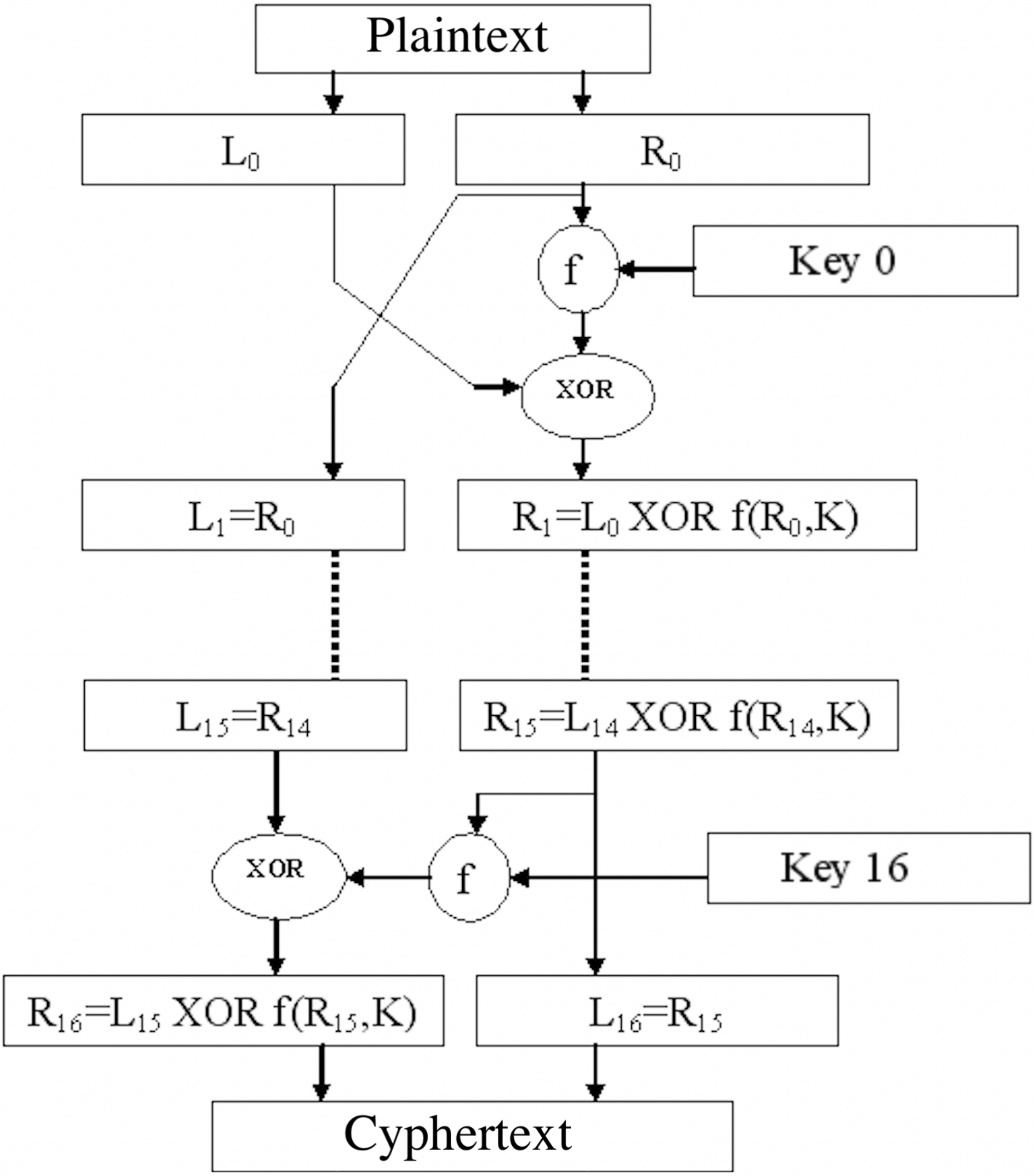
Exercise 6.2

Task

For a bit string X, let $\overline{X}$ denote the complement of 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 $\overline{Y} = DES_{\overline{K}}(\overline{X})$. 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.

Solution

Here's what the DES algorithm procedure looks like:



We can see that it is built upon the $\oplus$ and $f(R,K) = P(S(K \oplus E(R)))$ functions. If we can show that these two functions preserve the complemented property of the input, we'll get that the whole algorithm (which is just applying the above-mentioned functions multiple times) preserves the same property.

To start off, we can see that to get $L_1 = R_0$. In the complemented world, that would be $L_1 = \overline{R_0}$, which means that L_i preserves the complemented property as long as the R_{i-1} has it. So we just need to show that R_i preserves the property.

To get from R_0 to R_1 we apply the following function: $R_1 = R_0 \oplus f(R_0, K_0)$. Which in the complemented world is $R_1 = \overline{R_0} \oplus f(\overline{R_0}, \overline{K_0})$.

Let's start with $a \oplus b = c$. From the definition we pretty much get the property that if we take $\overline{a} \oplus \overline{b} = \overline{c}$. One can easily see that by looking at the table of all possible inputs

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

$\overline{a}$	$\overline{b}$	$\overline{a} \oplus \overline{b}$
1	1	0
1	0	1
0	1	1
0	0	0

a	$\overline{b}$	$a \oplus \overline{b}$
0	1	1
0	0	0
1	1	0
1	0	1

So we get that $a \oplus b = \overline{a} \oplus \overline{b}$. And $a \oplus \overline{b} = \overline{a \oplus b}$

As we already have $a = \overline{R_0}$ we need only show that $b = f(\overline{R_0}, \overline{K_0}) = f(R_0, K_0)$, and we'll get that in the $a \oplus \overline{b}$ situation, which gives us $a \oplus b$ with flipped bits.

$f(\overline{R}, \overline{K}) = P(S(\overline{K} \oplus E(\overline{R})))$

$E(x)$ is a function, which deterministically permutes the bits of x and deterministically copies half of its bits to a random index in the output (by inserting them, and not replacing existing bits). Therefore, $E(x)$ preserves the complemented property of its argument.

As we saw earlier, $a \oplus b = \overline{a} \oplus \overline{b}$ and we know that $E(x)$ preserves the complemented property. Therefore, $S(\overline{K} \oplus E(\overline{R})) = S(K \oplus E(R))$.

The last step of f is $P(x)$, but it's irrelevant because we know that it's argument $S(\ldots)$ is the same whether we are using the original inputs or the complemented ones.

So finally we get $R_1 = \overline{L_0} \oplus f(R_0, K_0)$, which is R_1 with flipped bits.

The rest of the algorithm is just repeating the above process 16 times. The last step is a little different, but it is essentially the same, except it applies the f and $\oplus$ functions to L_i instead of R_i, which doesn't change the way we reason about it.

Hence, we've shown that for a bit string X and $Y = DES_K(X)$, we have that $\overline{Y} = DES_{\overline{K}}(\overline{X})$.

To address the 2nd point

> 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.

Let's say we pass plaintext X and $\overline{X}$ to the oracle and received back the ciphertexts Y and $\overline{Y}$. Then to perform an exhaustive key search we'd have to iterate over all keys and check if encrypting X under each key will give us Y. However, we can iterate over half of the keys, such that no 2 keys are compliment of each other. On each step if we don't get either Y or $\overline{Y}$, we discard the key (in essence also discarding its compliment). This way we will do a brute-force key search in half the time.