1. Let $q(n)$ be a polynomial such that, for any $k \leftarrow \text{Gen}(1^n)$, $|\text{Enc}_k(0)| \leq q(n)$. Such a polynomial must exist because the encryption algorithm must run in an amount of time polynomial in $n$. Since the maximum encrypted length of 0 is bounded by $q(n)$, we would like our adversary to choose $m_0 = 0$ and $m_1$ so that $m_1$ will always encrypt to a string of length $> q(n)$. If the adversary can do this, it becomes trivial to determine which message was encrypted: if the ciphertext has length $\leq q(n)$, then the adversary knows that the algorithm encrypted $m_0$, and otherwise, that the algorithm encrypted $m_1$. This allows the adversary to win the indistinguishability experiment with probability 1, so Definition 3.8 cannot be satisfied.

We now show that the adversary can pick such a message $m_1$. Consider all strings of length $q(n) + 2$. Since there are $2^{q(n)+2}$ such strings and fewer than $2^{q(n)+1}$ strings of length $\leq q(n)$, there must be some string $s \in \{0, 1\}^{q(n)+2}$ that can only encrypt to strings of length $> q(n)$. If the adversary chooses $m_1 = s$, then he can always win the indistinguishability experiment, so $\Pi$ cannot satisfy Definition 3.8, as desired.

2. (a) **Solution 1.** $G'$ is not necessarily a pseudorandom generator. We say this because assuming that $G'$ isn’t a pseudorandom generator doesn’t contradict the assumption that $G$ is one.

If a polynomial-time algorithm $D'$ can distinguish the output of $G'$ from a random string, then a distinguisher $D$ can do the same for the output of $G$ when the second half of the seed consists of all 0’s. However, with a seed of length $n$ (where $n$ is even), only a $2^{-n/2}$ fraction of the seeds will end in $n/2$ 0’s, so $D$ would only gain an advantage in an exponentially small number of cases.

Even if $D'$ is able to flawlessly detect strings generated by $G'$,

$$|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]|$$

would increase by at most $2 \cdot 2^{-n/2}$, which is negligibly small. This increase is not large enough to produce a contradiction, so $G'$ is not necessarily a pseudorandom generator, as desired.

**Solution 2.** $G'$ is not necessarily a pseudorandom generator. Let $G$ be any pseudorandom generator with $|G(s)| > 2 \cdot |s|$, and consider the generator $G^*$ defined as

$$G^*(s) := \begin{cases} 0^{G(s)} & \text{if } s \text{ ends with } \lfloor |s|/2 \rfloor \text{ 0's}, \\ G(s) & \text{otherwise}. \end{cases}$$

**Claim.** $G^*$ is a pseudorandom generator.
Proof. We proceed by contradiction. Assume that $D$ is a probabilistic polynomial-time distinguisher such that for all negligible functions $\text{negl}$ and sufficiently large $n$,

$$|\Pr[D(r) = 1] - \Pr[D(G^*(s)) = 1]| > \text{negl}(n),$$

where $r$ is chosen uniformly at random from $\{0, 1\}^{\ell(n)}$ and $s$ is chosen uniformly at random from $\{0, 1\}^n$.

Then, consider the same distinguisher $D$ applied to $G$. Since $G^*(s) \neq G(s)$ on at most a $2^{n/2+1}/2^n = 2^{-n/2+1}$ fraction of the possible values of $s$, we have that

$$|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]| > \text{negl}(n) - 2^{-n/2+1},$$

for any negligible function $\text{negl}$. However, since any negligible function $\text{negl}_1$ can be written as $\text{negl}_1 = \text{negl}_2 - 2^{-n/2+1}$ (this is easy to prove), this implies that

$$|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]| > \text{negl}(n)$$

for any negligible function $\text{negl}$, contradicting the assumption that $G$ is a pseudorandom generator. Thus, $G^*$ is a pseudorandom generator, as desired. \qed

Since $G^*$ is a pseudorandom generator, it is possible that $G = G^*$. In this case, $G'$ would trivially not be a pseudorandom generator because it would only output strings consisting of 0’s. Thus, $G'$ is not necessarily a pseudorandom generator, as desired.

(b) $G'$ is a pseudorandom generator. We will prove this by contradiction—if $G'$ is not a pseudorandom generator, then an adversary can use the knowledge of how to break $G'$ to break $G$. Intuitively, this should work because the output of $G'$ “looks like” the output of $G$ on a half-length seed. If we can distinguish a string generated by $G'$ from a random string, then we should be able to distinguish one generated by $G$ from a random string as well.

On any input of length $n$, let $G$ output a string of length $\ell(n)$, so $G'$ outputs a string of length $\ell(n/2)$. Then, assume for the sake of contradiction that $D$ is a probabilistic polynomial-time distinguisher such that for all negligible functions $\text{negl}$ and sufficiently large $n$,

$$|\Pr[D(r) = 1] - \Pr[D(G'(s)) = 1]| > \text{negl}(n),$$

where $r$ is chosen uniformly at random from $\{0, 1\}^{\ell(n/2)}$ and $s$ is chosen uniformly at random from $\{0, 1\}^n$. In other words, we assume that $D$ is an algorithm that can distinguish between random strings and strings generated by $G'$.

In particular, note that when the input is of length $2n$, we have that for any negligible function $\text{negl}$,

$$|\Pr[D(r) = 1] - \Pr[D(G'(s)) = 1]| > \text{negl}(2n),$$

where $r$ is chosen uniformly at random from $\{0, 1\}^{\ell(n)}$ and $s$ is chosen uniformly at random from $\{0, 1\}^{2n}$.  

2
Now, we will use $D$ as a distinguisher for $G$. Since $G$ is a pseudorandom generator, there must exist a negligible function $f$ such that

$$\left| \Pr[D(r) = 1] - \Pr[D(G(s)) = 1] \right| \leq f(n)$$

$$\iff \left| \Pr[D(r) = 1] - \Pr[D(G'(ss')) = 1] \right| \leq g(2n),$$

where $r$ is chosen uniformly at random from $\{0, 1\}^\ell(n)$, $s$ and $s'$ are each chosen uniformly at random from $\{0, 1\}^n$, and $g(n) := f(n/2)$. However, since $g$ is negligible (which is easily proven) and the concatenation of two strings of length $n$ chosen uniformly at random is equivalent to a string of length $2n$ chosen uniformly at random, this contradicts equation (1). Thus, $G'$ must be a pseudorandom generator, as desired.

3. Let $n$ be the block length of the encryption scheme, and for simplicity assume that there is only one block. The following proof trivially generalizes to $\ell$ blocks, but this assumption simplifies the notation.

Consider the adversary that first outputs the messages $m_0 = 0^n$ and $m_1 = 1^n$, and receives the challenge ciphertext $c = IV||c_1$ (where $||$ denotes concatenation). Then, the adversary queries the encryption oracle on the plaintext $m' = m_0 \oplus IV \oplus (IV + 1)$, receiving the ciphertext $c' = (IV + 1)||c'_1$. If $c$ is an encryption of $m_0$, then we should have $c_1 = c'_1$ since

$$c'_1 = F_k((IV + 1) \oplus m')$$
$$= F_k((IV + 1) \oplus m_0 \oplus IV \oplus (IV + 1))$$
$$= F_k(m_0 \oplus IV)$$
$$= c_1.$$

Similarly, if $c$ is an encryption of $m_1$, then with $1 - \text{negl}(n)$ probability (for some negligible function $\text{negl}$) we should have $c_1 \neq c'_1$. Otherwise, we would have that $F_k(m_0 \oplus IV) = F_k(m_1 \oplus IV)$ with nonnegligible probability, so $F_k$ would not be a pseudorandom permutation.

Thus, by outputting 0 if $c_1 = c'_1$ and 1 otherwise, this adversary will win the experiment with probability $\geq 1 - \text{negl}(n) > 1/2$, so the scheme is not CPA-secure, as desired.

4. Encryption can be parallelized easily for Output Feedback mode, but decryption can’t be. Both encryption and decryption can be parallelized easily for Counter mode.