Homework 6

Do problem 3.9 (note it says variable output length PRG; see definition 3.18 in the book). Skim through section 4.6 in the book, and also read section 4.7.1, focusing on the proof of Theorem 4.14. Keep reading Victor Shoup’s article (section 5 is easier than the others, I think). Note: When Shoup uses the term “hash function” or “family of hash functions”, he doesn’t mean a hash function in the cryptographic sense (like MD5, SHA-1, etc), he means a hash function in the data structure sense, i.e., he just means a finite set of functions (all having the same range and domain) with some extra combinatorial property (that he defines).

(The first two problems are from my edition of Katz & Lindell—they are problems 4.2 and 4.3 in my edition, the ones I actually meant to assign for HW 5, sorry.)

**Additional Problem 1.** (K & L) Consider the following fixed-length MAC for messages of length $\ell(n) = 2n - 2$ using a pseudorandom function $F$: On input a message $m_0||m_1$ (with $|m_0| = |m_1| = n - 1$) and key $k \in \{0, 1\}^n$, algorithm $\text{Mac}$ outputs $t = F_k(0||m_0)||F_k(1||m_1)$. Algorithm $\text{Vrfy}$ is defined the natural way. Is $(\text{Gen}, \text{Mac}, \text{Vrfy})$ existentially unforgeable under a chosen message attack? Prove your answer.

**Additional Problem 2.** (K & L) Let $F$ be a pseudorandom function. Show that the following MAC for messages of length $2n$ is insecure: The shared key is a random $k \in \{0, 1\}^n$. To authenticate a message $m_1||m_2$ with $|m_1| = |m_2| = n$, compute the tag $(F_k(m_1), F_k(m_2))$.

**Additional Problem 3.** Complete the unfinished proof we started in class about how the existence of a PRG with 1-bit stretch implies the existence of a PRG with an arbitrary polynomial stretch (the proof where I was drawing red boxes, and blue dots). Write the whole statement and proof formally. Actually this proof is in the book, so you can just copy it out of the book if you want. But it will probably be more fun if you figure out the proof for yourself (and you will also remember it much better if you figure it out for yourself). Please also read the note on hybrid arguments below.

**Additional Problem 4.** Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ be an encryption scheme. Define the sheme $\Pi' = (\text{Gen}', \text{Enc}', \text{Dec}')$ by putting

- $\mathcal{K}_n' = \mathcal{K}_n \times \mathcal{K}_n$ where $\mathcal{K}_n$ is the space of $\Pi$, $\mathcal{K}_n'$ is the key space of $\Pi'$ (for security parameter $n$).
- $\text{Gen}'(1^n) = (\text{Gen}(1^n), \text{Gen}(1^n))$ (where the coins used for the first and second run of $\text{Gen}(1^n)$ are independent)
- $\text{Enc}'((k_1, k_2), m) = (\text{Enc}(k_1, m), \text{Enc}(k_2, m))$

Also, $\text{Dec}'$ works the obvious way (say $\text{Dec}'((k_1, k_2), (c_1, c_2)) = m$ if and only if $\text{Dec}'(k_1, c_1) = \text{Dec}'(k_2, c_2) = m$). Prove that $\Pi'$ is CPA-secure if $\Pi$ is CPA-secure.

*Note on hybrid arguments:* I am going to outline the overall structure of a hybrid argument, to make a subtle comment about uniformity versus non-uniformity in hybrid arguments.

We have an adversary $A$ that can distinguish between worlds $X$ and $Y$ and we need to build an adversary $B$ such that $B$ can distinguish between some other (unrelated) worlds $x$ and $y$ (or that can do some other task, like predict a bit, etc, but here I’ll assume that $B$ has a distinguishing task). For this purpose, we construct a sequence of “intermediate worlds”

$$W_0, W_1, W_2, \ldots, W_n$$

such that

$$X = W_0, Y = W_n$$
and such that if $A$ can distinguish between $\mathcal{W}^i$ and $\mathcal{W}^{i+1}$, then $B$ can use this to distinguish between $x$ and $y$.

Write $A$'s distinguishing advantage between two worlds as:

$$\Delta_A(w_1, w_2) := \Pr[A(w_1) = 1] - \Pr[A(w_2) = 1].$$

Then we have

$$\Delta_A(X, Y) = \sum_{i=0}^{n-1} \Delta_A(W^i, W^{i+1}).$$

So if $A$ has advantage $\Delta_A(X, Y) = \varepsilon_A(n)$, then there exists an $i \in \{0, \ldots, n-1\}$ such that

$$\Delta_A(W^i, W^{i+1}) \geq \varepsilon_A(n)/n.$$

We might be tempted, then, to define $B$ simply like this: “$B$ uses his inputs (which is either $x$ or $y$, $B$ doesn’t know) to run $A$ in a world which is equivalent to $\mathcal{W}^i$ if $B$’s input is $x$, and is equivalent to $\mathcal{W}^{i+1}$ if $B$’s input is $y$; therefore $B$’s distinguishing advantage will be exactly $\Delta_A(W^i, W^{i+1}) \geq \varepsilon_A(n)/n$”.

This definition of $B$ is actually not good and the reason is this: the “good” value of $i$ might change unpredictably with the security parameter $n$. For example, for $n = 128$ we might have that the “good $i$” is $i = 67$, and for $n = 512$ the “good $i$” might be $i = 213$, and so on—in fact, finding the “good $i$” might require running $A$ many times and doing probability estimates, etc. In other words, the issue is that we want a uniform algorithm $B$ that works for all $n$, and finding which is the “good $i$” is not easy. (Recall that a uniform algorithm is an algorithm that does not depend on the input length—same algorithm for all input lengths—whereas a non-uniform algorithm can receive a piece of “advice” that depends on the input length—here the advice would be the “good $i$”.)

So, what’s written above is not a good definition for $B$—it’s non-uniform. The right way to do things is to have $B$ select at random the value $i$ to be used. This will give $B$ advantage

$$\sum_{i=0}^{n-1} \frac{1}{n} \Delta_A(W^i, W^{i+1}) = \varepsilon_A(n)/n$$

which is the same as before, but without having to worry about “choosing the right $i$”.

Last comment: Of course, the number of hybrids could, in general, be any polynomial of $n$, it’s not necessarily exactly equal to the “security parameter” $n$. Also the number of hybrids could be constant, and the same remark would apply. For example if we had only three hybrids independently of $n$, such as

$$(\mathcal{W}^0, \mathcal{W}^1, \mathcal{W}^2)$$

where $X = \mathcal{W}^0$ and $Y = \mathcal{W}^2$, then either $\Delta_A(\mathcal{W}^0, \mathcal{W}^1) \geq \varepsilon_A(n)/2$ or $\Delta_A(\mathcal{W}^1, \mathcal{W}^2) \geq \varepsilon_A(n)/2$. You might be tempted to say “if $\Delta_A(\mathcal{W}^0, \mathcal{W}^1) \geq \varepsilon_A(n)/2$, then $B$ behaves this way, and if $\Delta_A(\mathcal{W}^1, \mathcal{W}^2) \geq \varepsilon_A(n)/2$, then $B$ behaves that way”. But again that’s not a good idea, because which case it is could generally depend on the value of $n$. Once again the “correct” way of doing the hybrid argument is to have $B$ choose at random between the pair $(\mathcal{W}^0, \mathcal{W}^1)$ and the pair $(\mathcal{W}^1, \mathcal{W}^2)$. 