Problem 1
Let E : $\{0,1\}$ k $\times$ $\{0,1\}$ n $\rightarrow$ $\{0,1\}$ n be a good PRF. Break the PRF security of $F_K(X) = E_K(X)$ || $E_K(E_K(X))$ using a few queries and analyze the advantage of your attack.

Solution

To break this PRF security, we want to determine if we are in the real or random world. Given that $F_K(X) = E_K(X)$ || $E_K(E_K(X))$, the completion of $F_K(X)$ would produce C_1. Query 0^n to produce C_1, and use the beginning half of C_1 (where X is encrypted before concatenation) calling it A (Where $A = E_k(X)$) to produce C_2. Broken down, the C_2 function (when called with A) would read: $E_k(E_k(X))||E_k(E_k(E_k(X)))$. The first part of C_2 where $E_k(E_k(X))$ may be called B. This is compared with the second half of C_1, called A' (thus A' = second half of $C_1 = E_K(E_K(X))$), and if $A' = B$ then return 1 and assume we are in the real world. Otherwise, return 0 and assume we are in the random world. The advantage here is as follows: $Adv_E^{PRF}(A) = 1 - \frac{2}{2^n}$

This is because we have probability of 1 to be in the real world and the probability of $\frac{1}{2^n}$ to be in the random world where n is half the length of the ciphertext, since A' will always equal to B given our query in the real world but only have $\frac{1}{2^n}$ probability for $A' = B$ in the random world. Through this, we can use the queries 0^n and A (produced during the generation of C_1) to break PRF security and determine if we are in the real or random world.

This is tricky, however, because in the random world you make 2 queries. They can be independent random strings (different) or the same random strings and there can be a case that the two random strings are equal (very small chance). Thus we should account for the small chance that in the random world the output is the same as input. Being union bound, the probability for the random world $\Rightarrow$ 1 should actually be $\frac{1}{2^n}$ + another small probability, which in this case is $\frac{1}{2^n}$. Adding these together brings us to the total probability we are in the random world $\Rightarrow$ 1 to be $\frac{2}{2^n}$.

Problem 2
Figure 2.1 below illustrates another way to do ciphertext stealing on CBC with a blockcipher $E : \{0,1\}$ $k\{0,1\}$ $n \rightarrow \{0,1\}$ n . This variant CBC-S appears in textbooks but is actually *insecure*. Break the real-or-random security of CBC-S using a few queries and analyze the advantage of your attack.

Solution

Since the adversary can use plaintext of any length and ciphertext stealing only happens if the plaintext length is less than n blocks, this determines why the last block in the CBC-S is simply $C_4 = E_k(M)$ if size of plaintext is equal to n blocks. This gives the CBC-S some vulnerabilities similar to ECB. Since C_0 is randomly generated, we want to look at C_4 to break real or random security in this case.

To accomplish this, we query $0^n0^n0^n0^n$ **twice**, where n is the length of each block and compare the first C_4 with the second C_4'. If $C_4 = C_4'$ then return 1 and assume we are in the real world. Otherwise, return 0 and assume we are in the random world. The advantage of this attack is $Adv_E^{RR}(A) = 1 - \frac{1}{2^n}$. This is because the real world has a probability of 1 while the random world has the probability of $\frac{1}{2^n}$. C_4' has probability $\frac{1}{2^n}$ to be equal to C_4 given n here is the length of the block.

Problem 3
CBC-Chain is a stateful blockcipher-based mode of operation that was actually used in SSH. To encrypt, we use CBC with an IV that is the last ciphertext block produced from the prior encryption. Initially, the IV is a random string. Give an attack that breaks the left-or-right security of CBC-Chain[E] using a few queries, and analyze its advantage.

Solution

To break left or right security, we query 2 random messages M and M' of block size 1 to get ciphertexts C_0C_1. We then query C_1 and a random string to give C_1C_2. Following this, we finally query C_2 and a random string to output C_2C_3. To break the left right security, compare C_2 and C_3 where if $C_2 = C_3$ return 1 and assume we are in the left room which took $C1$ and $C2$ as queries. Otherwise, return 0 and assume we are in the right room which took the random strings as queries.

This works because the messages we query aren't fully random. The left side was determined to be C_1 or C_2, thus allowing us to determine whether we are in the left or right room. If it queried the random

string, then C_1 and then C_2, $C_0C_1 = C_0||E_k(M \oplus C_0)$ when M is random, $C_1C_2 = C_1||E_k(C_1 \oplus C_1)$ when $M = C_1$, and $C_2C_3 = C_2||E_k(C_2 \oplus C_2)$ when $M = C_2$. Thus if we query a random string, C_1 and then C_2, it will always be the case given our query that $C_2 = C_3$. With this in mind, there is only $\frac{1}{2^n}$ probability of $C_2 = C_3$ if we query 3 random strings. This attack has an advantage of: $Adv_E^{lr}(A) = 1 - \frac{1}{2^n}$. This is the case because $\Pr[\text{left} \Rightarrow 1] = 1$ and $\Pr[\text{right} \Rightarrow 1] = \frac{1}{2^n}$, where n is the length of C_2 and C_3.