# How to construct PRFs

Recommended reading:
KL, Section 7.5
Boneh, Chapter 5

We saw that we can use PRFs to construct CPA-secure encryption.

So, the natural next step is to try to construct PRFs.
In particular, can they be constructed based on objects we have seen so far, or are they fundamentally stronger objects?

It turns out that PRGs (hence OWFs) suffice to construct PRFs. This was proved by Goldreich-Goldwasser-Micali (JACM 1986) https://mit6875.github.io/PAPERS/How_To_Construct_Random_Functions.pdf

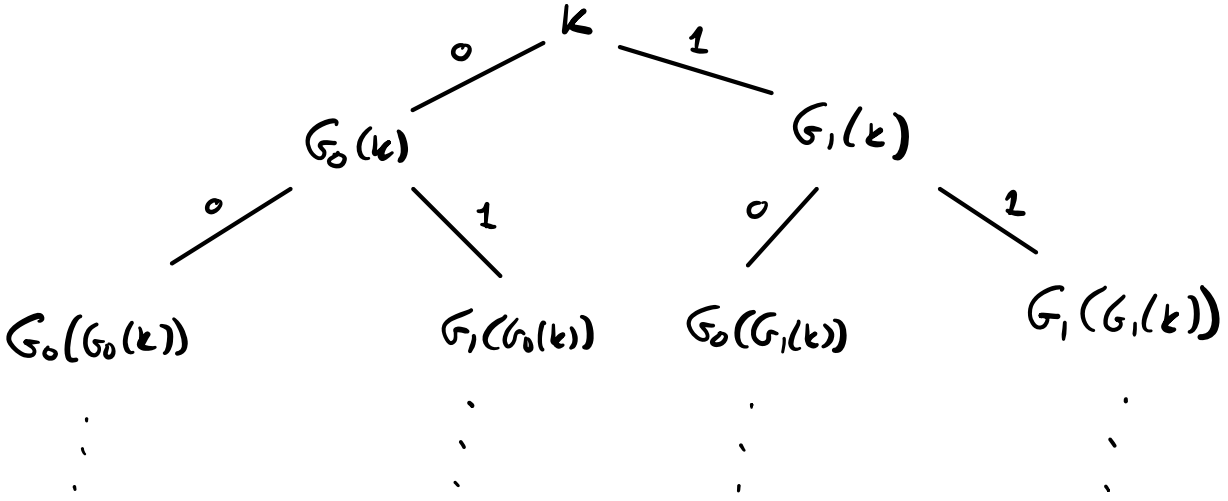
**Thm (GGM):** If PRGs exist then PRFs exist.

**Intuition:** Before we saw how to generate poly(n) many pseudorandom bits from an n-bit seed using a fixed PRG with even just output length n+1. This was done by sequentially applying the PRG to the seed, kind of like a linked list. To get the theorem above, we replace the linked list by a binary tree. :)

Here's the construction: Let $G$ be a PRG with expansion $\ell(n) = 2n$.

For any input $y \in \{0,1\}^n$ we write $G(y) = \underbrace{G_0(y)}_{n \text{ bits}} \| \underbrace{G_1(y)}_{n \text{ bits}}$

For a key $k \in \{0,1\}^n$, consider the tree




At level $n$ there are $2^n$ leaves. We may represent a path from root to a leaf by an $n$-bit string $x \in \{0,1\}^n \to 0 =$ "left", $1 =$ "right"

And this is exactly how we define our PRF $F$:

$$F_k(x) = \text{value at leaf of path def. by } x = G_{x_n}\left(\cdots\left(G_{x_2}\left(G_{x_1}(k)\right)\right)\cdots\right)$$

Before we show that $F$ is a PRF, we need the following lemma, which states that it is hard to distinguish a polynomial number of pseudorandom strings from unif. random strings.

**Lemma:** Let $G$ be a PRG with expansion $\ell(n)$ and $t(n)$ any polynomial. Then, for any PPT adv $A$ there is a negligible function $\varepsilon(n)$ such that

$$\left| \Pr_{s_1,\ldots,s_{t(n)} \overset{iid}{\sim} \{0,1\}^n} \left(A(G(s_1),\ldots,G(s_{t(n)})) = 1\right) - \Pr_{z_1,\ldots,z_{t(n)} \overset{iid}{\sim} \{0,1\}^{\ell(n)}} \left(A(z_1,\ldots,z_{t(n)}) = 1\right) \right| \leq \varepsilon(n).$$

**Proof:** Simple hybrid argument. Homework.

**Proof that $F$ is a PRF:**

We use a hybrid argument. Suppose that $F$ is not a PRF. Then, there exists a PPT adv. $A$ and a polynomial $p(n)$ such that

$$\left| \Pr_{k \leftarrow \{0,1\}^n} \left(A^{\mathcal{O}_{real}}(1^n) = 1\right) - \Pr\left(A^{\mathcal{O}_{ideal}}(1^n) = 1\right) \right| \geq \frac{1}{p(n)}$$

for infinitely many $n$'s. Fix an $n$ for which this holds.

We now define hybrids that slowly morph the interaction $A^{\mathcal{O}_{real,k}}(1^n)$ into $A^{\mathcal{O}_{ideal}}(1^n)$.

For $i \in \{0, \ldots, n\}$, we define the hybrid $H_i$ that, intuitively, replaces all values in tree nodes of $F$ down to level $i$ by independent and uniformly random values.

More precisely, $H_i$ is defined as follows:

- It runs the adversary $A(1^n)$.
- Suppose that $A$ queries the oracle on $x \in \{0,1\}^n$:
  - If the suffix $x[1:i]$ has not been seen before, sample $r_{x_1 \ldots x_i} \leftarrow \{0,1\}^n$ and output

$$G_{x_n}(\ldots(G_{x_{i+1}}(r_{x_1 \ldots x_i}))\ldots)$$

  - If some $x'$ with the same suffix $x'[1:i] = x[1:i]$ has been queried before, use the same $r_{x'_1 \ldots x'_i}$ that was sampled then and output

$$G_{x_n}(\ldots(G_{x_{i+1}}(r_{x'_1 \ldots x'_i}))\ldots)$$

We denote the output of $A$ on this oracle by $A^{O_i}$.

Note that $H_0$ corresponds to the real PRF $F$, so $A^{\mathcal{O}_{real,k}}$, while $H_n$ corresponds to a unif. random function, so $A^{\mathcal{O}_{ideal}}$.

By a hybrid argument, we know that there is an index $i$ such that

$$\left| \Pr_{k \leftarrow \{0,1\}^n}\left(A^{\mathcal{O}_{i-1}}(1^n)=1\right) - \Pr_{k \leftarrow \{0,1\}^n}\left(A^{\mathcal{O}_i}(1^n)=1\right)\right| \geq \frac{1}{n \cdot p(n)}.$$

We will use $A$ and $i$ to distinguish poly-many pseudorand strings from random, contradicting the lemma above.

Let $t(n)$ be an upper bound on the number of oracle queries that $A(1^n)$ makes. We design the following PPT adversary $D$ that distinguishes $t(n)$ pseudorandom $2n$-bit strings from unif. random strings with non-negligible advantage.

On input $1^n$, $y_1, \ldots, y_{t(n)}$, with $y_j = \underbrace{y_{j0}}_{\in \{0,1\}^n} \| \underbrace{y_{j1}}_{\in \{0,1\}^n} \in \{0,1\}^{2n}$,

$D$ behaves as follows:

→ Runs $A(1^n)$

→ When $A$ makes the $j$-th query $x \in \{0,1\}^n$ to the oracle

- If the suffix $x_i, \ldots, x_{i-1}$ appears for the first time, then $D$ sets $r_{x_i \ldots x_{i-1} 0} = y_{j0}$

$$r_{x_i, \ldots, x_{i-1} 1} = y_{j1}$$

and outputs $G_{x_n}(\cdots(G_{x_{i+1}}(r_{x_i \ldots x_{i-1} x_i}))\cdots)$

- If the suffix $x_i, \ldots, x_{i-1}$ already appears in a previous query $x'$, then use the previously set $r_{x'_i \ldots x'_{i-1} 0}$ and $r_{x'_i \ldots x'_{i-1} 1}$ and output

$$G_{x_n}(\cdots(G_{x_{i+1}}(r_{x'_i \ldots x'_{i-1} x_i}))\cdots)$$

→ Outputs whatever $A$ outputs

Note that when $(Y_1, \ldots, Y_{\ell(n)}) = (G(s_1), \ldots, G(s_{\ell(n)}))$ for $s_1, \ldots, s_{\ell(n)} \overset{iid}{\sim} \{0,1\}^n$, then $D$ emulates the oracle $\mathcal{O}_{i-1}$ when interacting with $A$. On the other hand, when $Y_1, \ldots, Y_{\ell(n)} \overset{iid}{\sim} \{0,1\}^{2n}$, then $D$ emulates the oracle $\mathcal{O}_i$ when interacting with $A$. Therefore,

$$\left| \Pr_{s_1, \ldots, s_{\ell(n)} \overset{iid}{\sim} \{0,1\}^n} \left( D(G(s_1), \ldots, G(s_{\ell(n)})) = 1 \right) - \Pr_{z_1, \ldots, z_{\ell(n)} \overset{iid}{\sim} \{0,1\}^{2n}} \left( D(z_1, \ldots, z_{\ell(n)}) = 1 \right) \right|$$

$$= \left| \Pr_{k \leftarrow \{0,1\}^n} \left( A^{\mathcal{O}_{i-1}}(1^n) = 1 \right) - \Pr_{k \leftarrow \{0,1\}^n} \left( A^{\mathcal{O}_i}(1^n) = 1 \right) \right| \geq \frac{1}{n \cdot p(n)},$$

which contradicts the lemma above. ▨