Homework 2

1 Problem 1 (20 points)

Suppose Enc is an encryption scheme that has ciphertext indistinguishability, with key space $\mathcal{K}_\lambda$, message space $\mathcal{M}_\lambda$, and ciphertext space $\mathcal{C}_\lambda$. Assume $\mathcal{K}_\lambda = \mathcal{M}_\lambda = \{0,1\}^\lambda$. The decryption algorithm Dec will be un-important for this problem. Show that the following encryption schemes are also guaranteed to have ciphertext indistinguishability. You do not need to show how to decrypt or to prove correctness.

(a) $\mathsf{Enc}_a(k, m) = \mathsf{Enc}\,(k, (m, r))$. Here, the message space for Enc_c is $\{0,1\}^{\lambda/2}$, r is a random $\lambda/2$-bit string chosen during encryption, and (m, r) is the concatenation of m and r.

(b) $\mathsf{Enc}_b(k, m) = (r, \mathsf{Enc}(k, m \oplus r))$. Here, r is a random λ-bit string, chosen during encryption.

(c) $\mathsf{Enc}_c(k, m) = (\mathsf{Enc}(k, r), \mathsf{Enc}(r, m))$. That is, choose a random $r \in \{0,1\}^\lambda$, encrypt r using k, and then encrypt m using r.

2 Problem 2 (20 points)

Suppose $(\mathsf{Enc}, \mathsf{Dec})$ is an encryption scheme that has ciphertext indistinguishability, with key space $\mathcal{K}_\lambda$, message space $\mathcal{M}_\lambda$, and ciphertext space $\mathcal{C}_\lambda$. Assume $\mathcal{K}_\lambda = \{0,1\}^\lambda$ and $\mathcal{M}_\lambda = \{0,1\}^{2\lambda}$.

Consider the encryption scheme $(\mathsf{Enc}', \mathsf{Dec}')$ where $\mathsf{Enc}'(k, m) = \mathsf{Enc}(k, (m, k))$ and $m \in \{0,1\}^\lambda$. That is, to encrypt, first concatenate k to the message, and then encrypt the result using k as the key. $\mathsf{Dec}'(k, c)$ computes $(m', k') \leftarrow \mathsf{Dec}(k, c)$, and then outputs m'.

In this question, you will explore whether the ciphertext indistinguishability of $(\mathsf{Enc}, \mathsf{Dec})$ is enough to prove the ciphertext indistinguishability of $(\mathsf{Enc}', \mathsf{Dec}')$

(a) Devise an encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ that has ciphertext indistinguishability and is correct, but for which $(\mathsf{Enc}', \mathsf{Dec}')$ does *not* have ciphertext indistinguishability. In order to build $(\mathsf{Enc}, \mathsf{Dec})$, you may assume a PRG PRG on λ bit

seeds and whatever output length you need. If it helps, you may assume different lengths for the seed of PRG, as long as the seed length is $O(\lambda)$. You must prove both the security and correctness of $(\mathsf{Enc}, \mathsf{Dec})$, and also demonstrate an attack on $(\mathsf{Enc}', \mathsf{Dec}')$ when using your $(\mathsf{Enc}, \mathsf{Dec})$.

(b) Devise a different encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ that has ciphertext indistinguishability and is correct, and for which $(\mathsf{Enc}', \mathsf{Dec}')$ *also* has ciphertext indistinguishability (correctness of $(\mathsf{Enc}', \mathsf{Dec}')$ follows from the correctness of $(\mathsf{Enc}, \mathsf{Dec})$, so you do not need to prove the correctness of $(\mathsf{Enc}', \mathsf{Dec}')$). Again, you may assume a PRG PRG in order to construct your scheme. You must prove both the security and correctness of $(\mathsf{Enc}, \mathsf{Dec})$, and also the security of $(\mathsf{Enc}', \mathsf{Dec}')$ when using your $(\mathsf{Enc}, \mathsf{Dec})$.

Thus, the ciphertext indistinguishability of $(\mathsf{Enc}', \mathsf{Dec}')$ depends on exactly which $(\mathsf{Enc}, \mathsf{Dec})$ you start from. In particular, unlike the examples in Problem 1, the ciphertext indistinguishability of $(\mathsf{Enc}, \mathsf{Dec})$ alone is *not* enough to justify the security of $(\mathsf{Enc}', \mathsf{Dec}')$, and stronger properties would be needed.

3 Problem 3 (20 points)

Let $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}$ be a PRG, for some function $m(\lambda) > \lambda$. We will think of PRG in the asymptotic setting, where λ is the security parameter. Suppose there is a constant d such that PRG has the property that each individual bit of output only depends on only at most d bits of input. More precisely, we mean that for each bit of output, there is a program that only reads some subset of at most d input bits in order to generate that output bit. The sets of $\leq d$ input bits may be different for different output bits, so that all input bits are potentially used somewhere. Such PRG are potentially useful in settings where the hardware is very constrained, as the computation of each output bit would be very simple.

(a) Let $\mathsf{PRG}_i(s)$ denote the ith bit of output on input s. Prove that if PRG is a secure PRG, then $\mathsf{PRG}_i(s)$ must be uniformly random for a random s. In other words, if $\Pr[\mathsf{PRG}_i(s) = 1]$ was not exactly $1/2$ for some i, demonstrate a distinguishing attack which efficiently breaks the security of PRG. *Hint: what are the possible values for* $\Pr[\mathsf{PRG}_i(s) = 1]$*, given the restriction that* $\mathsf{PRG}_i(s)$ *depends only on* d *bits of* s*?*

(b) Show that in the case $d = 2$, no such PRGs are possible. To do so, show that if PRG is a secure PRG of the required form, then each $\mathsf{PRG}_i(s)$ has one of the forms $s_t, 1 \oplus s_t, s_t \oplus s_u, 1 \oplus s_t \oplus s_u$ where $t, u \in [\lambda]$ and s_t, s_u are the tth and uth bit of s. Show that this in turn means that PRG is actually *insecure*.

(c) Explain why the attack in (b) does not immediately extend to the case $d = 3$. What changes between $d = 2$ and $d = 3$ that allows the $d = 2$ case to be attacked but not $d = 3$? Remark: using more a more sophisticated analysis, it is actually possible to attack the cases $d = 3$ and $d = 4$. Interestingly, however, the attacks stop working at $d = 5$, and theres some good evidence that $d = 5$ may in fact be secure.

(d) Prove that if $m(\lambda) > \binom{\lambda}{d} \times d$, then PRG cannot be a secure PRG. Here, $\binom{\lambda}{d} = \lambda!/d!(\lambda-d)!$ is the binomial coefficient. Note that $\binom{\lambda}{d}d = \Theta(\lambda^d)$ for any constant d. In other words, PRG cannot have too much stretch. This is in contrast to general PRGs, which can have arbitrary polynomial stretch. *Hint: can a function from d bits to $d+1$ bits be a PRG, if d is a constant? What does this say about* PRG*?*

4 Problem 4 (20 points)

Let $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^n$ be a PRG with $\lambda < n$. We can use PRG to get an encryption scheme as $\mathsf{Enc}(k, m) = \mathsf{PRG}(k) \oplus m$. We saw in class that if the PRG is a secure pseudorandom generator, then Enc has ciphertext indistinguishability (in the one-time setting).

(a) Prove the converse: if $\mathsf{Enc}(k, m) = \mathsf{PRG}(k) \oplus m$ has ciphertext indistinguishability, prove that PRG *must* be a pseudorandom generator.

(b) Consider the encryption scheme $\mathsf{Enc}(k, m) = (r, \mathsf{PRF}(k, r) \oplus m)$, where r is chosen freshly at random during each ciphertext. In class, we saw that if PRF is a secure PRF, then Enc must satisfy LoR security. Show that the converse is *not* necessarily true: give an example of a PRF that is *not* secure, but for which $\mathsf{Enc}(k, m)$ has LoR security. In constructing PRF, you may assume a secure PRF'.