Introduction to Modern Cryptography
Problem Set 1

Due date. Feb 9, 2025 (by 11:59pm). Please turn in a PDF document, and please try to typeset it in LaTeX. (I'll provide the source for this assignment, so you have a starting point.) Since you can work in groups of up to size three, please put all names on your solution set, and please include the last name of at least one of your group members in the filename.

Notational reminders. Recall that when X, Y are bitstrings, $X \,\|\, Y$ is the concatenation of the two. When $|X| = |Y|$, we write $X \oplus Y$ for their bitwise exclusive-or. When X is any object (e.g., a set, an integer), we write $\langle X \rangle$ to denote an unambiguous encoding of X as a bitstring; we write $\langle X \rangle_b$ to denote the encoding of X as a b-bit string. The set $\mathsf{Perm}(n)$ contains all possible permutations $\pi\colon \{0,1\}^n \to \{0,1\}^n$, the set $\mathsf{Func}(n,m)$ contains all possible functions $\rho\colon \{0,1\}^n \to \{0,1\}^m$, and the set $\mathsf{BC}(k,n)$ contains all blockciphers with k-bit keys and n-bit blocksize.

Problem 1. Let $F\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Define $E\colon \{0,1\}^{n+k} \times \{0,1\}^n \to \{0,1\}^n$ as $E_{K2 \,\|\, K1}(X) = F_{K1}(X) \oplus K2$. (To be clear, $|K1| = k, |K2| = n$.) Prove that if F is a secure PRP, then so is E. In other words, xoring in an additional n-bit secret doesn't "destroy" the PRP-security provided by F.

To get you started, here's a reminder of how to approach this proof. Given an adversary A that attacks the PRP-security of E (i.e., it tries to distinguish between $E_{K2 \,\|\, K1}(\cdot)$ for uniform, secret $K2 \,\|\, K1$, and $\pi(\cdot)$ uniformly sampled from $\mathsf{Perm}(n)$), build an adversary B that attacks the PRP-security of F (i.e., distinguishing between $F_{L1}(\cdot)$ for uniform, secret $L1$, and π' uniformly sampled from $\mathsf{Perm}(n)$). Carefully show that the PRP-advantage of B upperbounds the PRP-advantage of A, by analyzing the probabilility that B "wins" its experiment. When building your adversary B, try to make it as simple as possible, and as parsimonious as possible with respect to its resources. Also, it will help to convince yourself of the following two facts: for any fixed $V \in \{0,1\}^n$ and any $\pi \in \mathsf{Perm}(n)$, $f(X) = \pi(X) \oplus V$ is a permutation; and for any fixed $V \in \{0,1\}^n$ and $\pi' \in \mathsf{Perm}(n)$, when $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ we have $\Pr\left[f = \pi'\right] = 1/(2^n!)$. (More generally, the composition of a random permutation with any fixed permutation is itself a random permutation.)

Finally, try to write a nice theorem statement to encapsulate your result. Something like this:

Theorem 1 *Fix integers $n, k > 0$ and Let $F\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Define $E\colon \{0,1\}^{n+k} \times \{0,1\}^n \to \{0,1\}^n$ as $E_{K2 \,\|\, K1}(X) = F_{K1}(X) \oplus K2$. Let A be a PRP-adversary (for attacking E) that has time complexity t and asks at most q oracle queries. Then there exists an adversary B, explicitly given in the proof of this theorem, such that*

$$\mathbf{Adv}_E^{\mathrm{prp}}(A) \leq [???]\mathbf{Adv}_F^{\mathrm{prp}}(B) + [???]$$

and where B has time complexity $t' = [???]$ and asks $q' = [???]$ queries. ♢

```
$\mathsf{Exp}_E^{\mathrm{kr-ic}}(A)$:
    $K \xleftarrow{\$} \{0,1\}^{2n}$
    $F \xleftarrow{\$} \mathsf{BC}(n,n)$
    $K' \xleftarrow{\$} A^{\mathcal{O}(\cdot),\mathrm{IC}(\cdot,\cdot)}$
    Ret $[K' = K]$

oracle $\mathcal{O}(X)$:
    Ret $E_K(X)$

oracle $\mathrm{IC}(L,X)$:
    Ret $F_L(X)$
```

Figure 1: The key-recovery notion for function family $E\colon \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$, in the ideal cipher model for some (related) blockcipher F. In the current problem, E is specifically defined as $E_{K2\,\|\,K1}(X) = F_{K1}(X) \oplus K2$, so a single call $\mathcal{O}(X)$ effectively determines one value of the table for F_{K1}, in the lazy-sampling sense. (But *only* for that specific table, not any of the other $F_{K1'}$ tables for $K1' \neq K1$.)

You should be able to fill in the missing values as a result of your analysis.

Problem 2. The previous problem says, informally, that xoring in an additional secret key $K2$ doesn't "destroy" the PRP-security provided by F. In this problem, you'll show that, while this is true, it does not make key-recovery attacks harder. (So, really, what is gained?)

To this end, reconsider the blockcipher $E_{K2\,\|\,K1}(X) = F_{K1}(X) \oplus K2$ from the previous problem, with the specific setting of $k = n$. You might expect that, since both $K1, K2$ are random and secret, then it should take something like 2^{2n} work to recover the full $2n$-bit key. Not so, and you're going to show why. In fact, we will show it even if $F \xleftarrow{\$} \mathrm{BC}(n,n)$, i.e., if F is an *ideal cipher*. Note that $F \xleftarrow{\$} \mathrm{BC}(n,n)$ is equivalent to the following way of "building" F: $\mathsf{for}\ L1 \in \{0,1\}^n\ \mathsf{do}\ F_{L1} \xleftarrow{\$} \mathrm{Perm}(n)$. Thus, an ideal cipher *is* a uniformly random permutation for every key, as opposed to being a blockcipher that "looks like" a random permutation (on average, over the keyspace) to efficient tests[1].

Figure 1 gives the key-recovery experiment *in the ideal cipher model.* The adversary is challenged to recover the $2n$-bit key K of the constructed blockcipher E. As in the standard key-recovery experiment, the adversary can see input-output pairs $X, E_K(X)$ by asking a query X to the $\mathcal{O}$-oracle. But it also has an oracle that returns $F_L(X)$ for keys L and inputs X of its choosing[2]. Show that the *full* $2n$-bit key K can be recovered (with good probability) with something like 2^n calls to the IC-oracle — as opposed to 2^{2n} — and a small constant number of calls (say, 2) to the $\mathcal{O}$-oracle. As a hint, observe that for any $L1, L2, X$ of your choosing, you can compute $E_{L2\,\|\,L1}(X) = F_{L1}(X) \oplus L2$ with a single query $\mathrm{IC}(L1, X)$.

[1] Which would make it a good PRP.

[2] This is abstracting the fact that, in the real world, the blockcipher F would be instantiated with a real blockipher, and real blockciphers (e.g., AES) have public descriptions. Thus, in real world key-recovery attacks, one can compute $F_L(X)$ locally for any valid L, X. Having an explicit oracle for $\mathrm{IC}(L,X) = F_L(X)$ gives a way to account for each local computation in the complexity of the attack.