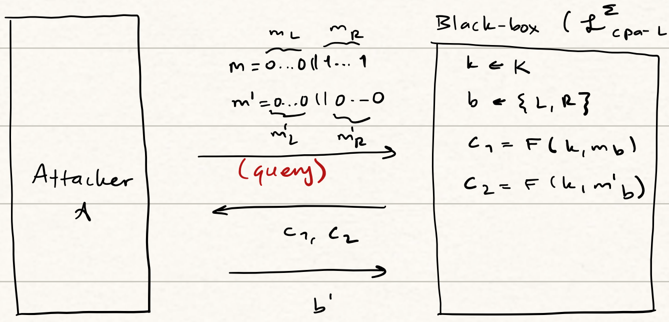
① 1. There are two messages $m = \underbrace{0\ldots0}_{B} \| \underbrace{1\ldots1}_{B}$ and $m' = \underbrace{0\ldots0}_{B} \| \underbrace{0\ldots0}_{B}$, each consisting of two blocks. In ECB, each block is encrypted with the same key. So, encryption of m' will be duplicate of the same block, whereas, encryption of m will be concatenation of distinct blocks. (Note that block ciphers/PRPs are invertible.)

2.



Recall: $\mathcal{L}^{\Sigma}_{cpa-L}$ encrypts the left block of the messages, similarly $\mathcal{L}^{\Sigma}_{cpa-R}$ does the right

Strategy: $b' = \begin{cases} L & \text{if } c_1 = c_2 \\ R & \text{if } c_1 \neq c_2 \end{cases}$

$$\Pr[\mathcal{A} \text{ wins}] = \Pr[b=b'] = \Pr[b=b' \cap b=L] + \Pr[b=b' \cap b=R]$$

$$= \underbrace{\Pr[b=b' \mid b=L]}_{1} \cdot \underbrace{\Pr[b=L]}_{\frac{1}{2}} + \underbrace{\Pr[b=b' \mid b=R]}_{1} \cdot \underbrace{\Pr[b=R]}_{\frac{1}{2}} = 1$$

② 1. Assume we encrypt without adding whole block of length B for padding. When we decrypt the encrypted message and see 01 at the end, we will think 1 represents the no. of bytes added to the original message m due to the designated padding. Then we will ignore 01 and treat the rest as the original message, which is not the case.

2. Notice that number of bytes to be added, namely b, is encoded as a single byte And an upper bound for b is B, i.e., $b \leq B$. So, B should fit into a single byte.. One byte is 8 bits and greatest number that can be written with 8 bits is 255. So, $B \leq 255$.

3. Let $m = m_1 \| m_2 \| \dots \| m_t$ where $|m_i| = B$ for $i = 1, \dots, t-1$ and $0 < |m_t| < B$, i.e., m is not a multiple of the block length.

```
r ← {0,1}^B
c_0 := r
for i = 1 to t-1:
    c_i := F(k, r) ⊕ m_i
    r := r + 1 % 2^B
c_t := F(k, r)[0:(|m_t| - 1)] ⊕ m_t
return c_0 || c_1 || ... || c_t
```

} exact steps of CTR

} here we chop $F(k,r)$ to do bitwise XOR with m_t
} as a result c_t has the same size with m_t and without padding we can encode the info. about size of the message m in the ciphertext

③ 1. Assume adversary $\mathcal{A}$ calls Sample($R \subset \{0,1\}^d$) and gets output r. If $r \in R$, adv. $\mathcal{A}$ knows that he talked to $L_{\text{GenSample}}$ library. If $r \in R$, adv. $\mathcal{A}$ cannot conclude something better than a random guess. More precisely, algo. $\mathcal{A}$:

```
r ← Sample(R ⊂ {0,1}^d)
if r ∈ R:
    output 1 (succeeding in distinguishing)
else:
    output 0
```

2. Observe that when $\mathcal{A}$ talks to $L_{\text{GenSampleIgnore}}$, $\mathcal{A}$ never outputs 1, i.e., $\Pr[\mathcal{A} \diamond L_{\text{GenSampleIgnore}} \Rightarrow 1] = 0$. On the other hand, when $\mathcal{A}$ talks to $L_{\text{GenSample}}$, $\mathcal{A}$ outputs 1 if and only if $r \in R$. So, outputs of $L_{\text{GenSample}}$ for which $\mathcal{A}$ will output 1 with the same probability as when $\mathcal{A}$ talks to $L_{\text{GenSampleIgnore}}$ are $r \notin R$.

And $\Pr[\mathcal{A} \diamond L_{\text{GenSample}} \Rightarrow 1] = \Pr[L_{\text{GenSample}} \text{ outputs } r \in R] = \frac{|R|}{2^d}$.

$\text{Adv}(\mathcal{A}) = |\Pr[\mathcal{A} \diamond L_{\text{GenSampleIgnore}} \Rightarrow 1] - \Pr[\mathcal{A} \diamond L_{\text{GenSample}} \Rightarrow 1]| = \frac{|R|}{2^d}$.

3. Let $r_1 \leftarrow \text{Sample}(R_1 \subset \{0,1\}^d)$ is the output of the 1st call and, similarly, $r_2 \leftarrow \text{Sample}(R_2 \subset \{0,1\}^d)$ is the output of the 2nd call. By part (2), we know that $\mathcal{A}$ can only distinguish if it talks to $L_{\text{GenSample}}$ and gets $r_1 \in R_1$ or $r_2 \in R_2$. So, distinguishing prob. is maximal when $\mathcal{A}$ talks to $L_{\text{GenSample}}$, and distinguishing prob. is $\Pr[r_1 \in R_1 \text{ or } r_2 \in R_2]$

$= 1 - \Pr[r_1 \notin R_1 \text{ and } r_2 \notin R_2] = 1 - \Pr[r_1 \notin R_1]\Pr[r_2 \notin R_2] = 1 - \left(1 - \frac{|R_1|}{2^d}\right)\left(1 - \frac{|R_2|}{2^d}\right)$.

(Recall: A and B are independent $\Leftrightarrow \Pr(A)\Pr(B) = \Pr(A \cap B)$)

4. Similar to (3), maximal prob. of distinguishing/success with q calls is $1-\prod_{i=1}^{q}\left(1-\frac{|R_i|}{2^d}\right)$.

5. $\mathcal{A}$ has polynomial runtime, therefore can make poly-many calls, i.e., $q \in poly(d)$. Also, in each call $\mathcal{A}$ generates a set R where generating each item takes 1 unit of time, therefore size of the set R in each call is $poly(d)$. Then prob. success $\underset{\substack{\downarrow \\ \text{by part} \\ (4)}}{\leq} \frac{q^2 \cdot |R|}{2^d} = \frac{poly(d)}{2^d}$ is negligible.