7.1 PRF using PRG

7.1.a G is injective

Let g be the return value of $G(x)$. Because $G(x)$ is injective this g can be only reached once by any input x. Therefore the following holds:

$$F'(k,x) = F(k,G(x)) = F(k,g)$$

Furthermore g is in $\{0,1\}^{2\lambda}$. Because we know that F is secure for any input k and g and g is an arbitrary 2λ bit-string, F' is also secure.

7.1.b H and G

We consider the following distinguisher:

Distinguisher A
pick $x \in \{0,1\}^{\lambda-1}$
$s1 = x \parallel 0$
$s2 = x \parallel 1$
return $\textsc{Lookup}(s1) = \textsc{Lookup}(s2)$

First we will pick a random seed x and concatenate it once with 0 and with 1. These two λ bit-strings are then put in the Lookup-function. In the end we will check if both outputs are equal.

Distinguisher A		$L^{F'}_{PRF-real}$		$L^{\tilde{G}}_{PRF-real}$
pick $x \in \{0,1\}^{\lambda-1}$	$\diamond$	$k \leftarrow \{0,1\}^{\lambda}$	$\diamond$	$h = H(x_1 \cdots x_{\lambda-1})$
$s1 = x \parallel 0$		Lookup(x)		**return** h
$s2 = x \parallel 1$		$g = \tilde{G}(x)$		
return $\textsc{Lookup}(s1) = \textsc{Lookup}(s2)$		**return** $F(k,g)$		

It is obvious that the algorithm combined with $L^{F'}_{PRF-real}$ will always output 1 because there will be no difference between the outputs if in the inputs only the last bit is different.

Distinguisher A		$L^{F'}_{PRF-rand}$
pick $x \in \{0,1\}^{\lambda-1}$	$\diamond$	$T := \textit{empty associated array}$
$s1 = x \parallel 0$		Lookup(x)
$s2 = x \parallel 1$		**if** $T[x]$ undefined:
return $\textsc{Lookup}(s1) = \textsc{Lookup}(s2)$		$T[x] \leftarrow \{0,1\}^{2\lambda}$
		return $T[x]$

This combination will only return 1 if for $s1$ and $s2$ the same outputs are saved in T. The probability for this is $2^{-\lambda}$.
For the advantage, we get:

$$Bias(A) = |\ P[A \diamond L^{F'}_{PRF-Real} \to 1] - P[A \diamond L^{F'}_{PRF-Rand} \to 1]\ | = 1 - 2^{-\lambda}$$

, which is clearly not negligible for $\lambda \to \infty$.

7.2 Pseudo-random Permutations

7.2.a Probability of collisions

The PRP can output 2^λ different permutations.
Furthermore if we have blocklength μ, there can be $\mu!$ different permutations. Therefore the probability that one of them will agreef with one permutation generated by the PRP will be:

$$Pr[permutation\ is\ one\ of\ PRP] = \frac{2^\lambda}{\mu!}$$

7.2.b $\lambda = \mu = 128$

When $\lambda = \mu = 128$ we have 2^{128} possible keys that can be created. For the probability we then get:

$$Pr[permutation\ is\ one\ of\ PRP] = \frac{2^{128}}{128!} \approx 8.82 * 10^{-178}$$

This probability is pretty much negligible and it is very much unlikely that this would be the case.

7.3 Insecurity of two-round keyed Feistel cipher

We know that $M = L_0 \| R_0$. Therefore we also know that:

$$C = L_2 \| R_2 = L_0 \oplus F(K_0, R_0) \| R_0 \oplus F(K_1, L_0 \oplus F(K_0, R_0))$$

With this we can compute the following things:

$$\begin{aligned} L_0 \oplus L_2 &= F(K_0, R_0) \\ R_0 \oplus R_2 &= F(K_1, L_0 \oplus F(K_0, R_0)) \end{aligned}$$

With these information it is not impossible to find out K_0 and K_1 and therefore the two-round keyed Feistel cipher is not secure.