1. (50 points) The CBC-MAC construction we learned in class is sequential: one has to finish a blockcipher call before one can proceed to the next one. In modern platforms, this sequential behavior is undesirable. Here’s an attempt to build a fully parallelizable MAC, which we’ll call XMAC. Let’s say we have a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$. Fix a number $r < n$. Write $[i]$ to denote an $r$-bit encoding of integer $i \leq 2^r$. Suppose that you have a message $M = P_1 \cdots P_m$, where $m \leq 2^r$ and each $|P_i| = n - r$. To sign it under key $K$, compute

$$T = E_K([1]||P_1) \oplus \cdots \oplus E_K([m]||P_m).$$

In practice $n = 128$, and one can choose $r = 32$. (Thus your message can be at most 64 GB.) In that case, XMAC is several times faster than CBC-MAC. Unfortunately, XMAC is insecure. Break its MAC security using 3 queries and analyze your advantage. Each message in your queries and output should have just two $(n-r)$-bit blocks.

2. (50 points) Let $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a good PRF. For every $X_1, X_2 \in \{0,1\}^{n-1}$, let $F_K(X_1||X_2) = E_K(0||X_1)||E_K(1||X_2)$. Is $F$ a good MAC? If you say no, give an attack and analyze its advantage. If you say yes, you need to give a game-based proof to justify the security.

3. (50 points) To avoid extension attacks on CBCMAC, one might be tempted to define the following variant CBCMAC2 with a randomized IV. Let $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be the underlying blockcipher. To MAC a message $M = X_1 \cdots X_m$ where each $|X_i| = n$, we pick $X_0 \leftarrow \{0,1\}^n$, run the ordinary CBCMAC on $(X_0 \oplus X_1)||X_2 \cdots X_m$ to obtain a tag $T$, and then output $(X_0, T)$. On input $(X_0, T)$, the verification algorithm tests if $T$ is the same as the CBCMAC of $(X_0 \oplus X_1)||X_2 \cdots X_m$.

Break CBCMAC2 using a few queries, and analyze the advantage of your attack.