The GGM Construction does NOT yield Correlation Intractable Function Ensembles

Oded Goldreich*
Department of Computer Science
Weizmann Institute of Science
Rehovot, Israel.
oded@wisdom.weizmann.ac.il

July 31, 2002

Abstract

We consider the function ensembles emerging from the construction of Goldreich, Goldwasser and Micali (GGM), when applied to an *arbitrary* pseudoramdon generator. We show that, in general, such functions fail to yield correlation intractable ensembles. Specifically, it may happen that, given a description of such a function, one can easily find an input that is mapped to zero under this function.

Introduction and Statement of Our Result

We consider a special case of the notion of a correlation intractable function ensemble, which was introduced by Canetti, Goldreich and Halevi [CGH98]. Specifically, for any $\ell : \mathsf{N} \to \mathsf{N}$, we consider function ensembles of the form

$$F \stackrel{\mathrm{def}}{=} \{f_s : \{0,1\}^{\ell(|s|)} \to \{0,1\}^{\ell(|s|)}\}_{s\in\{0,1\}^*} \tag{1}$$

Loosely speaking, F is correlation intractable with respect to a binary relation $R \subseteq \cup_k \{0,1\}^{\ell(k)} \times \{0,1\}^{\ell(k)}$ if every feasible adversary, given a uniformly distributed $s \in \{0,1\}^k$, fails to find an $x \in \{0,1\}^{\ell(|s|)}$ such that $(x, f_s(x)) \in R$, except with negligible probability.

Canetti *et. al.* [CGH98] showed that, *unless ℓ is decreasing*, no function ensemble may be correlation intractable w.r.t every adequate relation (for which a random function is correlation intractable). For example, $\{f_s\}$ is not correlation intractable w.r.t the "diagonalization" relation $D = \{(x, f_{x'}(x)) : x \in \{0,1\}^*\}$, where x' is a prefix (of adequate length) of x (i.e., $|x| = \ell(|x'|) \geq |x'|$).

Thus, we focus on function ensembles with monotonically decreasing $\ell : \mathsf{N} \to \mathsf{N}$ (and recall that for $\ell(k) < k/2$ no negative results are known). Furthermore, we will focus on the special case of "constant" relations; that is, relations of the form $R = \{(x,y) : x \in \{0,1\}^* \wedge y \in S \cap \{0,1\}^{|x|}\}$, for some (sparse) set $S \subset \{0,1\}^*$. We investigate natural candidates for function ensembles that are correlation intractable in such a restricted sense.

*Supported by the MINERVA Foundation, Germany.

The failure of pseudorandom functions. One natural candidate for restricted notions of correlation intractability is provided by pseudorandom function ensembles (as defined in [GGM84]). However, these ensembles fail (w.r.t correlation intractability) because no "security" is guaranteed w.r.t adversaries that are given the function's description (i.e., s). Specifically, in general, pseudorandom function ensembles may not be correlation intractable w.r.t some very simple relations (e.g., $R_0 = \{(x, 0^{|x|}) : x \in \{0,1\}^*\}$): The reason being that any pseudorandom function ensemble $\{f_s\}$ can be modified into a pseudorandom function ensemble $\{f'_{r,s}\}$ such that $f'_{r,s}(x) = 0^{|x|}$ if $x = r$ and $f'_{r,s}(x) = f_s(x)$ otherwise.

The failure of the GGM construction. Our aim is to show that a specific (natural) construction of pseudorandom functions (based on pseudorandom generators) fails w.r.t a simple relation (i.e., R_0). Specifically, we refer to the construction of pseudorandom functions due to Goldreich, Goldwasser and Micali [GGM84]. Recall that in their construction, hereafter referred to as the GGM construction, a function $f_s : \{0,1\}^{\ell(|s|)} \to \{0,1\}^{|s|}$ is define based on a pseudorandom generator G such that

$$f_s(x) \stackrel{\text{def}}{=} G_{x_\ell}(G_{x_{\ell-1}}(\cdots G_{x_1}(s)\cdots)) \tag{2}$$

where $G(z) = G_0(z)G_1(z)$, $\ell \stackrel{\text{def}}{=} \ell(|s|)$, and $x = x_1 \cdots x_\ell \in \{0,1\}^\ell$. A length preserving version of f_s is obtained by considering only the $\ell(|s|)$-bit long prefix of $f_s(x)$. Our main result is:

Theorem 1 *If there exists pseudorandom generators then there exists a pseudorandom generator G such that the function ensemble resulting from applying Eq. (2) to G is not correlation intractable with respect to the relation $R_0 = \{(x, 0^{|x|}) : x \in \{0,1\}^*\}$.*

That is, although the resulting function ensemble is pseudorandom (cf. [GGM84]), given the description s of a function in the ensemble, one can find in polynomial-time an input x such that $f_s(x) = 0^{|x|}$. The result can be easily extended to hitting other relations.

Proof of Theorem 1

Motivation

The underlying idea is that 0^ℓ is likely to have a preimage under f_s, and that for a carefully constructed G this preimage is easy to find given s. Intuitively, G is constructed such that either $G_0(s)$ or $G_1(s)$ is likely to have a longer all-zero prefix than s, and it is always the case that either $G_0(s)$ or $G_1(s)$ has an all-zero prefix that is at least as long as the one in s.

For $t = 0, ..., n-1$, let $S_t \stackrel{\text{def}}{=} \{0^t 1 \gamma : \gamma \in \{0,1\}^{n-(t+1)}\}$ be the set of n-bit long strings having a (maximal) all-zero prefix of length t. Let P_t be the set of strings $\alpha\beta \in \{0,1\}^{2n}$ such that $\alpha, \beta \in \cup_{i=0}^t S_i$ and either $\alpha \in S_t$ or $\beta \in S_t$. That is,

$$\begin{aligned} P_t &\stackrel{\text{def}}{=} \left\{\alpha\beta : \alpha, \beta \in (\cup_{i=0}^t S_i) \wedge (\alpha \in S_t \vee \beta \in S_t)\right\} &(3)\\ &= \left\{\alpha\beta : (\alpha, \beta \in S_t) \vee \left(\alpha \in S_t \wedge \beta \in \cup_{i=0}^{t-1} S_i\right) \vee \left(\alpha \in \cup_{i=0}^{t-1} S_i \wedge \beta \in S_t\right)\right\} &(4)\end{aligned}$$

Our aim is to construct a pseudorandom generator G such that for every $t \le \ell$ and $\alpha \in S_t$ it holds that $G(\alpha) \in \cup_{i \ge t} P_i$, and for a constant fraction of $\alpha \in S_t$ it holds that $G(\alpha) \in \cup_{i \ge t+1} P_i$. Intuitively, given $s_\lambda \stackrel{\text{def}}{=} s$ we may find an $x = x_1 \cdots x_\ell$ such that $f_s(x)$ has a all-zero prefix of length $\Omega(\ell)$, by iteratively inspecting both parts of $G(s_{x_1 \cdots x_i})$ for the current $s_{x_1 \cdots x_i}$ and setting x_{i+1} such that $s_{x_1 \cdots x_i x_{i+1}} \stackrel{\text{def}}{=} G_{x_{i+1}}(s_{x_1 \cdots x_i})$ is the part with a longer all-zero prefix.

An abstract construction

To implement and analyze the above idea, we first introduce a random process $\Pi : \{0,1\}^n \to \{0,1\}^{2n}$ with the intention of satisfying the following three properties:

1. $\Pi(U_n) \equiv U_{2n}$, where U_m denotes the uniform distribution on $\{0,1\}^m$.
2. For every $t \le \ell$ and $\alpha \in S_t$, $\Pi(\alpha) \in \cup_{i \ge t} P_i$.
3. For every $t \le \ell$ and $\alpha \in S_t$, $\Pr[\Pi(\alpha) \in \cup_{i \ge t+1} P_i] > c$, where $c > 0$ is a universal constant.

One natural way to define Π is to proceed in iterations, starting with $t = 0$. In each iteration, we map seeds in S_t to outcomes in P_t until P_t gets enough probability mass, and map the residual probability mass to $\cup_{i \ge t+1} P_i$ (first to P_{t+1}, next to P_{t+2}, etc). In order to satisfy Conditions 1 and 2 above, it must hold that, for every t, the fraction of n-bit seeds residing in $\cup_{i=0}^{t} S_i$ must be is at least as big as the fraction of $2n$-bit long outcomes in $\cup_{i=0}^{t} P_i$. In fact, to satisfy Condition 3 the former must be sufficiently bigger than the latter. Actually, Condition 3 follows from the other two conditions.

We now turn to the analysis of the desired process Π. Let $s_t \stackrel{\text{def}}{=} \Pr[U_n \in S_t] = 2^{-(t+1)}$, and $p_t \stackrel{\text{def}}{=} \Pr[U_{2n} \in P_t]$. By Eq. (3)-(4), $p_t = s_t^2 + 2s_t \sum_{i=0}^{t-1} s_i$. The following technical claim will play a key role in our analysis.

Claim 2 *For every $t \ge 0$:*

1. $\sum_{i=0}^{t} p_i = \left(\sum_{i=0}^{t} s_i\right)^2$.
2. $\sum_{i=0}^{t} s_i = \frac{1}{1-2^{-(t+1)}} \cdot \sum_{i=0}^{t} p_i > \left(1 + 2^{-(t+1)}\right) \cdot \sum_{i=0}^{t} p_i$.
3. $\Delta_t \stackrel{\text{def}}{=} \sum_{i=0}^{t} s_i - \sum_{i=0}^{t} p_i > \frac{1}{2} \cdot p_{t+1}$. *Furthermore,* $\Delta_t > (1 - 2^{-t}) \cdot p_{t+1}$.

Part 3 is not used in the actual analysis, and so its proof is moved to the Appendix.

Proof: We first establish Part 1:

$$\begin{aligned}
\sum_{i=0}^{t} p_i &= \sum_{i=0}^{t} \left(s_i^2 + 2s_i \sum_{j=0}^{i-1} s_j\right) \\
&= \sum_{i,j \in \{0,\ldots,t\}} s_i s_j \\
&= \left(\sum_{i=0}^{t} s_i\right)^2
\end{aligned}$$

Combining Part 1 and $\sum_{i=0}^{t} s_i = \sum_{i=0}^{t} 2^{-(i+1)} = 1 - 2^{-(t+1)}$, we get $\sum_{i=0}^{t} s_i = \left(1 - 2^{-(t+1)}\right)^{-1} \cdot \sum_{i=0}^{t} p_i$. Part 2 follows (using $(1-\epsilon)^{-1} > 1 + \epsilon$ for $\epsilon > 0$). ■

Using Claim 2, it follows that by the time we get to deal with seeds in S_t ($t \ge 1$), we have already spend a probability mass of $\sum_{i=0}^{t-1} s_i - \sum_{i=0}^{t-1} p_i > \frac{1}{2} p_t$ towards covering P_t. Thus, some seeds in S_{t-1} are mapped to P_t (or to $\cup_{i \ge t} P_i$). The following claim implies that seeds in S_{t-1} are actually mapped to either P_{t-1} or P_t (but never to $\cup_{i > t} P_i$).

Claim 3 $\sum_{i=0}^{t} s_i = \sum_{i=0}^{t+1} p_i \; - 2^{-(2t+4)} < \sum_{i=0}^{t+1} p_i$

Proof: Using Part 1 of Claim 2, we get:

$$\begin{aligned}\sum_{i=0}^{t+1} p_i &= \left(\sum_{i=0}^{t+1} s_i\right)^2 \\ &= \left(1-2^{-(t+2)}\right)^2 \\ &= 1-2^{-(t+1)}+2^{-(2t+4)} \\ &= 2^{-(2t+4)}+\sum_{i=0}^{t} s_i\end{aligned}$$

■

Defining Π. Given Claims 2 and 3, we explicitly define the process Π. On input $\alpha \in S_0$, with probability $p_0/s_0 = 1/2$, we output a uniformly selected element of P_0, otherwise we output a uniformly selected element of P_1. For $t \geq 1$, on input $\alpha \in S_t$, we first compute $\Delta_{t-1} = \sum_{i=0}^{t-1} s_i - \sum_{i=0}^{t-1} p_i$. (Note that by Claims 2 and 3 it holds that $0 < \Delta_{t-1} < p_t$, and $p_t - \Delta_{t-1} = s_t - \Delta_t < s_t$ follows.) With probability $(p_t - \Delta_{t-1})/s_t$, we output a uniformly selected element of P_t, otherwise we output a uniformly selected element of P_{t+1}. Indeed, $0 < (p_t - \Delta_{t-1})/s_t < 1$. Thus, Π is well-defined.

Note that Π can be implemented in probabilistic polynomial-time. Combining Claims 2 and 3, we get:

Proposition 4 (Π satisfies the desired properties):

1. $\Pi(U_n) \equiv U_{2n}$, *where* U_m *denotes the uniform distribution on* $\{0,1\}^m$.
2. *For every* $t \leq \ell$ *and* $\alpha \in S_t$, $\Pi(\alpha) \in P_t \cup P_{t+1}$.
3. *For every* $t \leq \ell$ *and* $\alpha \in S_t$, $\Pr[\Pi(\alpha) \in P_{t+1}] \geq 1/2$.

Part 3 (which follows from Part 3 of Claim 2) is not used in the actual analysis and is only given for intuition.

Proof: Part 2 is immediate by the construction. It is also clear that $\Pi(U_n)$ is uniform over each of the P_t's. Thus, to prove Part 1 it suffices to show that, for every t, it holds that $\Pr[\Pi(U_n) \in P_t] = p_t$. For $t = 0$, indeed

$$\begin{aligned}\Pr[\Pi(U_n) \in P_0] &= \Pr[U_n \in S_0] \cdot \Pr[\Pi(U_n) \in P_0 | U_n \in S_0] \\ &= s_0 \cdot \frac{p_0}{s_0} = p_0\end{aligned}$$

For $t \geq 1$ (using $\Delta_{-1} \stackrel{\text{def}}{=} 0$ in case $t = 1$), we have

$$\begin{aligned}\Pr[\Pi(U_n) \in P_t] &= \Pr[U_n \in S_t] \cdot \Pr[\Pi(U_n) \in P_t | U_n \in S_t] + \Pr[U_n \in S_{t-1}] \cdot \Pr[\Pi(U_n) \in P_t | U_n \in S_{t-1}] \\ &= s_t \cdot \frac{p_t - \Delta_{t-1}}{s_t} + s_{t-1} \cdot \left(1 - \frac{p_{t-1} - \Delta_{t-2}}{s_{t-1}}\right) \\ &= p_t - \Delta_{t-1} + s_{t-1} - p_{t-1} + \Delta_{t-2} \\ &= p_t\end{aligned}$$

since $\Delta_{t-1} = \Delta_{t-2} + s_{t-1} - p_{t-1}$.

Part 3 follows by noting that for every $\alpha \in S_t$ (with $t \geq 1$),

$$\begin{aligned}
\Pr[\Pi(\alpha) \in P_{t+1}] &= 1 - \frac{p_t - \Delta_{t-1}}{s_t} \\
&= \frac{\sum_{i=0}^{t} s_t - \sum_{i=0}^{t} p_i}{s_t} \\
&> \frac{(1-2^{-t}) \cdot s_t}{s_t} \geq \frac{1}{2}
\end{aligned}$$

where the strict inequality is due to $\Delta_t > (1-2^{-t}) \cdot 2^{-(t+1)} = (1-2^{-t}) \cdot s_t$ (see proof of Part 3 of Claim 2). For $\alpha \in S_0$, $\Pr[\Pi(\alpha) \in P_1] = 1 - (p_0/s_0) = 1/2$. ■

The randomly-labeled tree: We consider a depth ℓ binary tree with nodes labeled by n-bit long strings. The root is labeled with a uniformly selected string, and if a node is labeled with α then its children are labeled with the corresponding parts of $\Pi(\alpha)$. (The root is said to be in level 0 and the 2^ℓ leaves are in level ℓ.)

Using induction on $i = 0, 1..., \ell$ (and relying on Part 1 of Proposition 4), it follows that the nodes at level i are assigned independent uniformly distributed labels. Specifically, suppose that the claim holds for level i, then using Part 1 of Proposition 4 the claim holds for level $i+1$. On the other hand, by Part 2 of Proposition 4, the labels along each path from the root to a leaf belong to S_j's such that the sequence of j is monotonically non-decreasing and increases by at most one unit at each step.

Now, on one hand, with probability $s_0 + s_1 = 3/4$ the (level 0) root has a label in $S_0 \cup S_1$, whereas (on the other hand) with probability $1 - (1-s_\ell)^{2^\ell} = 1 - (1 - 2^{-(\ell+1)})^{2^\ell} > 0.39$ there exists a (level ℓ) leaf with label in S_ℓ. We conclude that, with probability at least $0.39 - 0.25 = 0.14$, the root has label in $S_0 \cup S_1$ and there exist a leaf with a label in S_ℓ. Furthermore, due to the mild-increasing property of the label sequence along each path, the i^{th} intermediate node on the path from the root to this leaf must have a label in $S_i \cup S_{i+1}$.[1] On the other hand, the expected number of level i nodes with label in $S_i \cup S_{i+1}$ is $2^i \cdot (2^{-(i+1)} + 2^{-(i+2)}) = 3/4$. Thus, except with exponentially vanishing probability, level i contains less than n nodes with label in $S_i \cup S_{i+1}$. To summarize, with probability at least 0.13, the following good event holds:

1. The root has label in $S_0 \cup S_1$.
2. There exist a leaf with a label in S_ℓ. Furthermore, the i^{th} intermediate node on the path from the root to this leaf has a label in $S_i \cup S_{i+1}$.
3. For every $i \leq \ell$, level i has at most n nodes that have a label in $S_i \cup S_{i+1}$.

The following search procedure is "geared towards" the above good event.

The (ideal) search procedure: Starting at the root, proceed in a DFS-like manner according to the following rule: *if the currently reached node is at level i and has a level not in $S_i \cup S_{i+1}$ then backtrack immediately else develop it according to the standard DFS-rule.* If we ever reach a leaf having a label in S_ℓ then the search is considered successful.

[1] Recall that a node with label in S_j has children with labels in $\cup_{k=0}^{j+1} S_k$. cannot have a label in $\cup_{k=0}^{i-1} S_k$. Since the root has label in $S_0 \cup S_1$, each node at level i has a label in $\cup_{k=0}^{i+1} S_k$. Furthermore, since the specific leaf on the said path has a label in S_ℓ, the i^{th} intermediate node on the said path cannot have a label in $\cup_{k=0}^{i-1} S_k$.

Assuming that the good event holds, the search is successful. Furthermore, in this case the search has visited at most $2n$ nodes at each level (i.e., the children of parents that were DFS-developed), and so the complexity is bounded by $O(\ell \cdot n)$. In fact, the complexity analysis depends only on the third condition (in the definition of a good event), and thus holds except for with exponentially vanishing probability.

The actual construction

Recall that we have given a probabilistic polynomial-time implementation of Π. We now consider a deterministic polynomial-time algorithm Π' satisfying $\Pi'(\alpha, U_m) \equiv \Pi(\alpha)$, where $m = \text{poly}(|\alpha|)$. Using suitable pseudorandom generators G' (i.e., $G' : \{0,1\}^n \to \{0,1\}^m$) and G'' (i.e., $G'' : \{0,1\}^n \to \{0,1\}^{4n}$), we replace $\Pi' : \{0,1\}^{n+m} \to \{0,1\}^{2n}$ by $\Pi'' : \{0,1\}^{n+2n} \to \{0,1\}^{2\cdot(n+2n)}$ such that

$$\Pi''(\alpha, r'r'') = ((\alpha_1, r_1), (\alpha_2, r_2)) \quad (5)$$

$$\text{where } (\alpha_1, \alpha_2) = \Pi'(\alpha, G'(r')) \text{ and } (r_1, r_2) = G''(r'') \quad (6)$$

That is, $|r_1| = |r_2| = |r'r''|$ and $|r'| = |r''| = |\alpha|$.

Theorem 5 *Let* $G \stackrel{\text{def}}{=} \Pi''$. *Then:*

1. *G is a pseudorandom generator.*

2. *Let $\{f'_s : \{0,1\}^{\ell(|s|)} \to \{0,1\}^{|s|}\}_{s\in\{0,1\}^*}$ be defined by applying Eq. (2) to G, and let $f_s : \{0,1\}^{\ell(|s|)} \to \{0,1\}^{\ell(|s|)}$ be defined by letting $f_s(x)$ equal the $\ell(|s|)$-bit long prefix of $f'_s(x)$. Then $\{f_s\}_{s\in\{0,1\}^*}$ is not correlation intractable with respect to the relation $R_0 = \{(x, 0^{|x|}) : x \in \{0,1\}^*\}$. That is, there exists a probabilistic polynomial-time algorithm that given a uniformly distributed $s \in \{0,1\}^n$, finds with probability at least $1/10$ a string $x \in \{0,1\}^{\ell(|s|)}$ such that $f_s(x) = 0^{\ell(|s|)}$.*

Theorem 1 follows.

Proof: In order to prove Part 1 we first observe that $\Pi'(U_n, U_m) \equiv U_{2n}$. Letting U_n, U'_n, U''_n denote independent random variables each uniformly distributed in $\{0,1\}^n$, we recall that $\Pi''(U_n, U'_nU''_n) = ((Z_1, R_1), (Z_n, R_n))$, where $(Z_1, Z_2) \stackrel{\text{def}}{=} \Pi'(U_n, G'(U'_n))$ and $(R_1, R_2) \stackrel{\text{def}}{=} G''(U''_n)$. Thus, $\Pi''(U_n, U'_nU''_n)$ is computationally indistinguishable from $((Z'_1, R'_1), (Z'_n, R'_n))$, where $(Z'_1, Z'_2) \stackrel{\text{def}}{=} \Pi'(U_n, U_m)$ and (R'_1, R'_2) is uniformly distributed over $\{0,1\}^{2n} \times \{0,1\}^{2n}$. It follows that $G(U_{3n}) \equiv \Pi''(U_n, U'_nU''_n)$ is computationally indistinguishable from $((U'_n, U'_{2n}), (U''_n, U''_{2n}))$. Since G is computable in polynomial-time, and $|G(U_{3n})| = 6n$, Part 1 follows.

In order to prove Part 2, we consider an algorithm that on input $s \in \{0,1\}^{3n}$ invokes the ideal search procedure described above, providing it with labels of an imaginary depth $\ell = \ell(n)$ binary tree as follows. The label of the root is the n-bit long prefix of s, and the $2n$-bit long suffix is called the secret of the root. If an internal node has label $\alpha \in \{0,1\}^n$ and secret $s's'' \in \{0,1\}^{2n}$ then its children will have labels corresponding to the two n-bit long parts of $\Pi'(\alpha, G'(s'))$ and secrets corresponding to the two $2n$-bit long parts of $G''(s'')$. We stress that the search procedure is only given the labels of nodes (at its request), but it is not given the nodes' secrets. Note that the way in which we label the nodes corresponds to the way the function ensemble $\{f_s\}$ is defined (using $G = \Pi''$).

Recall that the search procedure succeeds with probability at least 0.13 on the randomly-label tree, called the ideal setting, where the children of a node labeled by α are assigned labels that corresponding to the two n-bit long parts of $\Pi'(\alpha, U_m)$. Our aim is to show that approximately the same must occur in the real setting described above, where the tree is labeled according to Π'' (or, equivalently, according to $\Pi'(\cdot, G'(\cdot))$ and $G''(\cdot)$). To prove this claim, consider a hybrid setting in which all nodes are associated uniformly distributed secrets (rather than secrets derived by applying G'' to the second part of their parent's secret), and the children of a node labeled by α are assigned labels that corresponding to the two n-bit long parts of $\Pi'(\alpha, G'(s'))$, where s' is the first part of the parent's secret (and the second part is never used). We observe that:

1. The success probability of the search in the ideal setting is approximately the same as its success in the hybrid setting.

   Otherwise, we derive a contradiction to the hypothesis that G' is a pseudorandom generator. Specifically, we will show how to distinguish $n \cdot \ell$ samples of the distribution $G'(U_n)$ from $n \cdot \ell$ samples of the distribution U_m. Given a sequence of samples, we run the search procedure while feeding it with labels generated on-the-fly as follows.

   - The root is assigned a uniformly distributed label, and labels that were assigned to nodes are used whenever the node is visited.
   - When reaching a node (e.g., the root) for the first time, we assign labels to its children by using the next unused sample. Specifically, if the new node has label $\alpha \in \{0,1\}^n$ and the next sample in the input sequence is $s' \in \{0,1\}^m$ then we assign its children (as labels) the corresponding parts of $\Pi'(\alpha, s') \in \{0,1\}^{2n}$.

   Note that when the input sequence is taken from U_m, the above describes the ideal setting, whereas when the input sequence is taken from $G'(U_n)$ we get the hybrid setting.

2. The success probability of the search in the real setting is approximately the same as its success in the hybrid setting.

   Otherwise, we derive a contradiction to the hypothesis that G'' is a pseudorandom generator by considering ℓ additional hybrid settings. For $i = 1, ..., \ell$, the i^{th} hybrid (or i-hybrid) consists of running the search feeding it with labels generated on-the-fly as follows. The label of a node al level $j < i$ is generated as in the hybrid setting; that is, these nodes are assigned uniformly distributed secrets (and the children of such a node labeled by α are assigned labels that corresponding to the two n-bit long parts of $\Pi'(\alpha, G'(s'))$, where s' is the first part of the parent's secret). On the other hand, the label of a node al level $j \geq i$ is generated as in the real setting; that is, these nodes are assigned secrets that are derived from the second part of their parent's secret (and are assigned labels exactly as in case $j < i$). That is, if a node at level $j-1$ has secret $s's''$ then is children are always labeled according to $\Pi'(\alpha, G'(s'))$, whereas the secrets that they are assigned are either uniformly distributed or derived from $G''(s'')$ depending on whether $j < i$ or $j \geq i$. Note that the ℓ-hybrid corresponds to the hybrid setting, whereas the 1-hybrid corresponds to the real setting. Thus, it suffices to show that for every $i \in \{1, ..., \ell - 1\}$, the i-hybrid and $(i+1)$-hybrid are computationally indistinguishable. This is shown by using a potential distinguisher to violate the pseudorandomness of G''.

   Given a distinguisher of the i-hybrid and $(i+1)$-hybrid, we will show how to distinguish $n \cdot \ell$ samples of the distribution $G''(U_n)$ from $n \cdot \ell$ samples of the distribution U_{4n}. Specifically, given a sequence of samples, we run the search procedure while feeding it with secrets and labels generated on-the-fly as follows. When required to provide a label to a newly visited node we always provide the label according to $\Pi'(\alpha, G'(s'))$, where s' is the first part of the parent's secret (and α is the parent's label). The important issue is the generation of secrets:

- Nodes at level $j \leq i$ are assigned uniformly distributed secrets.
- Nodes at level $j \geq i+2$ are assigned secrets according to $G''(s'')$ where s'' is the second part of their parent's secret.
- Nodes at level $i+1$ are assigned secrets (on the fly) that equal the corresponding part of the next unused sample in the input sequence; that is, when a node at level i is first visited, its two children are assigned secrets according to the two parts of the next unused sample.

Note that when the input sequence is taken from U_{4n}, the above describes the $(i+1)$-hybrid, whereas when the input sequence is taken from $G''(U_n)$ we get the i-hybrid (although the secrets at level $i+1$ do not fit the second part of the secrets at level i but rather a re-randomization of the latter).

Combining the above two observations, we conclude that in the real setting the search procedure is successful with probability at least 0.1. Using the correspondence of the real setting to an attack on the function ensemble $\{f_s\}$, Part 2 (and so the entire theorem) follows. ■

Acknowledgments

The question was originally posed by Silvio Micali (in the early 1990's if I recall correctly), and re-posed by Boaz Barak in Summer 2001. I am grateful to both of them.

References

[CGH98] R. Canetti, O. Goldreich and S. Halevi. The Random Oracle Methodology, Revisited. In *30th STOC*, pages 209–218, 1998.

[GGM84] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *JACM*, Vol. 33, No. 4, pages 792–807, 1986.

Appendix

Proof of Part 3 of Claim 2: Using Part 2, we have

$$\begin{aligned}
\sum_{i=0}^{t} s_i - \sum_{i=0}^{t} p_i &> 2^{-(t+1)} \cdot \sum_{i=0}^{t} p_i \\
&= 2^{-(t+1)} \cdot \left(\sum_{i=0}^{t} s_i\right)^2 \\
&= 2^{-(t+1)} \cdot \left(1 - 2^{-(t+1)}\right)^2 \\
&> 2^{-(t+1)} \cdot \left(1 - 2^{-t}\right)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
p_{t+1} &= s_{t+1}^2 + 2s_{t+1} \sum_{i=0}^{t} s_i \\
&= s_{t+1} \cdot \left(s_{t+1} + 2\sum_{i=0}^{t} s_i\right) \\
&= 2^{-(t+2)} \cdot \left(2^{-(t+2)} + 2 \cdot \left(1 - 2^{-(t+1)}\right)\right) \\
&= 2^{-(t+1)} \cdot \left(1 - 2^{-(t+1)} + 2^{-(t+3)}\right) \\
&= 2^{-(t+1)} \cdot \left(1 - \frac{3}{8} \cdot 2^{-t}\right)
\end{aligned}$$

Thus,

$$\begin{aligned}
\Delta_t = \sum_{i=0}^{t} s_i - \sum_{i=0}^{t} p_i &> \frac{2^{-(t+1)} \cdot (1 - 2^{-t})}{2^{-(t+1)} \cdot (1 - \frac{3}{8} \cdot 2^{-t})} \cdot p_{t+1} \\
&= \frac{1 - 2^{-t}}{1 - \frac{3}{8} \cdot 2^{-t}} \cdot p_{t+1} \\
&= \left(1 - \frac{\frac{5}{8} \cdot 2^{-t}}{1 - \frac{3}{8} \cdot 2^{-t}}\right) \cdot p_{t+1} \\
&> \left(1 - \frac{\frac{5}{8} \cdot 2^{-t}}{1 - \frac{3}{8}}\right) \cdot p_{t+1} \\
&= \left(1 - 2^{-t}\right) \cdot p_{t+1}
\end{aligned}$$

Thus, $\Delta_t > \frac{1}{2} p_{t+1}$, provided $t \geq 1$. For $t = 0$, we note that $\Delta_0 = s_0 - p_0 = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$ whereas $p_1 = \frac{5}{16}$ and so $\Delta_0 = \frac{4}{5} \cdot p_1$. Part 3 follows. ■