Introduction to Modern Cryptography  
Problem Set 1 (Solutions)

Problem 1. Fix $k,n > 0$ and let $E: \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Define $F: \{0,1\}^{k+n} \times \{0,1\}^n \to \{0,1\}^n$ by $F(K_1\|K_2, X) = F_{K_1,K_2}(X) = E_{K_1}(X \oplus K_2)$, where $K_1 \in \{0,1\}^k$ and $K_2 \in \{0,1\}^n$. Notice that $F$ is a blockcipher with a $(k+n)$-bit key and an $n$-bit blocksize.

Modelling $E$ as an ideal cipher, what is the complexity of an exhaustive key-search attack on $F$? Can you recover the key $K_1 \parallel K_2$ (with overwhelming probability) using significantly fewer than $2^{k+n}$ local computations of $F$?

Solution. Let’s say that you know an input output pair $(X,Y = F_{K_1,K_2}(X))$. In particular, you know that $Y = E_{K_1}(X \oplus K_2)$, which is equivalent to saying that $E_{K_1}^{-1}(Y) = X \oplus K_2$. So do $O(2^k)$ local computations of $Z = E_{K_1}^{-1}(Y)$, one for each $L_1 \in \{0,1\}^k$. Note that for each $L_1$ there is exactly one possible second key $L_2$ to consider, namely $L_2 = Z \oplus X$. That is, the key pair $L_1,L_2$ maps $X$ to $Y$. So now you have a list of length $2^k$ of pairs $(L_1,L_2)$, and you know that one of is the correct pair, i.e., $(L_1,L_2) = (K_1,K_2)$. Which one? You can almost certainly tell with a second input-output pair $(U,V = F_{K_1,K_2}(U))$. Namely, for every $(L_1,L_2)$ in the list, compute $E_{L_1}(U \oplus L_2)$ and compare the result to $V$. This requires another $O(2^k)$ local computations. (If there still exists more than one candidate $(L_1,L_2)$ after this step, the total number will be very small, and a third input-output pair will find the correct one.) Thus you can find the full $(k+n)$-bit key with much less than $2^{k+n}$ local computations.

Note that this construction actually came up as part of the academic community’s response to U.S. export control laws in the 90s, which restricted the key-length of DES implementations in export versions of early web browsers. Although this construction doesn’t really give you any additional “effective” key-length, this one does $E_{K_1}(X \oplus K_2) \oplus K_3$. Check out the papers on the “DESX” construction.

Problem 2. BR notes, Chapter 3: problem1.  
Carefully describe your adversary, and give the PRP advantage of it. (Please use the notation we use in class when describing the adversary, etc.)

Solution. This problem a common one in cryptography, generally known as “domain extension”. In this case, how can we securely extend the domain of an $n$-bit blockcipher $E$ to $2n$-bit inputs? We are asked to show that $E'_K(x \parallel x') = E_K(x) \parallel E_K(x \oplus x')$ is not a secure way to do this, namely because the PRP-security of $E$ will not be maintained by this construction. Consider the following adversary $D$:

\footnote{https://cseweb.ucsd.edu/~mihir/cse207/w-prf.pdf}
We now analyze the advantage this adversary achieves. First, we note that
\[ \Pr[\text{Exp}^\text{prp}_{E'}(D) = 1 \mid b = 1] = 1 \]
where \( b \) is the challenge bit in the PRP experiment. What about when \( b = 0 \), i.e., the oracle \( O = \pi \) where \( \pi \) is a random permutation over \( \{0, 1\}^{2n} \)? In this case we have
\[ \Pr[\text{Exp}^\text{prp}_{E'}(D) = 1 \mid b = 0] = 1 - \Pr[\text{Exp}^\text{prp}_{E'}(D) = 0 \mid b = 0] = 1 - \Pr[\text{Exp}^\text{prp}_{E'}(D) : D \Rightarrow 1 \mid b = 0] = 1 - \frac{1}{2^n} \]
where, recall the notation \( D \Rightarrow 1 \) reads “\( D \) outputs 1”, and the probability of this event is over the coins used in the execution of \( \text{Exp}^\text{prp}_{E'}(D) \). When \( b = 0 \) the probability that \( D \Rightarrow 1 \) is exactly the probability that \( Y_\ell = Y_r \), and the chance of that is \( 1/2^n \) since there are exactly \( 2^n \) strings (of length \( 2n \)) such that the first and last halves are the same.

so by substituting these into the definition of PRP-advantage, we get \( \text{Adv}^\text{prp}_{E'}(D) = 1 - 2^{-n} \geq 1/2 \).

**Problem 3.** BR notes, Chapter 3: problem 6. Here you must give a security reduction! Since this is your first one, I’ll get you started. Given an adversary \( A \) that attacks the PRP-security of \( E^{(2)} \), build an adversary \( B \) that attacks the PRP-security of \( E \). Carefully show that the PRP-advantage of \( B \) upperbounds the PRP-advantage of \( A \), by analyzing the probability that \( B \) “wins” its experiment. When building your adversary \( B \), try to make it as simple as possible, and as stingy as possible with respect to its resources. Finally, write a nice theorem statement to encapsulate your result.

**Solution.** Adversary \( A \) expects an oracle that either implements \( E^{(2)}_K \) for a random key \( K \), or a random permutation \( \pi \). We will construct \( B \) to simulate this as closely as possible. Here it is:

<table>
<thead>
<tr>
<th>adversary ( B^{O} ):</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K1 \leftarrow {0, 1}^k )</td>
</tr>
<tr>
<td>When ( A ) queries ( X ):</td>
</tr>
<tr>
<td>\quad Return ( E_{K1}(O(X)) ) to ( A )</td>
</tr>
<tr>
<td>When ( A ) outputs bit ( b )</td>
</tr>
<tr>
<td>\quad Return ( b )</td>
</tr>
</tbody>
</table>

Notice that we do not return \( O(O(X)) \) to \( A \). This would be wrong in the case that \( O = E_K \), since then \( B \) would simulate \( E_K(E_K(X)) \) which has a single key. Let’s compute the PRP-advantage of this \( B \). What matters is \( \Pr[\text{Exp}^\text{prp}_{E'}(B) = 1] \), and we’ll do the usual trick of conditioning this on the value of the challenge bit \( b \) in \( \text{Exp}^\text{prp}_{E'}(\cdot) \). First, we observe that when \( b = 1 \) we perfectly simulate \( E^{(2)} \) when responding to \( A \)’s oracle queries. Thus \( \Pr[\text{Exp}^\text{prp}_{E'}(B) = 1 \mid b = 1] = \Pr[\text{Exp}^\text{prp}_{E^{(2)}}(A) = 1 \mid d = 1] \) where \( d \) is the challenge bit in \( \text{Exp}^\text{prp}_{E^{(2)}}(\cdot) \). On the other hand, when \( b = 0 \) we respond to query \( X \) with \( E_{K2}(\pi(X)) \). But the permutation \( \pi'(\cdot) = E_{K2}(\pi(\cdot)) \) is uniformly
random over Perm(ℓ), because π is. So Pr [Exp_{E}^{\text{prp}}(B) = 1 \mid b = 0] = Pr [Exp_{E(2)}^{\text{prp}}(A) = 1 \mid d = 0].

So we conclude that

\[
\text{Adv}_{E}^{\text{prp}}(B) = 2 \Pr [\text{Exp}_{E}^{\text{prp}}(B) = 1] - 1
\]

\[
= 2 \left( .5 \Pr [\text{Exp}_{E(2)}^{\text{prp}}(A) = 1 \mid d = 1] + .5 \Pr [\text{Exp}_{E(2)}^{\text{prp}}(A) = 1 \mid d = 0] \right) - 1
\]

\[
= 2 \left( \Pr [\text{Exp}_{E(2)}^{\text{prp}}(A) = 1] \right) - 1
\]

\[
= \text{Adv}_{E(2)}^{\text{prp}}(A)
\]

Let’s give a nice theorem statement to encapsulate the result. In the following, define notation

\[
\text{Time}(E)
\]

to be the worst-case running time, over (K, X) of some fixed, implicit implementation of E.

**Theorem 1** Fix k, ℓ > 0 and let E: \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell be a blockcipher. Define blockcipher E(2): \{0, 1\}^{2k} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell by E_{K_1 \parallel K_2}(x) = E_{K_1}(E_{K_2}(x)). Let A be a PRP-adversary for E(2) that has time-complexity t and asks q queries to its oracle. Then there exists an adversary B, constructed above, that has time-complexity at most t + q\text{Time}(E) and asks q queries, such that

\[
\text{Adv}_{E}^{\text{prp}}(B) = \text{Adv}_{E(2)}^{\text{prp}}(A).
\]