# Starting out with crypto reductions—Exercises

1. **Prove that flipping bits of a PRG doesn't affect its security.**
   Given a PRG $G$ that is secure (i.e., satisfies the indistinguishability game) construct a PRG $G'(x) := \overline{b_1}\overline{b_2}\cdots\overline{b_n}$ where $(b_1 b_2 \cdots b_n) = G(x)$, the output of $G$ on $x$ and $\overline{b}$ is the bit $b$ flipped. Show that $G'$ is still secure. To do this, construct an adversary $\mathcal{B}$ that given a string decides whether it is from $G$ (i.e., $G$ evaluated on a random input) or completely random with probability better than $1/2$. To do this, it uses $\mathcal{A}$ who claims to break the distinguishability test for $G'$. Therefore, $\mathcal{B}$ must somehow use its challenge string $w$ and concoct a string for $\mathcal{A}$ to help it decide whether it is random or not.

2. **Prove that permuting the bits of a PRG is still secure.**
   Given a PRG $G$, and a permutation $\pi$ (and its inverse permutation $\pi^{-1}$)[1], consider $G_\pi(x) = (b_{\pi(1)} b_{\pi(2)} \cdots b_{\pi(n)})$ where $(b_1 b_2 \cdots b_n) = G(x)$. Show, largely along the lines of Exercise 1 that $G_\pi$ is secure. In particular if $\pi$ is the identity this is trivial. If $\pi(i) = n + 1 - i$, i.e., $G'$ simply reverses the bits of the output, the proof is covered in the section. You need to use $\pi^{-1}$.

3. **Prove that a $\ell$-expanding PRG is a small domain PRF.**
   This is HW1 problem 7b. The formal statement can be found on the website. Here, to prove the security, given an adversary $\mathcal{A}$ for the small domain PRF, you are required to construct an adversary $\mathcal{B}$ that interacts with the PRG-Challenger and distinguishes whether the string given were random or pseudorandom. Remember, $\mathcal{B}$ must present a PRF-Challenger $\mathsf{Chal}$ to $\mathcal{A}$ and ensure that he is **faithfully** simulating either a random function or a pseudorandom function.

4. **Prove that truncating the output of a PRF is still secure.**
   More formally, given a PRF $F : K \times X \to \{0,1\}^m$, construct $F'(k, x) = F(k, x)|_\ell$, where $\ell < m$ is the output length of $F' : K \times X \to \{0,1\}^\ell$ is also secure. The way the proof would go would be to take an efficient adversary $\mathcal{A}$ that breaks the PRF indistinguishability game for $F'$ and construct $\mathcal{B}$ that has a challenger $\mathsf{Chal}$ as in the security definition for $F$ and must guess whether $\mathsf{Chal}$ is giving him his queries evaluated on $F$ or a random function. Note here $\mathcal{B}$ **must present** a challenger for $\mathcal{A}$. See Exercise 3 above for more ideas.

5. **Prove that any subset of the bits of a PRG are still indistinguishable.**
   This proof is largely along the lines of Exercise 4 above. This is much simpler because you aren't required to simulate queries (the challenger for a PRG simply presents one of two strings, and asks for a distinguisher).

6. **Introduce hardness of SVP in perp lattices and prove that $f_A(x) = A \cdot x \pmod q$ is a collision resistant hash function.**

[1] I.e., $\pi^{-1}(\pi(i)) = i$ for all $i$.

There is this hard problem that has recently received a lot of interest in the crypto community that we will refer to as SVP in random lattices. The formal statement is as follows: Consider a random matrix $A \in \mathbb{Z}_q^{n \times m}$. Finding a "short" $e \in \mathbb{Z}^m$ such that $A \cdot e = 0 \pmod{q}$ is hard. Here short implies that $\|e\|_2 \leq \beta$ for some parameter $\beta$. Therefore, an adversary $\mathcal{B}$ that breaks this system will, given a random matrix output $e$ that satisfies the above conditions.

Now, we can construct a simple collision-resistant hash function from this system. Consider $H : \{0,1\}^m \to [q]^n$, that takes $m$-bit strings and outputs $n$ elements in $[q]$. [2] For a random $A$, define $H_A(x) = A \cdot x \pmod{q}$. Check for yourself that the input and outputs of the hash function are correct. Show that finding collisions in $H_A(\cdot)$ are as hard as SVP in a random lattice.

A proof would require you to construct an adversary $\mathcal{B}$ that takes an adversary $\mathcal{A}$ that outputs collisions from $H_A(\cdot)$ and use this to break SVP.

7. **Prove that PRGs imply One Way Functions.**
Recollect in class, we mentioned several ways to construct one-way functions. The simplest way, and the one with least structure to exploit was this generic construction from pseudorandom functions. First, we will show that a PRG implies a one-way function. Let $G : \{0,1\}^s \to \{0,1\}^n$, $n > s$ be a PRG that takes a $s$-bit input and outputs $n$ pseudorandom bits. Show that $F(x) := G(x)$ is a one-way function. An adversary for a OWF takes the image of a random element and outputs a preimage that evaluates to the same value. Therefore, given $G(r)$ for random $r$, the adversary outputs $r'$ (that might be equal to $r$) such that $G(r) = G(r')$. Therefore show that you can construct an adversary $\mathcal{B}$ that distinguishes a PRG output from random given $\mathcal{A}$ that inverts the one-way function with non-negligble probability.

8. **Prove that a PRFs imply OWFs.**
Now we show that we can construct PRFs from OWFs. Recall that given sufficiently expanding PRGs we can construct PRFs therefore for sufficiently expanding PRGs, this proves the previous exercise as well. Consider $\text{OWF}(x) := F(x,0) \| F(x,1)$ for simplicity. Show that an adversary $\mathcal{A}$ that inverts the OWF can be used to construct an adversary $\mathcal{B}$ that interacts with a PRF challenger $\mathsf{Chal}$ and decides whether it is dealing with a random function or a pseudo-random function. This means that $\mathcal{B}$ must behave sufficiently differently for the two possible games the challenger plays with him (see the definition of PRFs).

Now, can you see how this proof extends to $\text{OWF}(x) := F(x,\alpha_1) \| F(x,\alpha_2) \ldots \| F(x,\alpha_q)$ for any *fixed constants* $\alpha_1, \ldots, \alpha_q$ where $q$ is at most a polynomial in $|x|$ ?

[2] Recollect that $[q] = \{1, \ldots, q-1, q\}$.