CMSC/Math 456: Cryptography (Fall 2022)

Lecture 7

Daniel Gottesman

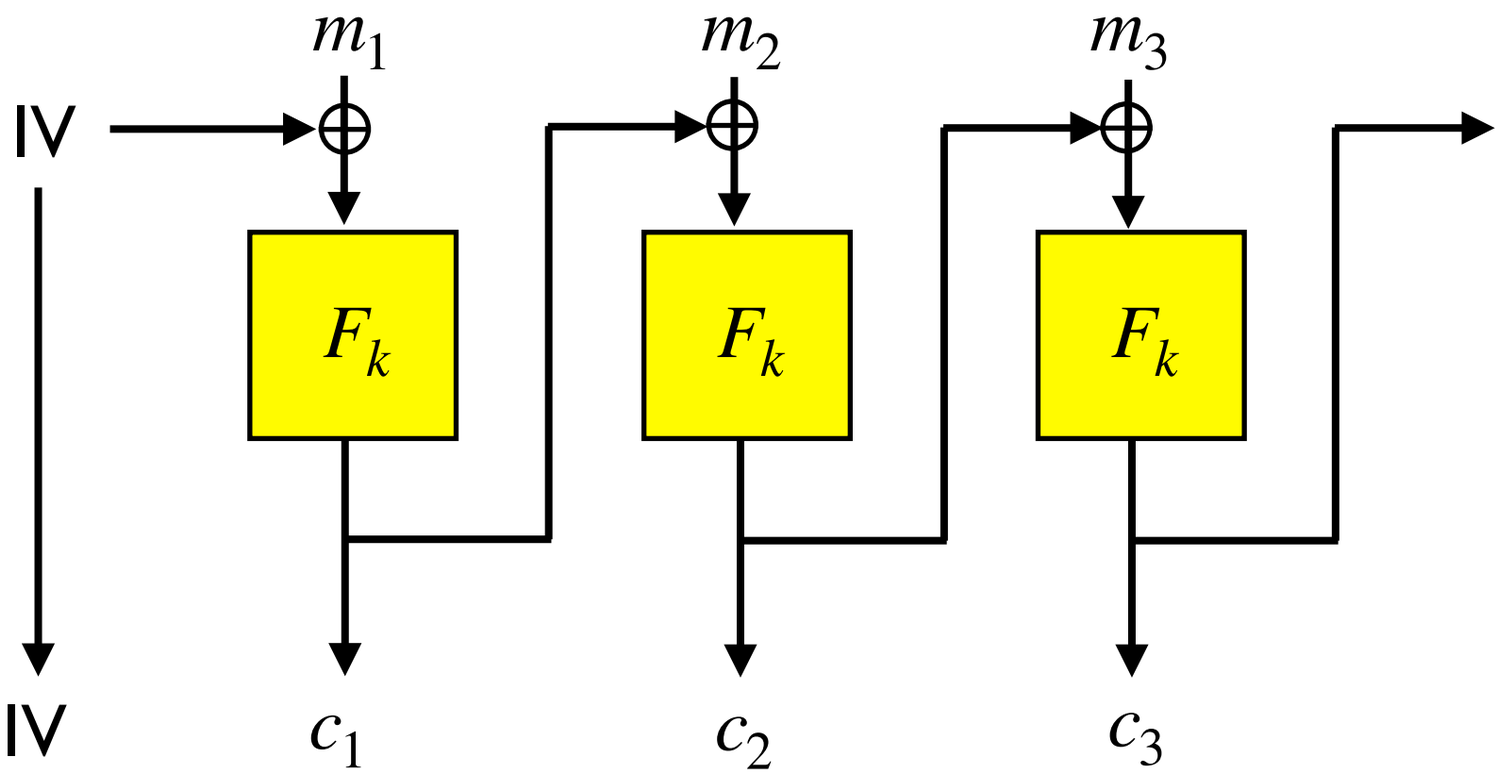
Administrative

Problem Set #3 will be out on Thursday. It will be a programming assignment, and the goal is break a stream cipher I will be providing. You will need to write programs to do this in Python.

This class is being recorded

Block Ciphers

Recall that we were discussing block ciphers that can be secure against a chosen plaintext attack. For example, CBC mode builds on smaller units, each of which is based on a pseudorandom permutation. DES and AES are examples of pseudorandom permutations.



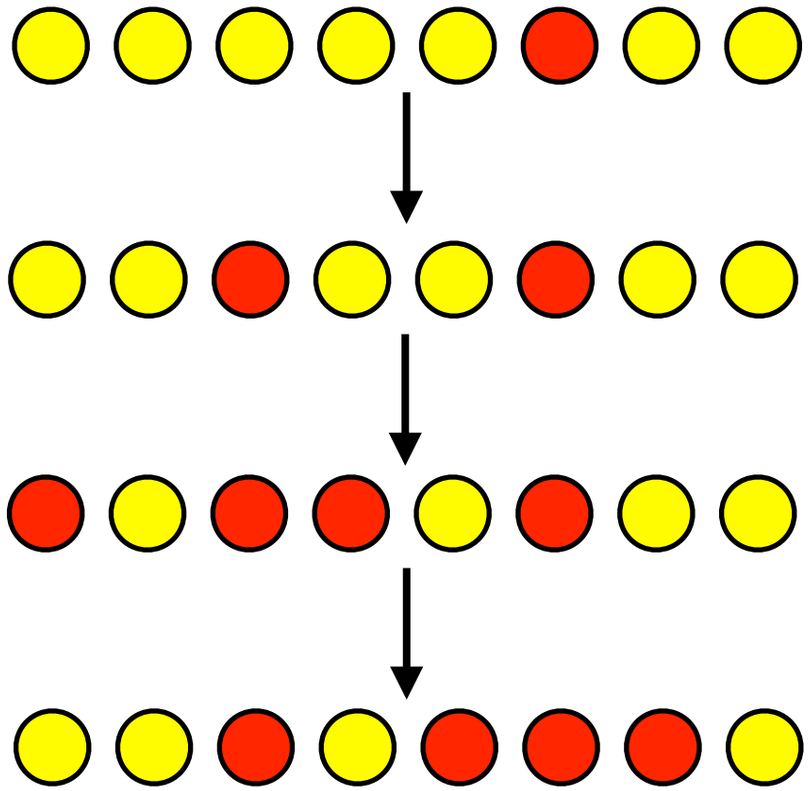
To decrypt, Bob must invert F_k on block i and XOR with c_{i-1}.

This class is being recorded

Goals of Block Cipher Design

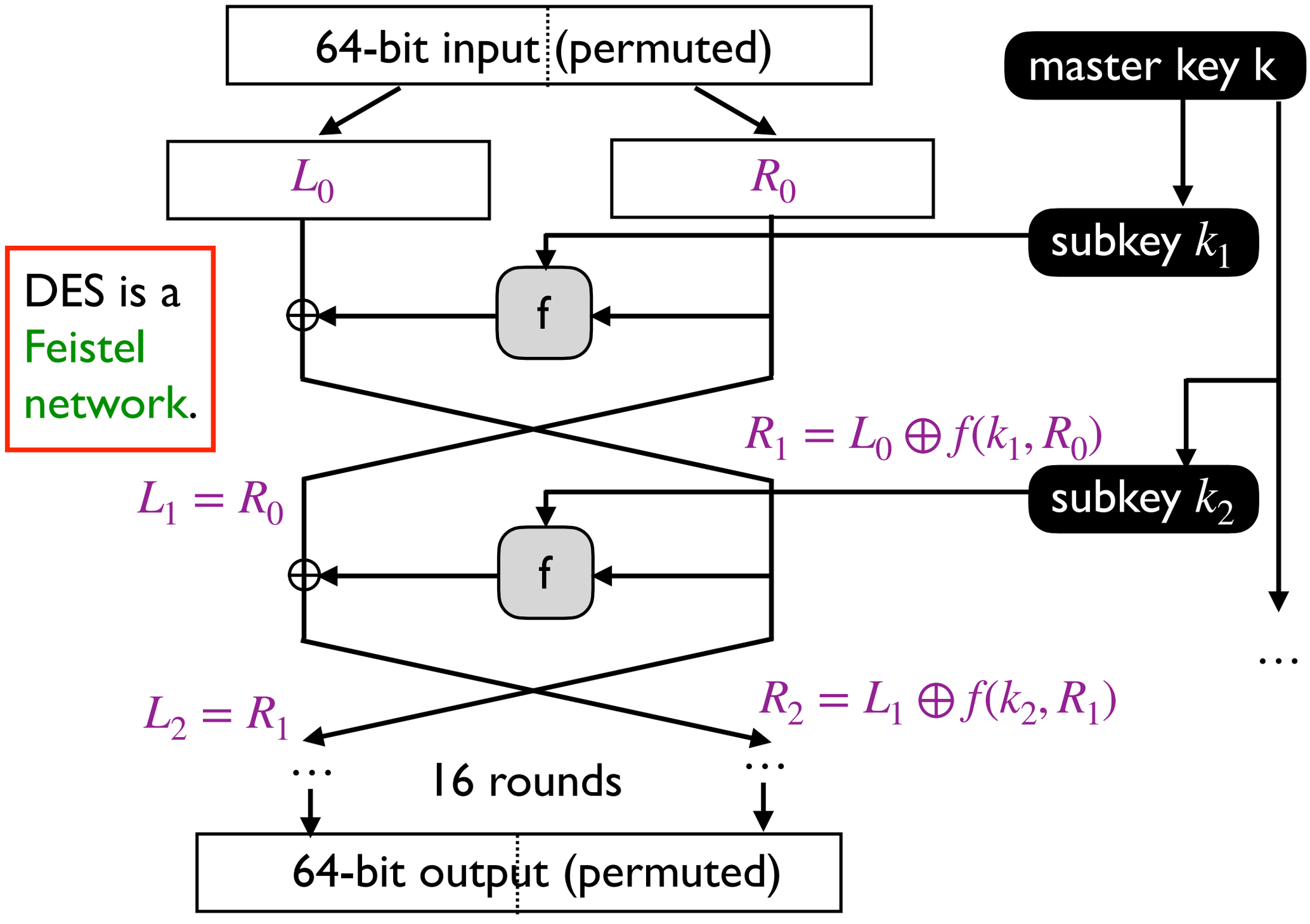
- Must be invertible to use with CBC mode (i.e., pseudorandom permutation rather than pseudorandom function).
- Even when the inputs are related, the outputs should be very different.

In particular, the change of even a single bit of the input should result in a totally different output. This is known as the "avalanche effect." It is often achieved by having multiple rounds, each of which magnifies small changes.



This class is being recorded

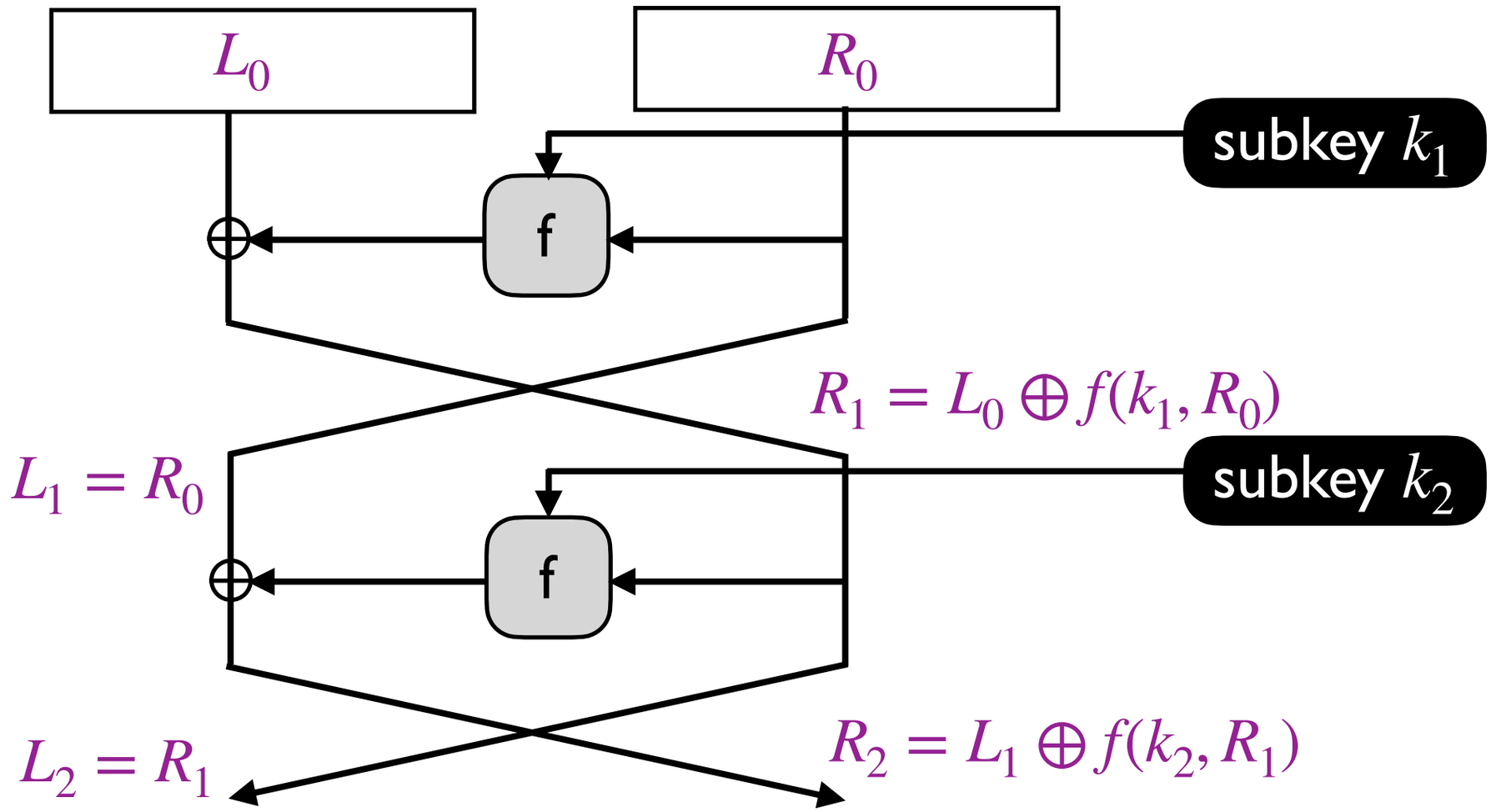
DES Overview



This class is being recorded

Feistel Network

A Feistel network consists of a sequence of rounds sequentially acting on the message, which is split into a left and right half.



In each round, the current right half is fed into a round function f with a key for the round and then XORed with the left half. The modified left half and old right half are then switched.

This class is being recorded

Inverse of Feistal Network

A Feistal network is automatically a permutation. (Permutation here means we are permuting the strings and not just the bits.) In particular, it has a straightforward inverse for someone with access to the key: Simply run the network backwards.

Suppose we know R_i and L_i. Since $L_i = R_{i-1}$, we can always determine what the input to the f function in round i was. Then we can calculate

$$L_{i-1} = R_i \oplus f(k_i, R_{i-1})$$

which gives us R_{i-1} and L_{i-1}. We can then work our back to the beginning.

This class is being recorded

DES Subkeys

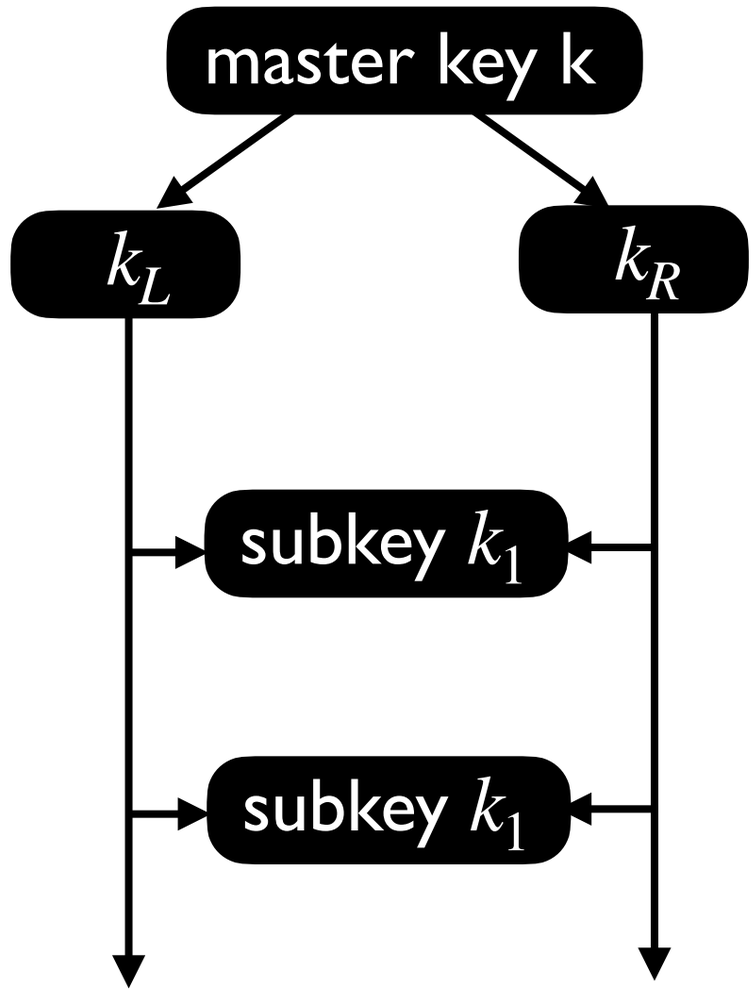
In DES, the master key k is 56 bits.

Each subkey k_i is derived from k via a key schedule:

The master key is then split into two 28-bit halves k_L and k_R.

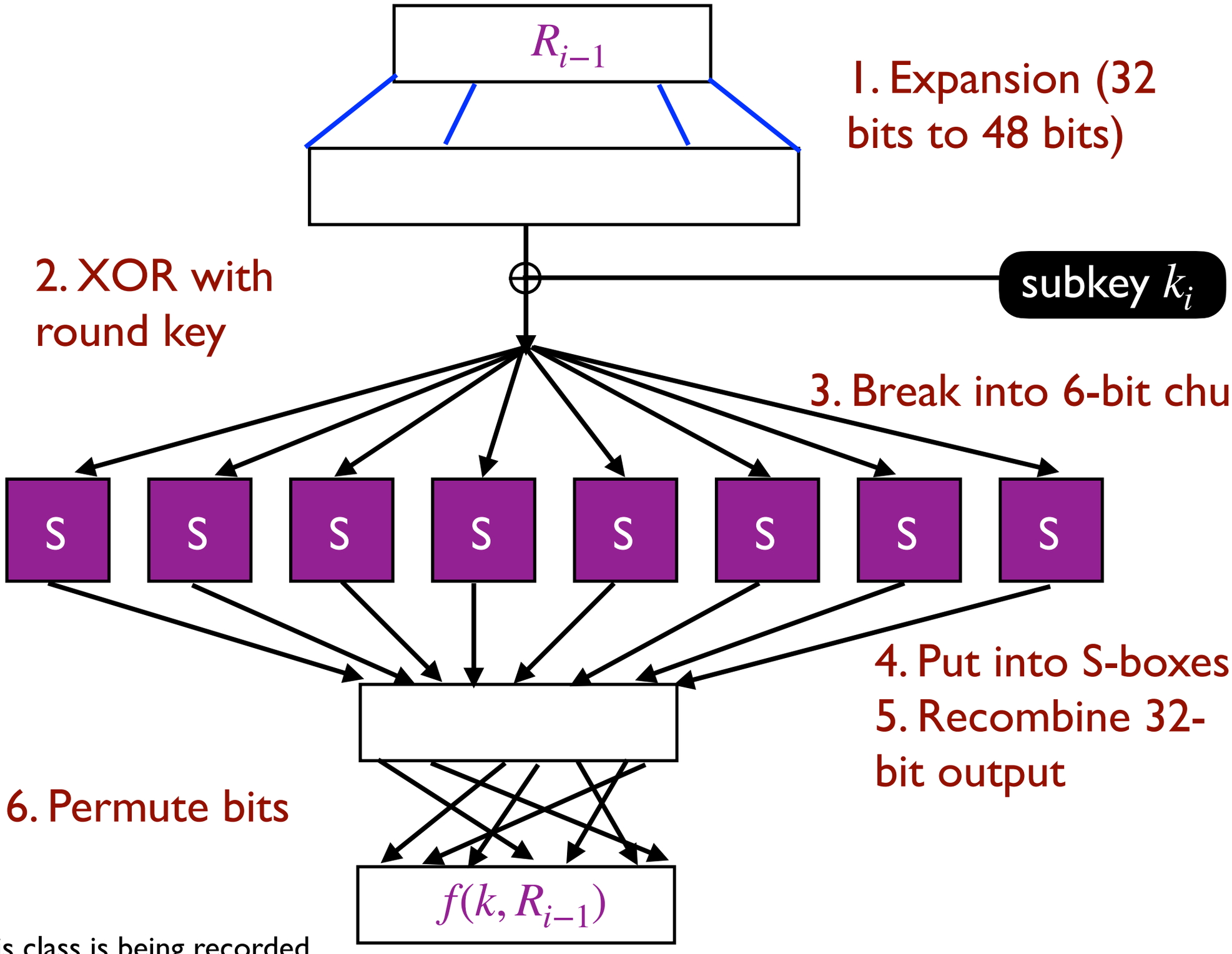
Each subkey k_i is then a permutation of 24 bits from k_L and 24 bits from k_R.

The choice of key bits used is rotated each round so that all key bits are used about the same number of times.



This class is being recorded

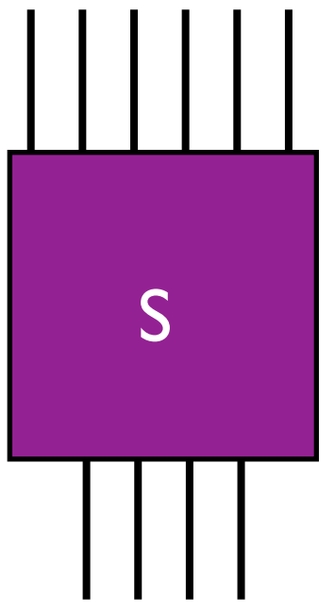
DES Mangler Function



This class is being recorded

DES S-Boxes

Each of the 8 S-boxes in a single mangler function are different, but the set of S-boxes is the same from round to round (as are the expansion function and the final permutation).



The S-box takes 6 bits as inputs and outputs 4 bits, so the 8 S-boxes together reduce the expanded 48-bit string back to the original 32 bits.

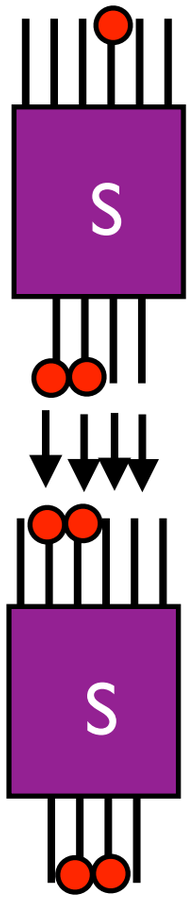
The S-boxes are carefully designed complicated functions defined by lookup tables.

They have the property that all possible outputs are equally likely on a random input (so each possible output can be reached with 4 possible inputs) and that if two possible inputs x_1 and x_2 differ in only one position, the outputs $S(x_1)$ and $S(x_2)$ differ in at least two positions.

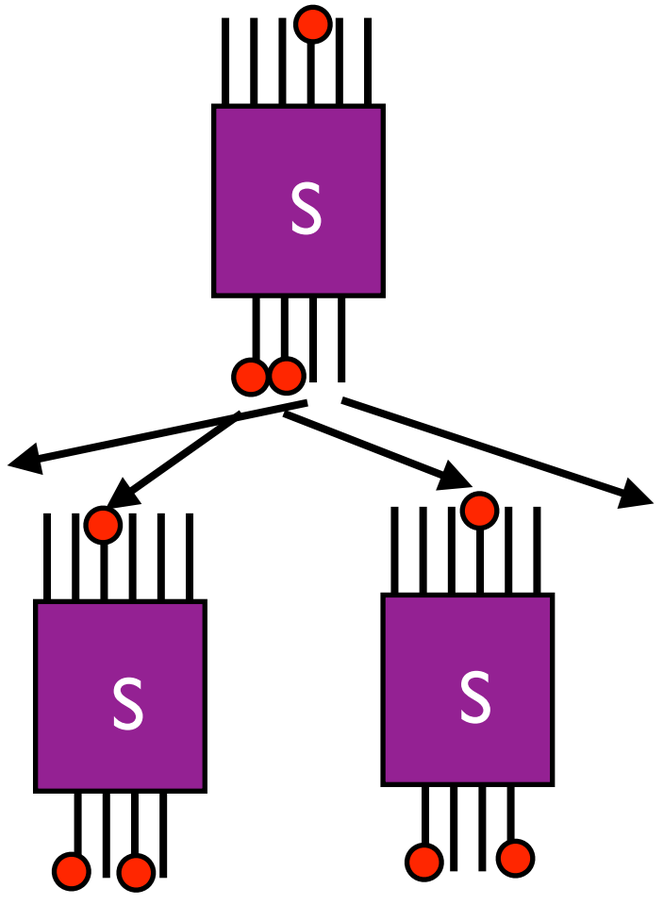
This class is being recorded

Avalanche Effect for Mangler

The S-boxes have the property that they double the size of single-bit disturbances. The expansion step may or may not increase the size of disturbances. The permutation cannot increase the size of a disturbance, but it is still important.



Any disturbance magnified in one round will return to be magnified more in future rounds, but without the permutation, the 2 flipped bits will likely end up in the same S-box and won't be further magnified. The permutation makes sure they hit different S-boxes, where each can be magnified again.

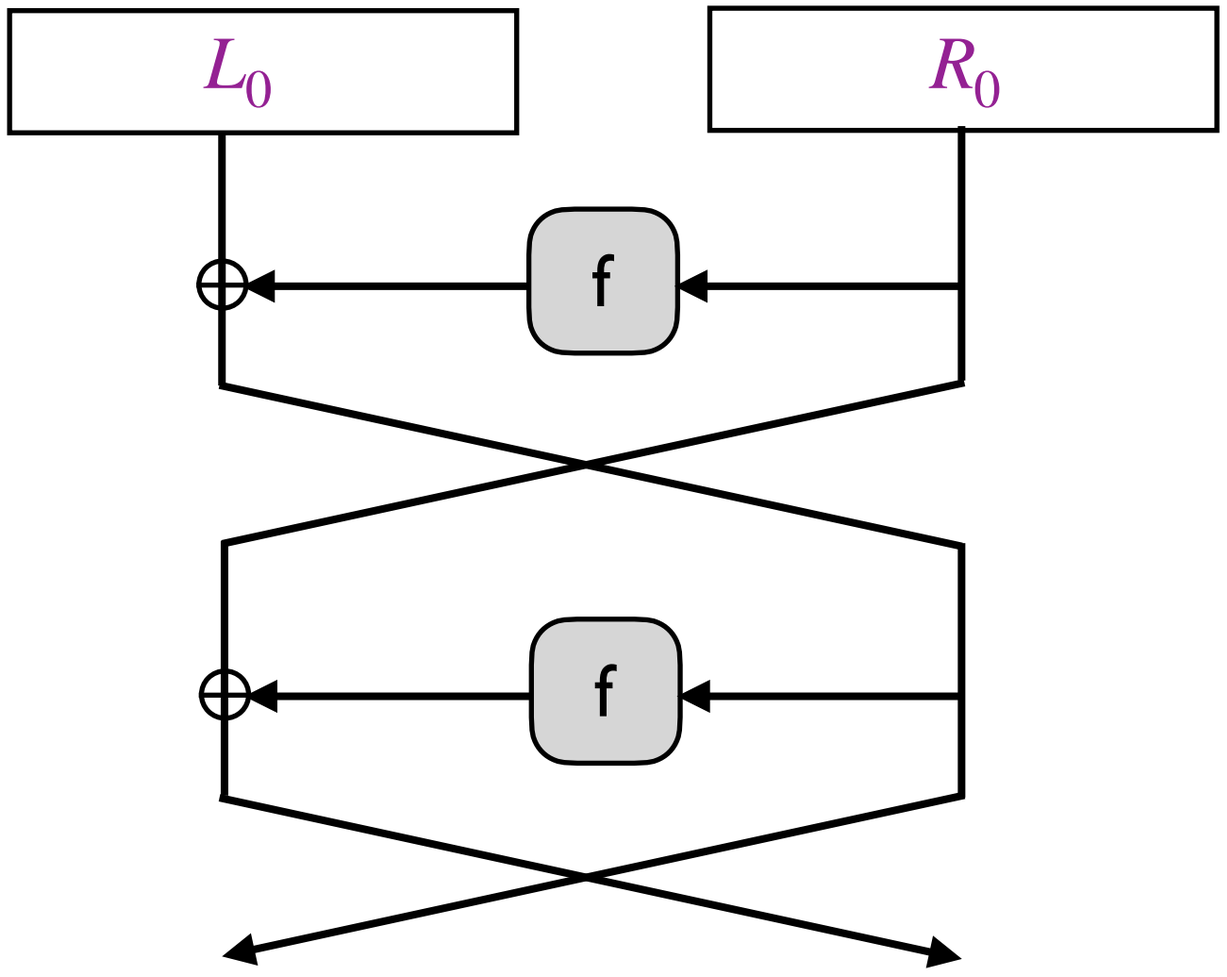


This class is being recorded

Avalanche Effect for Feistel Network

The multiple rounds of a Feistal network facilitate the avalanche effect provided the f functions magnify changes in their inputs.

Since f has the property that a single bit flipped in the input leads to 2 or more bit flips in the output, then a bit flip in R_0 leads to two bit flips in R_1, which in turn leads to 4 bit flips in R_2, and so on. 16 rounds is plenty to make sure a single bit change in the input leads to totally different outputs.



This class is being recorded

Breaking DES

DES was created in the 1970s, and at that time the key length of 56 bits was adequate (although still a bit short). However, by the 1990s, it was possible to break DES via a brute force attack on specialized hardware.

> (There are faster theoretical attacks but they use too many chosen plaintexts to be practical.)

Since DES was widespread, this problem has been addressed with 3DES, which simply runs DES 3 times with different keys.

> There is a "meet-in-the-middle" attack against 3DES which uses time more like 2^{2*56} rather than the brute force time of 2^{3*56}. However, this is still secure in practice.

However, AES is a more modern encryption protocol and is better.

This class is being recorded