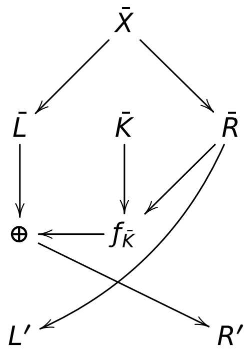
Exercise 1

> *For a bit string X, let $\bar{X}$ denote the complement of $\bar{X}$, that is, the string obtained by flipping all bits in X. Show that for any plaintext block X and DES key K, it holds that if $Y = DES_K(X)$, then $\bar{Y} = DES_{\bar{K}}(\bar{X})$.*

DES encryption is a composition of a number of functions. If we can show that each of these functions has the property $\overline{F_K(X)} = F_{\bar{K}}(\bar{X})$ then we can infer that the whole encryption function will have the same property.

We work through the diagram of a single cycle in $DES_{\bar{K}}(\bar{X})$:



Clearly the projections of $\bar{X}$ onto the left and right halves will maintain the complement, as will the switching of the halves at the end. It's known as well that $\overline{A \oplus B} = \bar{A} \oplus \bar{B}$. So all that is left to show is that $f_{\bar{K}}(\bar{X}) = \overline{f_K(X)}$. From the definition:

$$f_{\bar{K}}(\bar{R}) = P(S(\bar{K} \oplus E(\bar{R}))).$$

The first function we apply is E, which copies the input, duplicating a few select bits. So if a bit is flipped before being input, it will be copied and duplicated the same way. So we have $E(\bar{R}) = \overline{E(R)}$.

$$f_{\bar{K}}(\bar{R}) = P(S(\bar{K} \oplus \overline{E(R)})).$$

And, as already discussed, the operation $\oplus$ maintains the complement;

$$f_{\bar{K}}(\bar{R}) = P(S(\overline{K \oplus E(R)})).$$

Finally, we see that S, P behave nicely with complements. The division of a bitstring into blocks, and the permutation of the blocks both do nothing to the bits themselves, only to their ordering.

$$f_{\bar{K}}(\bar{R}) = \overline{P(S(K \oplus E(R)))} = \overline{f_K(R)}.$$

And so we have our desired result.

Exercise 2

> *Also show that, given a chosen plaintext attack where you may ask for the encryption of 2 plaintexts, you can use this property to do exhaustive key search in half the time it would normally take.*

Suppose the oracle chooses some key K. Choose any arbitrary plaintext X, and request the encryptions of X and X'. Then begin brute force encrypting X with each K_i, being sure to keep track and never try $\bar{K}_i$ for any i we previously checked. After encrypting, we check each ciphertext C_i against $E_K(X)$ and $\bar{C}_i$ against $\overline{E_K(X)}$. This cuts down half the keys needed to try.