The plaintext is split into 64-bit blocks, and each block is permuted and exclusive-ored many times, to produce a 64-bit result. The 64-bit result blocks are re-appended together to form the output. The 56-bit encryption key controls the permutations and exclusive-oring at each stage. The algorithm was very cleverly designed to by symmetric (same key performs both encryption and decryption) and uncrackable. Even having a huge document both in its plaintext form and the encrypted results isn't enough to work out what the encryption key was.

The algorithm:

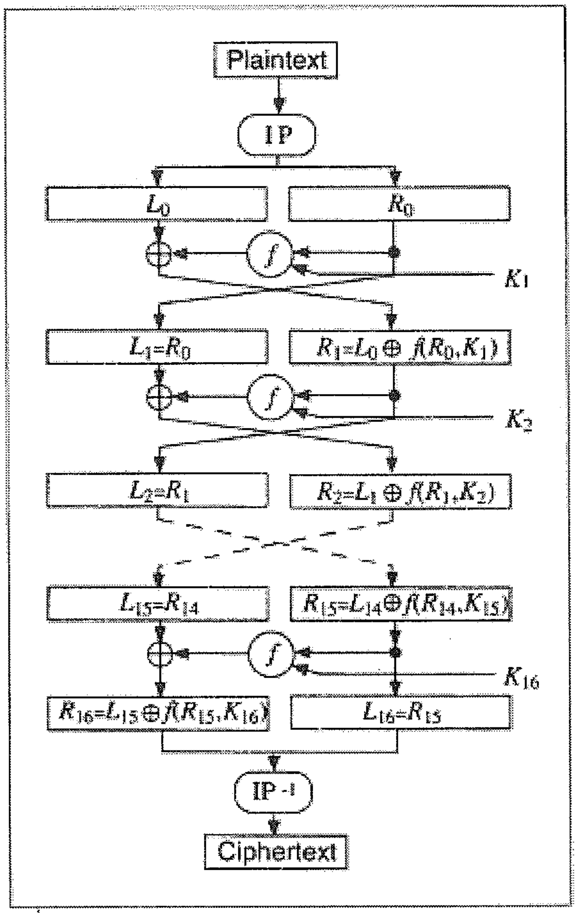
1. Each 64-bit block of input is rearranged by an (irrelevant) input permutation.
2. This step is repeated 16 times:
   - 2a. Bits 0-31 of the output are a copy of bits 32-63 of the input
   - 2b. Bits 32-63 of the input are also expanded to 48 bits by repeating some of them and permuting them.
   - 2c. Those 48 bits are exclusive-ored with 48 bits selected from the key (which 48 bits depends on which of the 16 stages this is)
   - 2d. Those 48 bits permuted. This permutation is called an "S-box", and there is a different S-box for each of the 16 stages.
   - 2e. Those 48 bits are permuted again and reduced back to 32 bits. This reduction/permutation is called a "P-box".
   - 2f. Those 32 bits are exclusive-ored with a copy of bits 0-31 of the input, to provide bits 32-63 of the output
3. After 16 stages, the 64-bit result goes through the inverse of the input permutation to produce the final output.

At each of the 16 repetitions of step 2, the key is rotated and permuted, so a different portion of it is used for each of the 16 repetitions of step 2c.

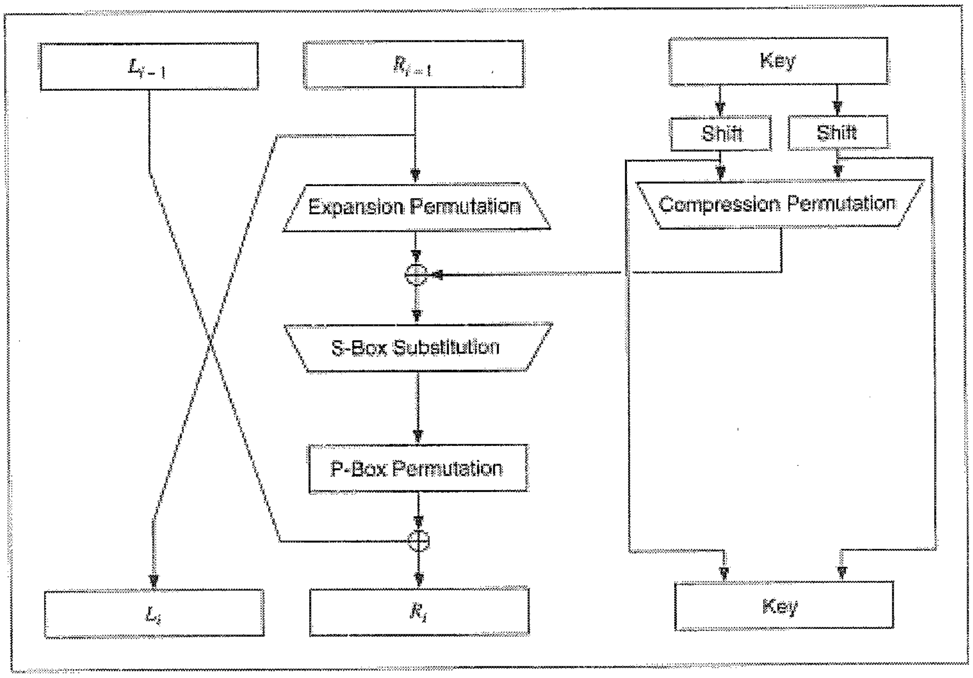
The whole thing is used as a pipe-line process: the input is fed in, 8 bytes at a time, and the output falls out, also 8 bytes at a time. Thus, DES is called a Block Stream Cypher. If you need to encrypt less than 64 bits at a time (e.g. an interactive telnet session, where each keystroke has to be transmitted as soon as it is typed), the inputs must be padded with random rubbish to build them up to 64 bits. You must not pad with zeros.

People often say the key-length for DES is 64 bits, but 8 of those bits are ignored, so it really is only 56, and 2^{56} is not a *very* big number. People also often say that you could easily extend DES to use larger blocks and a larger keys, but DES was very carefully designed by major-league experts and heavily analysed over a number of years. Making your own version is very likely to result in something totally insecure for reasons you might never think of imagining.

The whole DES process


Plaintext
IP
L_0
R_0
f
K_1
$L_1=R_0$
$R_1=L_0 \oplus f(R_0,K_1)$
K_2
$L_2=R_1$
$R_2=L_1 \oplus f(R_1,K_2)$
$L_{15}=R_{14}$
$R_{15}=L_{14} \oplus f(R_{14},K_{15})$
K_{16}
$R_{16}=L_{15} \oplus f(R_{15},K_{16})$
$L_{16}=R_{15}$
IP^{-1}
Ciphertext

One of the 16 rounds:


L_{i-1}
R_{i-1}
Key
Shift
Shift
Expansion Permutation
Compression Permutation
S-Box Substitution
P-Box Permutation
L_i
R_i
Key