CS 427, Assignment 3

Cody Malick
malickc@oregonstate.edu

February 3, 2017

1

a

This PRG is insecure. We can show this by having two QUERY functions, $QUERY_1$ and $QUERY_2$. $QUERY_2$ returns a uniform random distribution, while $QUERY_1$ uses a psuedorandom function $H(s)$:

```
$QUERY_1() :$
        $s \leftarrow \{0,1\}^\lambda$
        $return\ H(s)$
```

```
$QUERY_2() :$
        $z \leftarrow \{0,1\}^{2\lambda}$
        $return\ z$
```

I claim that an adversary can tell the difference between these two functions by only looking at the result of calling either function. The adversary can tell which library he is calling, by simply looking at the second half of the output of $QUERY_1$. The calling function $\mathcal{F}$:

```
$\mathcal{F}() :$
    $x := QUERY()$
    $x_1 := first\ half\ of\ x$
    $x_2 := second\ half\ of\ x$

    $y := QUERY()$
    $y_1 := first\ half\ of\ y$
    $y_2 := second\ half\ of\ y$

    $if\ (\ x_2 == y_2\ )\ \{$
      $return\ true$
    $\}$
    $return\ false$
```

Given that $\mathcal{F}$ calls $QUERY_1$ twice, then $\mathcal{F}$ will return true with a probability of 1. While if $\mathcal{F}$ calls $QUERY_2$ twice, then the probability of $x_2 == y_2$ is quite small, $\frac{1}{2^\lambda} * \frac{1}{2^\lambda}$. Given that information, the PRG is not secure.

b

I claim this PRG is secure due to OTP. To show that this PRG is secure, we must show that, given a calling function, we cannot differentiate between two queries, $QUERY_1, QUERY_2$:

```
$QUERY_1()$ :
        $s \leftarrow \{0,1\}^\lambda$
        $return\ H(s)$
```

```
$QUERY_2()$ :
        $z \leftarrow \{0,1\}^{2\lambda}$
        $return\ z$
```

Inside of $H(s)$, getting two random strings and xoring them together looks a lot like OTP. We can try and convert $H(s)$ into a different form. With the original $H(s)$:

```
$H(s)$ :
        $x := G(s)$
        $y := G(0^\lambda)$
        $return\ x \oplus y$
```

We can abstract out the xor into a function $\mathcal{F}$:

```
$\mathcal{F}(k, m)$ :
        $return\ k \oplus m$
```

```
$H(s)$ :
        $x := G(s)$
        $y := G(0^\lambda)$
        $return\ \boldsymbol{\mathcal{F}(x, y)}$
```

I claim this does not change the calling function, as we are performing the same operations, just with a linking function rather than in $H(s)$. $\mathcal{F}$ is identical to OTP, which we have proven has a uniformly distributed output. With that in mind, both queries, $QUERY_1, QUERY_2$ both return a uniformly distributed output, and we can differentiate between them. They are, then, indistinguishable. $H()$ is a secure PRG.

c

This problem is very similar to the first problem, but the primary difference is that the function will `always` return the seed as the first half of the return value. I claim that this function is secure, as the seed is randomly generated from a uniform distribution:

```
$H(s)$ :
        $x := G(s)$
        $return\ s||x$
```

We can first abstract the call to $G(s)$ as a uniformly random distribution, as we know that $G()$ is secure:

```
$H(s)$ :
        $x \leftarrow \{0,1\}^{2\lambda}$
        $return\ s||x$
```

It is clear now that we have two values, both of which have a uniform random distribution. The resulting probability distribution is then $\frac{1}{2^{2\lambda}} * \frac{1}{2^\lambda} = \frac{1}{2^{3\lambda}}$. With that output, then $H()$ is a secure PRG

2

This is a relatively simplistic proof. Our end goal is simply to show that we can freely exchange G_1 and G_2. Starting with $\mathcal{L}_{which-prg}^{G_1}$:

```
$QUERY():$
    $s \leftarrow \{0,1\}^{\lambda}$
    $return\ G_1(s)$
```

My first step would be abstract away the call to G_1 into a seperate function:

```
$QUERY():$
    $s \leftarrow \{0,1\}^{\lambda}$
    $\boldsymbol{z := \mathcal{F}(s)}$
    $return\ \boldsymbol{z}$
```

Where $\mathcal{F}$:

```
$\mathcal{F}(s):$
    $z := G_1(s)$
    $return\ z$
```

Because we assume that G_1 is a secure PRG, then we can simply set the output of the function to a uniformly random distribution:

```
$\mathcal{F}(s):$
    $\boldsymbol{z \leftarrow \{0,1\}^{2\lambda}}$
    $return\ z$
```

We can take advantage of the fact that the output is uniformly random, to substitute G_2 in place of the uniform distribution, as we assume G_2 is a secure PRG: input and output ranges:

```
$\mathcal{F}(s):$
    $z := \boldsymbol{G_2(s)}$
    $return\ z$
```

With that changed, we can substitute the resulting function back into $QUERY()$:

```
$QUERY():$
    $s \leftarrow \{0,1\}^{\lambda}$
    $\boldsymbol{z := G_2(s)}$
    $return\ z$
```

With that in place, we've shown that $\mathcal{L}_{which-prg}^{G_1}$ is indistinguishable from $\mathcal{L}_{which-prg}^{G_2}$