Introduction to Cryptography - Exercise session 4

Prof. Sebastian Faust

November 14, 2018

The purpose of this exercise session is to first exercise (again) the concept of a psedurorandom function (PRF) and CPA-security. In the second part of the exercise session, we discuss the definition of a pseudorandom permutation (PRP) which was intuitively explained at the end of the lecture. In addition, we explain the concept of Feistel Networks.

Exercise 1 (Extending the range of a PRF)

Let F be a PRF. Below there are two attempts to make another PRF F'. In each case either prove that the result is also a PRF or design a ppt algorithm which breaks it.

(a) $F'_s(x) := F_{0^n}(x) \parallel F_s(x)$, for $F_s \colon \{0,1\}^n \to \{0,1\}^n$.

Solution:

This is not a PRF. We construct a distinguisher D as follows: On input 1^n and having access to oracle $\mathcal{O}$, D behaves in the following way:

- D chooses $x \in \{0,1\}^n$ arbitrarily and receives $y_L \parallel y_R := \mathcal{O}(x)$ as an answer.
- D outputs 1, when $y_L = F_{0^n}(x)$. Note here the value of $F_{0^n}(x)$ is uniquely determined.

We will now prove that the constructed distinguisher D with non-negligible probability can distinguish between $\mathcal{O}$ being a F'_s for some $s \in \{0,1\}^n$ or a random function $F \colon \{0,1\}^n \to \{0,1\}^{2n}$.
If the $\mathcal{O} = F'_s(\cdot)$, then we have $y_L \parallel y_R = F_{0^n}(x) \parallel y_R$. And therefore

$$\Pr_{s \leftarrow \{0,1\}^n}[\mathsf{D}^{F'_s(\cdot)}(1^n) = 1] = 1. \tag{1}$$

Now if $\mathcal{O}$ is a truly random function $f \colon \{0,1\}^n \to \{0,1\}^{2n}$, we have that $f(x)$ is a random string. Hence the probability that D outputs 1 is equal to the probability that the first n bits of $f(x)$ are equal to $F_{0^n}(x)$. This implies that

$$\Pr_{f \leftarrow \mathsf{Func}(n,2n)}[\mathsf{D}^{f(\cdot)}(1^n) = 1] = 2^{-n}, \tag{2}$$

where $\mathsf{Func}(n, 2n)$ is a set of all function that map n bit strings to $2n$ bit strings. From Eqs. 1, 2 we conclude that

$$|\Pr_{s \leftarrow \{0,1\}^n}[\mathsf{D}^{F'_s(\cdot)}(1^n) = 1] \; - \Pr_{f \leftarrow \mathsf{Func}(n,2n)}[\mathsf{D}^{f(\cdot)}(1^n) = 1]| = 1 - 2^{-n}$$

which is clearly not negligible. It follows that F' is not a PRF.

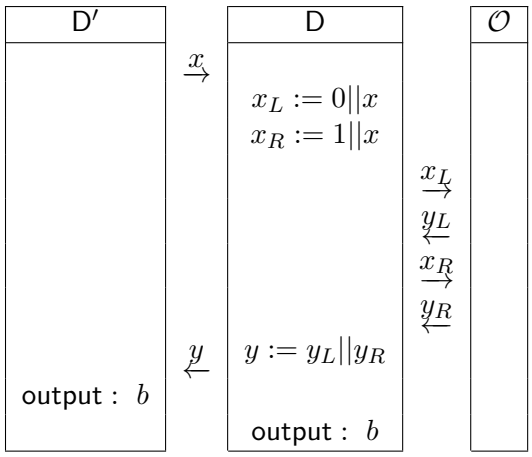
(b) $F'_s(x) := F_s(0 \parallel x) \parallel F_s(1 \parallel x)$, for $F_s \colon \{0,1\}^{n+1} \to \{0,1\}^n$.

Solution:
We prove that this is a PRF. For sake of contradiction, we assume that F'_s is not a PRF. Then there exists a ppt distinguisher D' and a positive polynomial p such that

$$|\Pr_{s \leftarrow_\$ \{0,1\}^n}[\mathsf{D}'^{F'_s(\cdot)}(1^n) = 1] - \Pr_{f' \leftarrow_\$ \mathsf{Func}(n,2n)}[\mathsf{D}'^{f'(\cdot)}(1^n) = 1]| > 1/p(n),$$

where $\mathsf{Func}(n, 2n)$ is a set of all function that map n bit strings to $2n$ bit strings. We use D' to build a distinguisher D which breaks the PRF F as follows:

- On input 1^n and having access to oracle $\mathcal{O}$, D runs the distinguisher D': if D' makes a query $x \in \{0,1\}^n$, then D makes two queries to $\mathcal{O}$: $y_L = \mathcal{O}(0 \parallel x)$ and $y_R = \mathcal{O}(1 \parallel x)$.
- D returns $y = y_L \parallel y_R$ to D' as an answer to the query x.
- When D' makes his guess 0 or 1, then D just repeats it.



First of all we observe that if D' is a ppt algorithm, then also D is a ppt algorithm.
Secondly, we observe that D perfectly simulates the oracle queries for D'.
If $\mathcal{O} = F_s$ for a random key $s \in \{0,1\}^n$, then $y = F'_s(x)$.
If $\mathcal{O} = f$ for a random function $f \colon \{0,1\}^{n+1} \to \{0,1\}^n$, then both y_L and y_R are random n-bit strings and hence y is a random $2n$-bit string. Therefore

$$|\Pr[\mathsf{D}^{F_s(\cdot)}(1^n) = 1] - \Pr[\mathsf{D}^{f(\cdot)}(1^n) = 1]| = |\Pr[\mathsf{D}'^{F'_s(\cdot)}(1^n) = 1] - \Pr[\mathsf{D}'^{f'(\cdot)}(1^n) = 1]| \geq 1/p(n)$$

which is a contradiction with the assumption that F is a PRF.

Here "$\parallel$" denotes concatenation of bit strings.

PSEUDORANDOM PERMUTATION

Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an efficient, length-preserving, keyed permutation. F is a *pseudorandom permutation* if for all probabilistic polynomial-time distinguishers D, there exists a negligible function negl such that:

$$|\Pr[\mathsf{D}^{F_k(\cdot)}(1^n) = 1] - \Pr[\mathsf{D}^{f(\cdot)}(1^n) = 1]| \leq \mathsf{negl}(n)$$

where the first probability is taken over uniform choice of $k \in \{0,1\}^n$ and the randomness of D, and the second probability is taken over uniform choice of $f \in \mathsf{Perm}_n$ and the randomness of D.

Solution:

Concepts to be explained during the exercise session:

- Emphasize that the definition of PRP is analogous to the definition of PRF exept that $f \in \mathsf{Perm}_n$ instead of $f \in \mathsf{Func}_n$.
- Keyed permutation
  Let $F\colon \{0,1\}^{\ell_{key}(n)} \times \{0,1\}^{\ell_{in}(n)} \to \{0,1\}^{\ell_{out}(n)}$ be a keyed function. We call F a *keyed permutation* if $\ell_{in} = \ell_{out}$ and for all $k \in \{0,1\}^{\ell_{key}(n)}$, the function $F_k\colon \{0,1\}^{\ell_{in}(n)} \to \{0,1\}^{\ell_{out}(n)}$ is one-to-one (i.e. F_k is a permutation).
- Length preserving keyed permutation
  A keyed permutation is length preserving if $\ell_{key}(n) = \ell_{in}(n) = n$
- Efficient keyed permutation
  A keyed permutation is efficient if there is a polynomial-time algorithm for computing $F_k(x)$ given k and x, as well as a polynomial-time algorithm for computing $F_k^{-1}(y)$ given y and k.
- Perm_n is a set of all permutations (i.e. bijections) on $\{0,1\}^n$.

Exercise 2 (PRP)

Let n be an even number and assume that $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a PRP. We define a fixed-length encryption scheme $\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows: On input $m \in \{0,1\}^{n/2}$ and key $k \in \{0,1\}^n$, algorithm Enc chooses a uniform string $r \in \{0,1\}^{n/2}$ and computes $c := F_k(r||m)$.

(a) Show how the algorithm Dec works.

Solution:

Since F is a PRP, both F_k and F_k^{-1} has to be computable in polynomial time for every k. The algorithm Dec_k on input c first runs $x := F_k^{-1}(c)$, then it parses $(r, m) := x$ and outputs m.

(b) Prove that this scheme is CPA-secure for messages of length $n/2$.

Solution:
Our proof strategy is similar as the proof strategy for Theorem 3.12. explained during the lecture. That is:

Step 1: Replace PRP with a random permutation

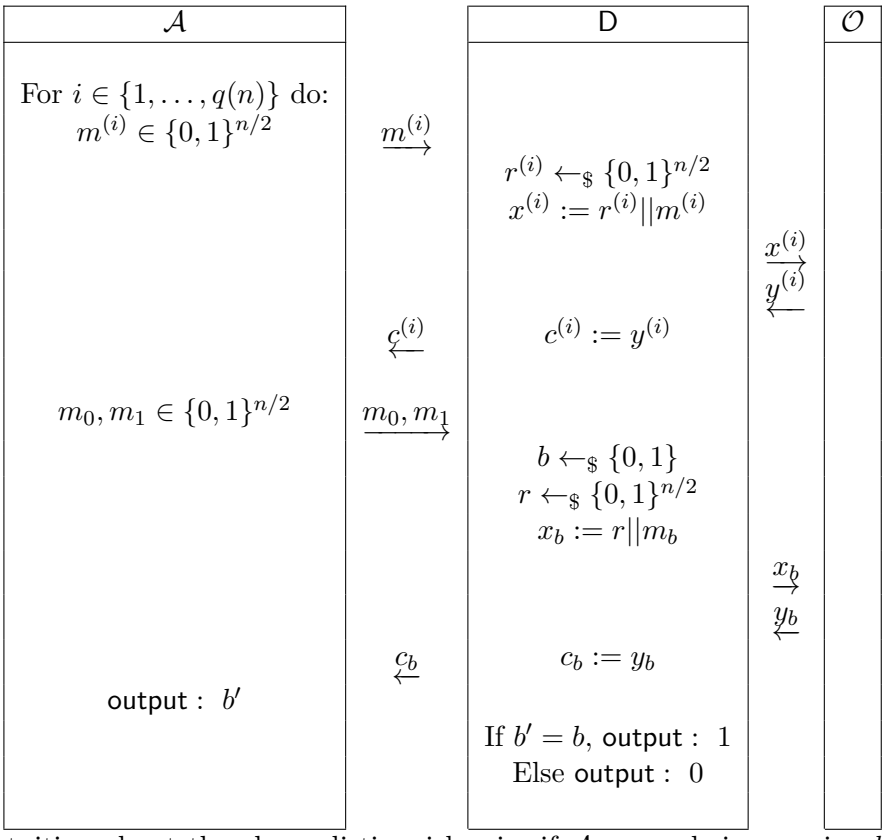
Step 2: Use probabilistic analysis

Step 3: Complete the proof by combining results from Step 1 and 2.

Step 1:
Let $\widetilde{\Pi} = (\widetilde{\mathsf{Gen}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$ be as the encryption scheme Π except that we use $f \leftarrow_{\$} \mathsf{Func}_n$ instead of F_k. Let us fix a PPT adversary $\mathcal{A}$ that makes at most $q(n)$ queries to Enc or $\widetilde{\mathsf{Enc}}$. Our goal is to show that

$$|\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n/2) = 1] - \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1]| \leq \mathsf{negl}(n). \tag{3}$$

We prove the above statement by reduction, i.e. we use $\mathcal{A}$ to build D to break the PRP.



The intuition about the above distinguisher is: if $\mathcal{A}$ succeeds in guessing b, then D guesses that $\mathcal{O}$ was the PRP F_k and otherwise D guesses that $\mathcal{O}$ was a random function f.

- First observe that D is a PPT algorithm if $\mathcal{A}$ is a PPT algorithm.
- Secondly we observe that D simulates $\mathcal{A}$'s environment in the CPA experiment. More precisely, if $\mathcal{O}(\cdot) = F_k(\cdot)$, then the view of the adversary $\mathcal{A}$ is

the same as in the experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n/2)$ which implies

$$\Pr[\mathsf{D}^{F_k(\cdot)}(1^n) = 1] = \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n/2) = 1]. \tag{4}$$

If $\mathcal{O}(\cdot) = f(\cdot)$, then the view of the adversary $\mathcal{A}$ is the same as in the experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n/2)$ which implies

$$\Pr[\mathsf{D}^{f(\cdot)}(1^n) = 1] = \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1]. \tag{5}$$

Since we assume that F is a PRP, Eq. (4) and (5) imply that Eq. (3) holds.

Step 2:
Our goal is to prove that the following equation holds:

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1] \leq \frac{1}{2} + \frac{q(n)}{2^{n/2}} \tag{6}$$

Let $r \in \{0,1\}^{n/2}$ be the random string used to encrypt the challenge message m_b and let `NotFresh` be the event that $r \in \{r^{(1)}, \ldots, r^{(q(n))}\}$, i.e. that the random string was used already before the challenge phase. By law of total probability, we have

$$\begin{aligned}\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1] = &\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1 | \texttt{NotFresh}] \Pr[\texttt{NotFresh}] + \\ &\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1 | \neg\texttt{NotFresh}] \Pr[\neg\texttt{NotFresh}] \\ \leq &\Pr[\texttt{NotFresh}] + \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1 | \neg\texttt{NotFresh}].\end{aligned}$$

The above inequality holds since probability is always within the interval $[0,1]$.
If the string r is fresh, then $f(r||m_b)$ is a random value which means that $\mathcal{A}$ learns nothing about b from the value c_b, formally:

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1 | \neg\texttt{NotFresh}] = \frac{1}{2} \tag{7}$$

The probability that r is not a fresh value is equal to the probability that for $q(n)$ randomly chosen values from $\{0,1\}^{n/2}$ we hit r, which is

$$\Pr[\texttt{NotFresh}] = \frac{q(n)}{2^{n/2}} \tag{8}$$

Eq. (7) and (8) complete the proof of the Eq. (6).

Step 3:
The results from Step 1 and Step 2 give us

$$\Big|\Pr\Big[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n/2) = 1\Big] - \frac{1}{2} - \frac{q(n)}{2^{n/2}}\Big|$$

$$\Big|\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n/2) = 1] - \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n/2) = 1]\Big| \quad \text{|}\wedge \text{ Eq. (6)}$$

$$\mathsf{negl}(n) \quad \text{|}\wedge \text{ Eq. (3)}$$

This implies that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n/2)=1\right] \leq \frac{1}{2}+\frac{q(n)}{2^{n/2}}+\mathsf{negl}(n)=\frac{1}{2}+\mathsf{negl}'(n/2).$$

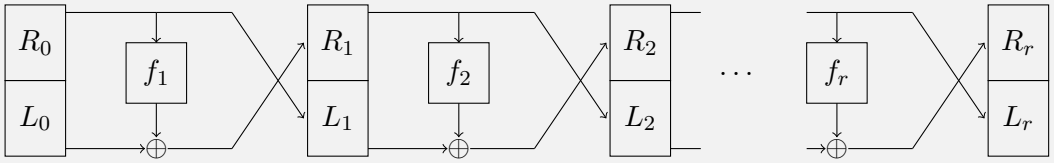
which completes the proof (in the last equality we use the Exercise 2.3.b and 2.3.e)

FEISTEL NETWORKS

As discussed during the lecture, Feistel networks offer another approach for constructing block cipher. A Feistel network operates in r rounds. The input $m \in \{0,1\}^\ell$ is split in two halves, i.e. $L_0||R_0 := m$, where $L_0 \in \{0,1\}^{\ell/2}$ is called the left half and $R_0 \in \{0,1\}^{\ell/2}$ is called the right half of the input. In each round $i \in \{1,\ldots,r\}$, a keyed round function $f_i\colon \{0,1\}^{\ell/2} \to \{0,1\}^{\ell/2}$ is applied in the following manner:

$$L_i := R_{i-1} \in \{0,1\}^{\ell/2}$$
$$R_i := L_{i-1} \oplus f_i(R_{i-1}) \in \{0,1\}^{\ell/2}.$$

The output of the r rounds Feistel network is $c := L_r||R_r \in \{0,1\}^\ell$. See the figure below for pictorial representation of the Feistel network.



Exercise 3 (Inverting Feistel network)

Assume that you know all the round functions $\{f_i\}_{i\in[r]}$. Show how to invert the Feistel network, i.e. knowing $c = L_r||R_r$, show how to compute $m = L_0||R_0$ (do not make any addition assumptions on the round functions f_i).

Solution:

Knowing L_i, R_i and f_i, one can compute L_{i-1}, R_{i-1} as follows:

$$R_{i-1} := L_i$$
$$L_{i-1} := R_i \oplus f_i(R_{i-1}).$$

Note that f_i is evaluated only in the forward direction so it does not need to be invertible!

FEISTEL NETWORK using PRF

Let $F : \{0,1\}^{\ell/2} \times \{0,1\}^{\ell/2} \to \{0,1\}^{\ell/2}$ be a PRF. We can use this function to construct a r-round Feistel network in the following way:

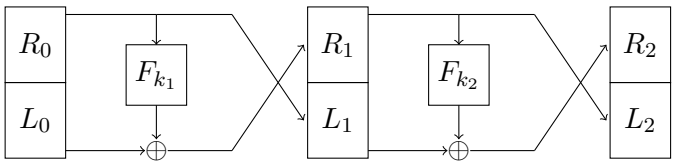
1. Choose $(k_1, \ldots, k_r) \leftarrow_\$ \{0,1\}^{r \times \ell/2}$
2. Define $f_i := F_{k_i}$

Theorem 1 *For $r \geq 3$, the r-round Feistel network constructed using the PRF F as described above is a PRP.*

In one of the homework exercises, we show that this is not true for $r = 2$.

Voluntary homework exercises

Exercise 4 (Two round Feistel network - Voluntary homework 1)
Let $F : \{0,1\}^{\ell/2} \times \{0,1\}^{\ell/2} \to \{0,1\}^{\ell/2}$ be a PRF. Let us denote F': $\{0,1\}^{\ell} \to \{0,1\}^{\ell}$ the 2-round Feistel network constructed using F. Show that F' is **not** a PRP.



Solution:

We construct a distinguisher D as follows: on input 1^ℓ and having oracle access to $\mathcal{O}$, the distinguisher D chooses $L_0^{(a)}, L_0^{(b)}, R_0 \in \{0,1\}^{\ell/2}$ arbitrarily and queries the oracle on:

1. $m^{(a)} := (L_0^{(a)}, R_0)$ and receives an answer $c^{(a)} := (L_2^{(a)}, R_2^{(a)})$;
2. $m^{(b)} := (L_0^{(b)}, R_0)$ and receives an answer $c^{(b)} := (L_2^{(b)}, R_2^{(b)})$.

If $L_0^{(a)} \oplus L_0^{(b)} = L_2^{(a)} \oplus L_2^{(b)}$, then D outputs 1. Otherwise it outputs 0.

We will now prove that the constructed distinguisher can distinguish between $\mathcal{O}$ being the permutation F' or a random permutation $f\mathsf{Perm}_\ell$ with non-negligible probability.

If $\mathcal{O} = F'_{k_1,k_2}$ for some (randomly chosen) k_1, k_2, we have

$$L_2^{(a)} = R_1^{(a)} = L_0^{(a)} \oplus F_{k_1}(R_0)$$
$$L_2^{(b)} = R_1^{(b)} = L_0^{(b)} \oplus F_{k_1}(R_0)$$

This implies that

$$\begin{aligned} L_2^{(a)} \oplus L_2^{(b)} &= L_0^{(a)} \oplus F_{k_1}(R_0) \oplus L_0^{(b)} \oplus F_{k_1}(R_0) \\ &= L_0^{(a)} \oplus L_0^{(b)}. \end{aligned}$$

Hence we have that

$$\Pr_{(k_1,k_2)\leftarrow\{0,1\}^\ell}[\mathsf{D}^{F'_{(k_1,k_2)}(\cdot)}(1^\ell) = 1] = 1.$$

If $\mathcal{O}$ is a truly random permutation, then both $L_2^{(a)}$ and $L_2^{(b)}$ are random strings and so is $L_2^{(a)} \oplus L_2^{(b)}$. This implies that

$$\Pr_{f\leftarrow\mathsf{Perm}_\ell}[\mathsf{D}^{f(\cdot)}(1^\ell) = 1] = 2^{-\ell/2}.$$

We can conclude that

$$\left| \Pr_{(k_1,k_2)\leftarrow\{0,1\}^\ell}[\mathsf{D}^{F'_{(k_1,k_2)}(\cdot)}(1^n) = 1] \; - \; \Pr_{f\leftarrow\mathsf{Perm}_\ell}[\mathsf{D}^{f(\cdot)}(1^\ell) = 1] \right| = 1 - 2^{-\ell/2}$$

which is not a negligible function.

Exercise 5 (PRG from PRF - Voluntary homework 2)

Prove that if $F\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a length-preserving PRF, then

$$G(s) := F_s(1)||F_s(2)||\ldots||F_s(l)$$

is a PRG with expansion factor $l \cdot n$.

Solution:

First of all, since F is length-preserving, the expansion factor is trivially $l \cdot n$.
We will prove now that G is a PRG by reduction. Assume there is a PPT algorithm D that can distinguish between $G(s)$ and a random string $r \in \{0,1\}^{ln}$ with non-negligible probability. We construct in the following an adversary $\mathcal{A}_F$ that can distinguish with non-negligible probability F from a random function.

- Given an input 1^n and access to an oracle $\mathcal{O}$, we compute

  $$t = \mathcal{O}(1)||\mathcal{O}(2)||\ldots||\mathcal{O}(l)$$

  by making l queries to $\mathcal{O}$.
- We simulate D on t and output 1 if and only if D outputs 1.

- It results:

$$
\begin{aligned}
&|\Pr_{s\leftarrow\{0,1\}^n}[\mathcal{A}_F^{F_s(\cdot)}(1^n)=1]-\Pr_{f\leftarrow\mathsf{Func}_n}[\mathcal{A}_F^{f(\cdot)}(1^n)=1]|\\
=&|\Pr_{s\leftarrow\{0,1\}^n}[\mathsf{D}(F_s(1))||\ldots||\mathsf{D}(F_s(l))=1]-\Pr_{f\leftarrow\mathsf{Func}_n}[\mathsf{D}(f(1))||\ldots||\mathsf{D}(f(l))=1]|\\
=&|\Pr_{s\leftarrow\{0,1\}^n}[\mathsf{D}(G(s))=1]-\Pr_{r_1\leftarrow\{0,1\}^n,\ldots,r_l\leftarrow\{0,1\}^n}[\mathsf{D}(r_1||\ldots||r_l)=1]|\\
=&|\Pr_{s\leftarrow\{0,1\}^n}[\mathsf{D}(G(s))=1]-\Pr_{r\leftarrow\{0,1\}^{l\cdot n}}[\mathsf{D}(r)=1]|\\
\geq&\mathsf{negl}
\end{aligned}
$$

  Where the last inequality holds because of the hypothesis on D.

We can therefore conclude that G is a PRG.