1 Fun with PRGs

a. $G_a$ is not always a PRG. As a counterexample, consider $G_1 = G_2$, and note that $G_a(s) = 0^{2k}$ for all $s$ in this case.

b. $G_b$ is a PRG. To prove this, define the distributions:

$$D_0 = \{ s_1, s_2 \leftarrow \{0, 1\}^k : G_3(s_1) \oplus G_3(s_2) \}$$
$$D_1 = \{ s_1 \leftarrow \{0, 1\}^k; R \leftarrow \{0, 1\}^{3k} : G_3(s_1) \oplus R \}$$
$$D_2 = \{ R \leftarrow \{0, 1\}^{3k} : R \}$$

We wish to show that $D_0 \approx D_2$. Since $D_1$ and $D_2$ are identically distributed, it suffices to show that $D_0 \approx D_1$ and to conclude with the transitivity of indistinguishability.

Assume for the sake of contradiction that there exists some adversary $A$ which distinguishes between $D_0$ and $D_1$ with nonnegligible advantage. We wish to construct a p.p.t. $B$ that distinguishes between

$$D_G = \{ s \leftarrow \{0, 1\}^k : G_3(s) \}$$

and

$$D_R = \{ R \leftarrow \{0, 1\}^{3k} : R \}$$

On input $X$, $B$ chooses $s_1 \leftarrow \{0, 1\}^k$ and outputs $A(G_3(s_1) \oplus X)$. If $X$ is random, then $G_3(s_1) \oplus X$ is identically distributed to $D_1$; if $X$ is some $G_3(s)$, then $G_3(s_1) \oplus X$ is distributed identically to $D_0$.

$B$ will succeed exactly when $A$ succeeds, meaning that $B$ contradicts our assumption that $G_3$ is a PRG. $B$ cannot exist, and thus $D_0 \approx D_2$, so $G_b$ is a PRG.
2 Pseudorandom Generators and the Hybrid Argument

Solution written by Tracy Chin.

We will show that given \( A \) distinguishing \( H_i \) and \( H_{i+1} \), we can construct \( B \) distinguishing \( B_{pk}(x) \) from a random bit.

We construct \( B \) as follows:

On input \((f(x), b)\):

\[
\begin{align*}
    r_1 \ldots r_i &\leftarrow \{0, 1\}^{i-1} \\
    b_j &\leftarrow B_{pk}(f^{(j-1)}(x)), \text{ for } j = i + 2, \ldots, k \\
    s_k &\leftarrow f^{(k-1)}(f(x)) \\
    w &\leftarrow s_k r_1 \ldots r_{i-1} \circ b \circ b_{i+1} \ldots b_k \\
    a &\leftarrow A(w, pk) \\
\end{align*}
\]

if \( a = i \) then

\begin{align*}
    \text{return } 0 \quad \triangleright b \text{ is } B_{pk}(x) \\
\end{align*}

else

\begin{align*}
    \text{return } 1 \quad \triangleright b \text{ is random} \\
\end{align*}

end if

If \( b = B_{pk}(x) \), then \((w, pk)\) is distributed exactly like \( H_i \), and if \( b \) is random, then \((w, pk)\) is distributed exactly like \( H_{i+1} \).

Thus, we find:

\[
\begin{align*}
    \Pr[B \text{ correct}] &= \frac{1}{2} \Pr[x \leftarrow \{0, 1\}^k; b \leftarrow \{0, 1\} : B(f(x), b) = 1] \\
    &\quad + \frac{1}{2} \Pr[x \leftarrow \{0, 1\}^k; b \leftarrow B_{pk}(x) : B(f(x), b) = 0] \\
    &= \frac{1}{2} \Pr[w \leftarrow H_{i+1}(k) : A(w) = i + 1] + \frac{1}{2} \Pr[w \leftarrow H_i(k) : A(w) = i] \\
    &= \Pr[A \text{ correct}] \\
\end{align*}
\]

Therefore, if \( A \) is correct with nonnegligible probability, then \( B \) can distinguish a hardcore bit from a random bit with nonnegligible probability.

However, we showed in class that hardcore bits are indistinguishable from random bits, so we must have \( H_i(k) \approx H_{i+1}(k) \).

Further, this reduction is not dependent on the value of \( i \), so the same \( \nu(k) \) that bounds distinguishing \( B_{pk}(x) \) from a random bit will also bound distinguishing \( H_i \) from \( H_{i+1} \) for all \( i \).

Homework 6 Solutions
3 Decisional Diffie-Hellman Assumption and PRGs

Solution written by David Ellis Hershkowitz.

a. I use the following notation for the various distributions:

\[ D_1 = (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{a_1b}, g^{a_2b}, \ldots, g^{a_{w-b}}) \]
\[ D_2 = (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{c_1}, g^{c_2}, \ldots, g^{c_2}) \]
\[ D_3 = (g, g^a, g^b, g^{ab}) \]
\[ D_4 = (g, g^a, g^b, g^c) \]

We can prove this by a hybrid construction argument. Define the \( i \)th hybrid distribution by

\[ H_i = (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{c_1}, g^{c_2}, \ldots, g^{c_i}, g^{a_{i+1}b}, g^{a_{i+2}b}, \ldots, g^{a_{w-b}}) \]

Under the DDH assumption we cannot distinguish between \( H_i \) and \( H_{i+1} \). Assume for the sake of contradiction that we could with a ppt algorithm \( A \). That is, \( A \) distinguishes between these two distributions:

\[ H_i = (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{c_1}, g^{c_2}, \ldots, g^{c_i}, g^{a_{i+1}b}, g^{a_{i+2}b}, \ldots, g^{a_{w-b}}) \]
\[ H_{i+1} = (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{c_1}, g^{c_2}, \ldots, g^{c_i}, g^{c_{i+1}}, g^{a_{i+2}b}, \ldots, g^{a_{w-b}}) \]

We could use \( A \) to construct an algorithm \( B \) that, given an input from \( D_3, (g, g^a, g^b, g^{ab}) \), has a non-negligible advantage of guessing \( D_3 \) versus \( D_4 \). Construct \( B \) as follows. Draw \( b \), all \( a_j \)s and all \( c_k \) from \( \mathbb{Z}_q \) uniformly at random. Construct samples \( s_1 \) and \( s_2 \) as:

\[ s_1 = (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{c_1}, g^{c_2}, \ldots, g^{c_i}, g^{a_{i+1}b}, g^{a_{i+2}b}, \ldots, g^{a_{w-b}}) \]
\[ s_2 = (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{c_1}, g^{c_2}, \ldots, g^{c_i}, g^{c_{i+1}}, g^{a_{i+2}b}, \ldots, g^{a_{w-b}}) \]

Provide \( s_1 \) and \( s_2 \) as input to \( A \) and, if it guesses \( H_i \), then guess \( D_3 \); otherwise, guess \( D_4 \).

\( B \) has a non-negligible probability of distinguishing between \( D_3 \) and \( D_4 \). We assumed for the sake of contradiction that \( A \) has a non-negligible probability of guessing correctly if given inputs uniformly at random – \( A \) is given inputs uniformly at random so it has a non-negligible probability of succeeding. When \( A \) succeeds, so does \( B \), so \( B \) has a non-negligible probability of succeeding.

Given our assumption of DDH, this is a contradiction, so we conclude that the \( i \)th hybrid distribution is indistinguishable from the \( i+1 \)th hybrid distribution.

Lastly, note that we have a polynomial number of hybrids since \( w \) is a polynomial of \( \log(q) \) and so the 0th hybrid is indistinguishable from the \( w \)th, which are \( D_1 \) and \( D_2 \) respectively.
b. At a high level, the way the PRG $G$ works is by taking in sufficient true randomness to generate all $a$s and $b$ – and then outputting not only the $g$s which are generated from $a$s and $b$s, but also those that are generated from $c$s.

More concretely, $G$ works as follows. $G$ takes as input an $m$-bit string where $m = (w + 1) \log(q)$. $\log(q)$ is how much space is required to represent a single element of $\mathbb{Z}_q$, of which we would like $w + 1$ since we want to represent all of the $a$s (of which there are $w$ total) as well as $b$. Next we output $g^{a_1} \circ g^{a_2} \circ \ldots \circ g^{a_w} \circ g^{a_1-b} \circ g^{a_2-b} \circ \ldots \circ g^{a_w-b}$ where each group element is represented in binary with $\log(q)$ bits. This means that the length of our output in bits, $n$, is $2w \log(q)$ – meaning we are almost length doubling!

Now note that under the DDH assumption this PRG is secure. We can prove the security of $G$ by way of a reduction to show the contrapositive. Suppose $G$ is not a PRG. This must mean that there exists some ppt algorithm, $A$, which is able to distinguish between $\{0, 1\}^n$ and $\{s \leftarrow \{0, 1\}^m : G(s)\}$. We can use $A$ to construct another algorithm, $B$, which distinguishes between $D_1$ and $D_2$ (see part a for what these distributions are) with non-negligible probability.

Construct $B$ as follows. Given an element of $D_1$, $B$ simply provides this as input to $A$ and, if $A$ returns $\{0, 1\}^n$, $B$ guesses $D_2$; otherwise, it guesses $D_1$.

$B$ breaks the DDH assumption. First note that if an element of $G$ is chosen uniformly at random then each bit of its $\lceil \log(q) \rceil$ binary representation can be thought of as chosen uniformly at random. This is because every element of $G$ uniquely corresponds to an element of $\{0, 1\}^{\log(q)}$. Therefore, $A$ receives input as it would in “the wild,” and so given an element of $\{0, 1\}^n$ it has a non-negligible absolute difference in the probability of it guesses $\{0, 1\}^n$ minus the probability it guesses $\{s \leftarrow \{0, 1\}^m : G(s)\}$. When $A$ succeeds, so does $B$, and so $B$ breaks the DDH assumption.

Thus we have proven the contrapositive we sought to prove; if we buy into the DDH assumption, our PRG is secure.

c. At a high level we can distinguish between $g^{ab}$ and $g^c$ because $g^{ab}$ is much more likely to be a square, which we can efficiently test.

In more detail, consider the following ppt adversary $A$ which distinguishes between $(g, g^a, g^b, g^{ab})$ and $(g, g^a, g^b, g^c)$ under $\mathbb{Z}_p^*$, thereby breaking the DDH assumption in $\mathbb{Z}_p^*$. Given an input $x$, test if $x$ is a square by taking it to the $\frac{p-1}{2}$ power and seeing if the result is congruent to 1. If it is congruent to 1, then guess $(g, g^a, g^b, g^{ab})$; if not, guess $(g, g^a, g^b, g^c)$.

$A$ breaks the indistinguishability of $(g, g^a, g^b, g^{ab})$ and $(g, g^a, g^b, g^c)$ under $\mathbb{Z}_p^*$. Suppose $A$ is given a sample from $(g, g^a, g^b, g^{ab})$. Note that the probability that $g^{ab}$ is a square is the probability that $a$ or $b$ is even. By a simple counting argument, the probability that $a$ or $b$ is even is $\frac{1}{2} + \frac{1}{2} - \frac{1}{4} = \frac{3}{4}$. The probability that $A$ guesses
\((g, g^a, g^b, g^{ab})\) is the probability that \(g^{ab}\) is a square which is \(\frac{3}{4}\) whereas the probability that \(A\) guesses \((g, g^a, g^b, g^c)\) is the probability that \(g^{ab}\) is not a square which is \(\frac{1}{4}\). The absolute difference in these probabilities is \(\frac{1}{2}\) which is clearly not negligible so \(A\) distinguishes between the two distributions with non-negligible probability, thereby showing that the DDH assumption does not hold in \(\mathbb{Z}_p^*\).

4 Random Self-Reducibility of RSA

a. First, we note that the group \(\mathbb{Z}_N^*\) is cyclic. This means that there is some generator \(g\) such that every element \(r\) in the group can be written uniquely as \(r = g^i\). Now, raising some fixed \(r\) to the \(e^{th}\) power gives us \(r^e = g^{ie}\). If \(ie > \varphi(N)\) then we simply reduce to get an exponent that makes sense (so we use the smallest exponent \(k\) such that \(g^{ie} \equiv g^k \mod N\)). It should be clear that this process is bijective, since if any \(r^e\) doesn’t hit an element of the group, we would have that \(g^{ie}\) isn’t an element of the group, which is a contradiction by the definition of \(g\) being a primitive root. For injectivity, just note that \(i\) is defined uniquely for each \(r\) and therefore if \(i \neq j\) it is impossible that \(ie = je\) and \(g^{ie} = g^{je}\). Therefore, \(D_2\) is identical to \(D_1\).

Multiplying by an element \(y \in \mathbb{Z}_N^*\) follows the exact same logic. We can write each \(r^e\) as some \(s \in \mathbb{Z}_N^*\) by what we showed above, and then say that \(s = g^j\) and \(y = g^k\) for some \(j, k\). Then \(sy = g^{j+k}\), which is again going to be some element of the group. The same exact logic we followed for showing that exponentiation permutes the group applies here; multiplying by an element of the group will just return the group in a different order (permuted). So \(D_3\) is also the same.

b. Since \(\frac{z^e}{r^e} = y\), we can see that \(\frac{z}{r} = x\), where \(x\) is an integer such that \(x^e = y\). (Note that when we write \(\frac{1}{r^e}\), what we really mean is multiplication by the multiplicative inverse of \(r^e\) modulo \(N\)).

c. Our algorithm first picks some random \(r \in \mathbb{Z}_N^*\). Then, it computes \(r^e \cdot y\), which we proved in part (a) is going to look the same as picking a random \(r\) from the group. Now, using \(A\), we find \(z\) such that \(z^e = r^e y\) with probability \(\varepsilon\). Based on what we found in part (b), we know that we can now find \(x\) such that \(x^e = y\), and that the overall probability of doing so is \(\varepsilon\).

d. If RSA is hard, it is clear that our second assumption will also be hard, since \(\text{QR}_N\) is a subset of \(\mathbb{Z}_N^*\) (and not a subset of negligible size; we always have \(|\text{QR}_N| \leq \frac{1}{4}|\mathbb{Z}_N^*|\) for \(N = pq\) a Blum integer). Therefore, \(B\) can just run its input through \(A\) and be correct (i.e. have a quadratic residue) with probability \(\frac{1}{4}\), meaning our overall probability of success is \(\frac{\varepsilon}{4}\).

To prove the other direction, we prove the contrapositive. If RSA is easy, so the probability that someone finds the \(e^{th}\) root is actually non-negligible \(\varepsilon\), we can reduce
the second problem in \( QR_N \) to working over \( Z_N^* \). As we have done in previous parts of the problem, we can transform an element \( y \in QR_N \) to look like a normal element by multiplying it by some \( r^e \), where \( r \) is a random element chosen from \( Z_N^* \). This is now just the RSA problem, and since we assuming that was easy we do what we did in part (c) to see that the second problem will be easy as well. Therefore, the two assumptions are equivalent.