Pseudorandom generators (PRG) stretch a short random string to a longer string which is hard for a computationally limited machine to tell the difference between the uniform string. As a cryptographic primitive, PRGs play important roles in the many applications. However, natural construction for a PRG is not known. Meanwhile, there are lots of natural problems as the candidates for another primitive, one-way function, which can be computed efficiently but hard to invert.

So the construction of PRGs from OWFs is an interesting problem\(^1\). There are lots of results under special assumption of the structural properties of OWFs. The paper by Impagliazzo, Levin and Luby gave the first construction without any requirement for the OWFs, however they use a non-uniform reduction to get the result. Soon, Håstad gave a uniform proof. Altogether, we have the following theorem:

**Theorem 1.** *If there exists a one-way function, there is a pseudorandom generator.*

In this report we give the most important part of the construction in HILL paper: the construction of a “false entropy generator” from any one-way function using “small” advice, where “FEG” is a generalization of pseudorandom generator and the small advice means a logarithmic length non-uniform advice. Formally,

**Definition 1 (False Entropy Generator).** Let \(G : \{0,1\}^n \to \{0,1\}^\ell(n)\) and \(D\) be the distribution of the output of \(G\) with uniform input. \(G\) is a false entropy generator if there exists a distribution \(E\) such that

- \(D\) and \(E\) are computationally indistinguishable
- \(\text{Ent}(E) > \text{Ent}(D)\)

where \(\text{Ent}(.)\) denotes the entropy of the distribution.

## 1 Preliminaries

In this section, we state some notations, basic construction and results without proof. We will focus on the mildly non-uniform proof of the Theorem 2 in Section 2.

**Definition 2.** Let \(f : \{0,1\}^n \to \{0,1\}^\ell(n)\) be a function. Define function \(d : \{0,1\}^\ell(n) \to \{0,1,\cdots,n-1\}\) as

\[
d(f(x)) = \lfloor \log(\#\text{pre}(f(x))) \rfloor
\]

Define the small advice

\[
\lambda_n = \Pr_{I,X}[I \leq d(f(X))]
\]

Set some parameters \(k(n) \geq 125n^3\) and \(m(n) = \lambda_n k(n) - 2k(n)^2\).

\(^1\)From the complexity point of view, the existence of one-way functions and pseudorandom generators are based on the assumption that \(P \neq \text{NP}\).
Let $h$ be a universal hash function mapping $n$-bit string to $\ell(n) + 2\lceil\log n\rceil$-bit string. We construct $f'$ as

$$f'(x, i, y) = \langle f(x), h_y(x)\{i, 2, \ldots, i+2\lceil\log n\rceil\}, y, i \rangle$$

where $i \in \{0, 1, \cdots, n-1\}$, $y \in \{0, 1\}^{p(n)}$ is the index of the hash function and "$,>$" denotes the concatenation. We have

**Lemma 1.1.** If $f$ is a one-way function and $f'$ is constructed above

- If $i \leq d(f(x))$, $f'$ is also a one-way function, i.e., given $f'(x, i, y)$, it is hard to get $x$ for a polynomial time bounded machine.
- If $i \geq d(f(x))$, $x$ is determined by $f'(x, y, i)$ with probability at least $1 - 1/2n$.

**Lemma 1.2.** Let $W = (X, I) \in_R T = \{(x, i) \mid x \in \{0, 1\}^n, i \in \{0, 1, \cdots, n-1\}, i \leq d(f(x))\}$, the following two distribution

$$D_1 = \langle f'(W, Y), X \cdot Z, Z \rangle$$

$$D_2 = \langle f'(W, Y), \beta, Z \rangle$$

are computationally indistinguishable where $Y \in_R \{0, 1\}^{p(n)}, Z \in_R \{0, 1\}^n, \beta \in_R \{0, 1\}$

Having these two lemma, we construct the function $g$ as

$$g(X, Y, I, Y', Z) = \langle f'^{k(n)}(X, I, Y), h'_{Y'}(X \cdot Z), Z, Y' \rangle$$

where $X, Z \in \{0, 1\}^{k(n) \times n}, I \in \{0, 1, \cdots, n-1\}^{k(n)}, Y \in \{0, 1\}^{k(n) \times p(n)}$, and $h'$ is a universal hash function mapping $k(n)$ bits to $m(n)$ bits with index $Y' \in \{0, 1\}^{p(n)}$. We will show that

**Theorem 2.** Let $f$ is a one-way function, $g$ is a false entropy generator.

## 2 Proof of the Main Theorem

By the definition of false entropy generator, let $D$ be in distribution of the output of $g$ with the uniform inputs and let $E$ be the distribution exactly the same as $D$ except for replacing the hashed by $m(n)$ truely random bits, formally

$$D = \langle f'^{k(n)}(X, I, Y), h'_{Y'}(X \cdot Z), Z, Y' \rangle$$

$$E = \langle f'^{k(n)}(X, I, Y), U_{m(n)}, Z, Y' \rangle$$

(1)

(2)

Theorem 2 is proved by the following two claims

**Claim 2.1.** $\text{Ent}(E) \geq \text{Ent}(D) + 10n^2$.

**Claim 2.2.** $D$ is computationally indistinguishable from $E$.

**Proof of Claim 2.1.** Firstly, we give the definition of conditional entropy

**Definition 3 (Conditional Entropy).** Let $E_1$ and $E_2$ be two distribution over $S$, $X_1$ and $X_2$ be two random variable chosen from $S$ with respect to $E_1$ and $E_2$ respectively. The entropy of $E_1$ conditional on $E_2$ is defined as

$$\text{Ent}(E_1 \mid E_2) = E_{x \in E_2}S[\text{Ent}(E_1 \mid X_2 = x)]$$

2
Consider the distributions $\mathcal{E}$ and $\mathcal{D}$, the only difference is the hashed bits and uniform bits, so

$$
\text{Ent}(\mathcal{E}) - \text{Ent}(\mathcal{D}) = m(n) - \text{Ent}(h'_Y(X \cdot Z) \mid f'(X, I, Y), Z, Y') \\
\geq m(n) - \text{Ent}((X \cdot Z) \mid f'(X, I, Y), Z)
$$

where

$$
\begin{align*}
\text{Ent}((X \cdot Z) \mid f'(X, I, Y), Z) &= k(n)\text{Ent}((X_l \cdot Z_j) \mid f'(X_l, I_j, Z_j), Z_j) \\
&= k(n)(\lambda_n - \frac{1}{n})\text{Ent}((X_l \cdot Z_j) \mid f'(X_l, I_j, Z_j), Z_j \text{ and } I_j < d(f(x))) \\
&\quad + k(n)(1 - \lambda_n + \frac{1}{n})\text{Ent}((X_l \cdot Z_j) \mid f'(X_l, I_j, Z_j), Z_j \text{ and } I_j \geq d(f(x))) \\
&\leq k(n) \left( (\lambda_n - \frac{1}{n}) \cdot 1 + (1 - \lambda_n + \frac{1}{n}) (1 - \frac{1}{2n}) \cdot 0 + \frac{1}{2n} \cdot 1 \right) \\
&\leq k(n) \left( \lambda_n - \frac{1}{2n} \right)
\end{align*}
$$

where the first inequality is followed from Lemma 1.1. And by our setting of parameters, we get

$$
\text{Ent}(\mathcal{E}) - \text{Ent}(\mathcal{D}) \geq 10n^2
$$

In order to prove Claim 2.2, we show that

**Lemma 2.3.** Let $A$ be the adversary which distinguishes $\mathcal{D}$ and $\mathcal{E}$ with probability

$$
\delta(n) = \Pr[A(\mathcal{D}) = 1] - \Pr[A(\mathcal{E}) = 1]
$$

there exists an oracle adversary $M^A$ which distinguishes $\mathcal{D}_1$ and $\mathcal{D}_2$ with probability $\rho = \delta(n)/16k(n)$, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are defined in Lemma 1.2.

It is clear that if $f$ is one-way function, the adversary $A$ does not exists, hence $\mathcal{D}$ and $\mathcal{E}$ are computationally indistinguishable.

### 2.1 Inefficient Machine $M^A$

We will first give an inefficient construction of the oracle adversary $M^A$, where the "inefficiency" means the adversary runs in polynomial time, uses the non-uniform advice $\lambda_n$ and has the ability to sample points from $T(\overline{T})$ efficiently.\(^2\)

By the definition, in each application of the function $f'$, we choose $w_j = (X_j, I_j)$ from the set $U = \{0, 1\}^n \times \{0, 1, \ldots, n - 1\}$. The advice $\lambda_n$ tells us that with probability exactly $\lambda_n$ (resp. $1 - \lambda_n$), the $w_j$ is chosen from $T$ (resp. $\overline{T}$). So we can view the distributions $\mathcal{D}$ and $\mathcal{E}$ as\(^3\)

---

\(^2\)Notice that the set $T = \{(x, i) \mid i \leq d(f(x))\}$ might be exponentially large even given the advice $\lambda_n$ ( $\lambda_n$ is defined as the probability over all $x$ and $i$ such that $i \leq d(f(x))$).

\(^3\)Clearly, in these alternative views of the distributions $\mathcal{D}$ and $\mathcal{E}$, they are not efficiently constructed because we need sample the set $T$ and $\overline{T}$. It is this reason that leads to the construction of our adversary inefficient.
Algorithm 1: An Alternative View of Distribution $\mathcal{D}$ (resp. $\mathcal{E}$)

1. for $j = 1$ to $k(n)$
2. Let $c_j = \begin{cases} 1, & \text{with probability } \lambda_n; \\ 0, & \text{with probability } 1 - \lambda_n. \end{cases}$
3. if $c_j = 1$ then Choose $w = (x, i) \in R T$
4. else Choose $w = (x, i) \in R T = \{(x, i) \mid i > d(f(x))\}$
5. Let $X_j = x, I_j = i$ and $Y_j \in \{0, 1\}^{\nu(n)}, Z_j \in \{0, 1\}^n$
6. Compute $f'(X_j, I_j, Y_j)$ and $X_j \cdot Z_j$
7. For $\mathcal{D}$: Choose $Y' \in R \{0, 1\}^{\nu(n)}$ and hash $X \cdot Z$ by $h'_y$.
8. For $\mathcal{E}$: Choose $U_m \in R \{0, 1\}^m(n)$
9. Put the resulting bits in the position according to Eqn.(1).

The high level picture of the proof is to construct a new distribution $\mathcal{D}'$ which is statistically close to $\mathcal{E}$ and computationally indistinguishable from $\mathcal{D}$. The idea to construct $\mathcal{D}'$ comes from the Lemma 1.2, i.e., $x \cdot z$ looks like a random bit given $f'(x, i, y), z$ and $i \leq d(f(x))$. Therefore,

Algorithm 2: The Distribution $\mathcal{D}'$

1. for $j = 1$ to $k(n)$
2. Let $c_j = \begin{cases} 1, & \text{with probability } \lambda_n; \\ 0, & \text{with probability } 1 - \lambda_n. \end{cases}$
3. if $c_j = 1$ then Choose $w = (x, i) \in R T$
4. else Choose $w = (x, i) \in R \overline{T} = \{(x, i) \mid i > d(f(x))\}$
5. Let $X_j = x, I_j = i$ and $Y_j \in \{0, 1\}^{\nu(n)}, Z_j \in \{0, 1\}^n$
6. Compute $f'(X_j, I_j, Y_j)$ and $X_j \cdot Z_j$
7. if $c_j = 1$ then replace $X_j \cdot Z_j$ by $\beta_j \in_r \{0, 1\}$
8. Choose $Y' \in R \{0, 1\}^{\nu(n)}$ and hash $X \cdot Z$ by $h'_y$.
9. Put the resulting bits in the position according to Eqn.(1).

Note that the only difference between $\mathcal{D}$ and $\mathcal{D}'$ is at the line (7).

Claim 2.4. $\mathcal{D}'$ and $\mathcal{E}$ are statistically close.

Proof Sketch: Consider the bits before hashing in $\mathcal{D}'$, it is clear that the entropy of these bits is at least the number of truly random bits which is the size of set $\{c_j \mid c_j = 1, 1 \leq j \leq k(n)\}$. We know that $\Pr[c_j = 1] = \lambda_n$, so with very high probability $|\{c_j \mid c_j = 1, 1 \leq j \leq k(n)\}|$ will concentrate on $\lambda_n k(n) - k(n)^{\frac{3}{2}}$ by the Chernoff bound. According to the hash smoothing lemma and $m(n) = \lambda_n k(n) - 2 k(n)^{\frac{3}{2}}$, the hashed bits is statistically close to $U_m(n)$.

Since $\mathcal{D}' \sim \mathcal{E}$, the adversary $A$ in Lemma 2.3 would also distinguish $\mathcal{D}$ and $\mathcal{D}'$. By the hybrid argument, we have

Claim 2.5. There exists an oracle adversary $M^A$ which distinguishes $\mathcal{D}_t$ and $\mathcal{D}$ with probability $\delta(n)/k(n)$.

Proof Sketch: Let $\mathcal{H}^{(0)} = \mathcal{D}$, we define a sequence of distributions $\mathcal{H}^{(t)}$ for $1 \leq t \leq k(n)$. For $\mathcal{H}^{(t)}$, we construct it using the algorithm
Algorithm 3: The Construction of Distribution \( \mathcal{H}(t) \)

(1) for \( j = 1 \) to \( k(n) \)
(2) Let \( c_j = \begin{cases} 1, & \text{with probability } \lambda_n; \\ 0, & \text{with probability } 1 - \lambda_n. \end{cases} \)
(3) if \( c_j = 1 \) then Choose \( w = (x, i) \in R T \)
(4) else Choose \( w = (x, i) \in R T = \{(x, i) \mid i > d(f(x))\} \)
(5) Let \( X_j = x, I_j = i \) and \( Y_j \in R \{0, 1\}^n \), \( Z_j \in R \{0, 1\} \)
(6) Compute \( f'(X_j, I_j, Y_j) \)
(7) if \( j \leq t \) and \( c_j = 1 \) then replace \( X_j \cdot Z_j \) by \( \beta_j \in r \{0, 1\} \)
(8) Choose \( U_{m(n)} \in R \{0, 1\}^m(n) \)
(9) Put the resulting bits in the position according to Eqn.(2).

In words, \( \mathcal{H}(t) \) is constructed as \( D' \) when \( j \leq t \) and as \( D \) when \( j > t \). So \( \mathcal{H}(k(n)) = D' \), hence

\[
\Pr[A(\mathcal{H}(0)) = 1] - \Pr[A(\mathcal{H}(k(n))) = 1] = \delta(n)
\]

and

\[
E_j[\Pr((\mathcal{H}(j-1)) = 1] - \Pr[A(\mathcal{H}(j)) = 1] = \frac{\delta(n)}{k(n)}
\]

Therefore we construct the oracle adversary \( M^A \) as

Algorithm 4: Non-Uniform Adversary for \( D_1 \) and \( D_2 \)

Input: \( f'(\tilde{x}, \tilde{i}, \tilde{y}), \tilde{b}, \tilde{z} \).

Output: 1/0

(1) Choose \( j \) uniformly from \( \{1 \cdots k(n)\} \);
(2) Let \( c_j = \begin{cases} 1, & \text{with probability } \lambda_n; \\ 0, & \text{with probability } 1 - \lambda_n. \end{cases} \)
(3) if \( c_j = 0 \) then return \( \beta \in r \{0, 1\} \)
(4) Modify the “\( j \)th position” of \( \mathcal{H}^j \) according to the input
(5) \( X_j \mapsto \tilde{x}, Z_j \mapsto \tilde{z}, Y_j \mapsto \tilde{y} \)
(6) \( X_j \cdot Z_j \mapsto \tilde{b} \)
(7) Sample a point \( \Pi \) from the modified distribution
(8) return \( A(\Pi) \)

Note that this machine is inefficient because we need to sample point from \( T \) which could be exponentially large. We claim that this adversary does distinguish \( D_1 \) and \( D_2 \).

According to the construction, if \( c_j = 1 \), the \( j \)th bit to the hash function is \( \tilde{b} \) from the input. So when the input is sampled w.r.t. \( D_1 \), \( \tilde{b} \) is indeed the inner product bit of \( \tilde{x} \) and \( \tilde{z} \), the result distribution is identical to \( \mathcal{H}(j-1) \). When the input is sampled w.r.t. \( D_2 \), the new distribution is exactly the same as \( \mathcal{H}(j) \). So

\[
\Pr[M^A(D_1) = 1] - \Pr[M^A(D_2) = 1] = \Pr[A(\mathcal{H}(j-1)) = 1] - \Pr[A(\mathcal{H}(j)) = 1] = \frac{\delta(n)}{k(n)}
\]

\( \square \)
2.2 Efficient Construction of $M^A$

As stated earlier, the inefficiency in above reduction is coming from the polynomial-time non-samplable set $T$. In this section we construct a efficient machine $M^A$ which only uses the small advice $\lambda_n$. We call this reduction as *mildly non-uniform reduction.*

In the straightforward hybrid argument, the distribution sequence $\mathcal{H}^{(j)}$’s is progressively from $\mathcal{D}$ to $\mathcal{D}'$. For randomly chosen a $\mathcal{H}^{(j)}$, the “distance” to its neighbor is polynomially related to the distance between $\mathcal{D}$ and $\mathcal{D}'$. It is this “distance” that make our $M^A$ break $\mathcal{D}_1$ and $\mathcal{D}_2$. However, those $\mathcal{H}^{(j)}$’s are all polynomial-time un-samplable because during the construction of $\mathcal{H}^{(j)}$, we have to sample point from the set $T$ or $\overline{T}$) according to the value of $c_j$.

As efficient adversary, our new machine first will produce a sequence of polynomial-time samplable distribution pair $(\mathcal{D}^j, \mathcal{E}^j)$ for $1 \leq j \leq k(n)$. The construction of the new distribution $\mathcal{D}^j$ uses the same framework as $\mathcal{H}^{(j)}$ except for selecting a point $w$ from $U = \{0,1\}^n \times \{0,1,\cdots,n-1\}$, which is polynomial-time samplabe, and fix it. The $\mathcal{E}^j$ is same as $\mathcal{D}^j$ except for replacing the hashed bits by uniform bits. The problem is how to choose the point $w$. Let $\delta^j = \Pr[A(\mathcal{D}^j) = 1] - \Pr[A(\mathcal{E}^j) = 1]$, we hope there exists an almost same gap between $\delta^j$ and $\delta^{j+1}$ for a randomly chosen $j$. Intuitively our solution is to choose a point $w$ such that maximize the $\delta^j$.

Formally, the oracle adversary $M^A$ consists of two phase. There are $k(n)$ steps in the Phase I which produce $k(n)$ pairs of polynomial-time samplable distributions $(\mathcal{D}^j, \mathcal{E}^j)$ for $1 \leq j \leq k(n)$ with $\mathcal{D}^0 = \mathcal{D}$ and $\mathcal{E}^0 = \mathcal{E}$. For the easy of later proof, we consider the $j + 1$st step:

**Algorithm 5:** The $j + 1$st Step of Phase I

**Input:** The Distributions $\mathcal{D}^j$ and $\mathcal{E}^j$

**Output:** The Distributions $\mathcal{D}^{j+1}$ and $\mathcal{E}^{j+1}$

**$M(A)$**

1. Let $c_{j+1} = \begin{cases} 1, & \text{with probability } \lambda_n; \\ 0, & \text{with probability } 1 - \lambda_n. \end{cases}$
2. for $s = 1$ to $t = 64n^2/\rho$

3. Sample a point $\hat{w}_s = (\hat{x}, \hat{i})$ uniformly from $U$;
4. Let $\mathcal{D}^j_{c_{j+1}}(\hat{w}_s)$ be modified version of $\mathcal{D}^j$ where

5. $X_j \mapsto \hat{x}, I_j \mapsto \hat{i}, X_j \cdot Z_j \mapsto \hat{x} \cdot Z_j$
6. if $c_{j+1} = 1$ then replace $\hat{x} \cdot Z_j$ by $\beta \in_R \{0,1\}$
7. Let $\mathcal{E}^j(\hat{w}_s)$ be modified version of $\mathcal{E}^j$ where

8. $X_j \mapsto \hat{x}, I_j \mapsto \hat{i}$
9. Let $\delta_{c_{j+1}}^j(\hat{w}_s) = \Pr[A(\mathcal{D}^j_{c_{j+1}}(\hat{w}_s)) = 1] - \Pr[A(\mathcal{E}^j(\hat{w}_s)) = 1]$
10. Sample polynomial times from $\mathcal{D}^j_{c_{j+1}}(\hat{w}_s)$ and $\mathcal{E}^j(\hat{w}_s)$ and get approximation $\Delta^j_{c_{j+1}}(\hat{w}_s)$ of $\delta_{c_{j+1}}^j(\hat{w}_s)$ such that

$$\Pr[|\Delta^j_{c_{j+1}}(\hat{w}_s) - \delta_{c_{j+1}}^j(\hat{w}_s)| \leq \rho] \geq 1 - 2^{-n}$$

11. Let $s_0$ be the index such that

$$\Delta^j_{c_{j+1}}(\hat{w}_{s_0}) = \max_{1 \leq s \leq t} \Delta^j_{c_{j+1}}(\hat{w}_s)$$

and let $\mathcal{D}^{j+1} = \mathcal{D}^j_{c_{j+1}}(\hat{w}_{s_0})$ and $\mathcal{E}^{j+1} = \mathcal{E}^j(\hat{w}_{s_0})$

Note that we view $c_{j+1}$ as a value since line (1). The phase II of the adversary is an algorithm
which distinguishes $D_1$ and $D_2$.

**Algorithm 6: Phase II of $M^A$**

**Input:** $f'(\tilde{x}, \tilde{i}, \tilde{y}), \tilde{b}, \tilde{z}$.

**Output:** $1/0$

1. Choose $j$ randomly from $\{0, 1, \ldots, n-1\}$;
2. Let $D_j(\tilde{w}, \tilde{b}, \tilde{y}, \tilde{z})$ be the distribution exactly the same as $D_j$ except for the “$j+1$ position” $W_{j+1}, Z_{j+1}, Y_{j+1}$;
3. Let $\tilde{y}$ be the same as $\tilde{y}$;
4. Replace the $j+1$st bit to hash function $h'$ by $\tilde{b}$;
5. Let $E_j(\tilde{w}, \tilde{y}, \tilde{z})$ be the distribution exactly the same as $E_j$ except for the “$j+1$ position” $W_{j+1}, Z_{j+1}, Y_{j+1}$;
6. Sample $\Pi$ w.r.t. $D_j(\tilde{w}, \tilde{b}, \tilde{y}, \tilde{z})$ and $\Sigma$ w.r.t. $E_j(\tilde{w}, \tilde{y}, \tilde{z})$;
7. if $A(\Pi) = A(\Sigma)$ then return $\beta \in \{0, 1\}$
else return $A(\Pi)$

Let $d_j(\tilde{w}, \tilde{b}, \tilde{y}, \tilde{z}) = \Pr[A(D_j(\tilde{w}, \tilde{b}, \tilde{y}, \tilde{z})) = 1]$ where the probability is over and $e_j(\tilde{w}, \tilde{y}, \tilde{z}) = \Pr[A(E_j(\tilde{w}, \tilde{y}, \tilde{z})) = 1]$, it is easy to get

\[
\Pr[M^A(f'(\tilde{x}, \tilde{i}, \tilde{y})) = 1] = \frac{1}{2} \Pr[A(\Pi) = A(\Sigma)] + \Pr[A(\Pi) = 1 \text{ and } A(\Sigma) = 0] + \frac{1}{2} (d_j(\tilde{w}, \tilde{b}, \tilde{y}, \tilde{z}) - e_j(\tilde{w}, \tilde{y}, \tilde{z}))
\]

And using the notation in the Phase I, we have

\[
\begin{align*}
\delta_j^0(\tilde{w}) &= d_j(\tilde{w}, \tilde{X}, \tilde{Z}, \tilde{Y}, \tilde{Z}) - e_j(\tilde{w}, \tilde{Y}, \tilde{Z}) \\
\delta_j^1(\tilde{w}) &= d_j(\tilde{w}, \beta, \tilde{Y}, \tilde{Z}) - e_j(\tilde{w}, \tilde{Y}, \tilde{Z})
\end{align*}
\]

where $\tilde{X}, \tilde{Z}, \tilde{Y}$ are all uniform. Hence

\[
\begin{align*}
\Pr[D_1|M^A(D_1) = 1] - \Pr[D_2|M^A(D_2) = 1] &= \frac{1}{2} \mathbb{E}_j[\delta_j^0(W) - \delta_j^1(W)] \\
&= \frac{1}{2} \mathbb{E}[\epsilon^j]
\end{align*}
\]

where $W$ is uniformly distributed on $T$ and $\epsilon^j$ is defined as $\mathbb{E}_W[\delta_j^0(W) - \delta_j^1(W)]$. We only need to prove that $\mathbb{E}[\epsilon^j] \geq 2\rho$ which is equivalent to prove

\[
\mathbb{E}\left[\sum_{j=0}^{k(n)-1} \epsilon^j\right] \geq \frac{\delta(n)}{8}\tag{3}
\]

where the expectation is over the random choices of $M^A$ in the first phase.
Claim 2.6. If $\mathbf{E}[\delta^j - \delta^{j+1}] \leq \epsilon j + 4\rho$,
\[
\mathbf{E}\left[\sum_{j=0}^{k(n)-1} \epsilon^j\right] \geq \frac{\delta(n)}{4}
\]

Proof Sketch: Using almost the same argument in Lemma 2.4, we could prove that $\delta^{k(n)}$ is close to 0 with very high probability, so that $\mathbf{E}[\delta^{k(n)}] \leq \delta(n)/2$. Hence
\[
\frac{\delta(n)}{2} \leq \delta(n) - \mathbf{E}[\delta^{k(n)}] = \sum_{j=0}^{k(n)-1} (\mathbf{E}[\delta^j - \delta^{j+1}]) \leq \sum_{j=0}^{k(n)-1} \mathbf{E}[\epsilon^j] + 4k(n)\rho = \sum_{j=0}^{k(n)-1} [\epsilon^j] + \frac{\delta(n)}{4}
\]

Now we show that $\mathbf{E}[\delta^j - \delta^{j+1}] \leq \epsilon j + 4\rho$. By the definition of $\delta^j$, $W_{j+1} = (X_{j+1}, I_{j+1})$ are chosen uniformly and with probability $\lambda_n$, $W_{j+1}$ belongs to $T$, therefore
\[
\delta^j = \Pr[A(D^j) = 1] - \Pr[A(E^j) = 1] = \lambda_n \mathbf{E}[\delta_0^j(W)] + (1 - \lambda_n)(\mathbf{E}[\delta_0^j(W)]) = \lambda_n (\mathbf{E}[\delta_0^j(W)] - \mathbf{E}[\delta_1^j(W)]) + \lambda_n \mathbf{E}[\delta_1^j(W)] + (1 - \lambda_n)\mathbf{E}[\delta_0^j(W)] \leq \epsilon j + \lambda_n \mathbf{E}[\delta_1^j(W)] + (1 - \lambda_n)\mathbf{E}[\delta_0^j(W)]
\]

where $W \in_R T$ and $\overline{W} \in_R \overline{T}$. The remaining thing is to prove
\[
\mathbf{E}[\delta^{j+1}] \geq \lambda_n \mathbf{E}[\delta_1^j(W)] + (1 - \lambda_n)\mathbf{E}[\delta_0^j(\overline{W})] - 4\rho
\] (4)

Looking back to our construction of $D^{j+1}$ and $E^{j+1}$, we choose $t$ points from $U$. By our setting of parameters, with probability $1 - 2^{-n}$, there are at least $n/\rho$ points belong $T$ and at least $n/\rho$ points belong $\overline{T}$. Then by the Chernoff bound, we have
\[
\Pr[\max_{1 \leq s \leq t} \delta_c^j(w_s) \geq \max\{\mathbf{E}_W[\delta_c^j(W)], \mathbf{E}_{\overline{W}}[\delta_c^j(\overline{W})]\} - \rho] \geq 1 - 2^{-n}
\]

where $c$ is a value chosen from $\{0, 1\}$. Altogether with probability at least $1 - 3 \cdot 2^{-n}$,
\[
\delta_c^j(w_{s_0}) \leq \Delta_c^j(w_{s_0}) - \rho = \max_{1 \leq s \leq t} \Delta_c^{j+1}(w_s) - \rho \geq \max_{1 \leq s \leq t} \delta_c^{j+1}(w_s) - 2\rho \geq \max\{\mathbf{E}_W[\delta_c^j(W)], \mathbf{E}_{\overline{W}}[\delta_c^j(\overline{W})]\} - 3\rho
\]

Hence $\mathbf{E}[\delta_c^j(w_{s_0})] \geq \max\{\mathbf{E}_W[\delta_c^j(W)], \mathbf{E}_{\overline{W}}[\delta_c^j(\overline{W})]\} - 4\rho$. therefore,
\[
\mathbf{E}[\delta^{j+1}] = \lambda_n \mathbf{E}[\delta_1^j(w_{s_0})] + (1 - \lambda_n)\mathbf{E}[\delta_0^j(w_{s_0})] \geq \lambda_n \mathbf{E}_W[\delta_1^j(W)] + (1 - \lambda_n)\mathbf{E}_{\overline{W}}[\delta_0^j(\overline{W})] - 4\rho
\]

which conclude the claim. As a final remark, note that the proof still holds when we use an approximation $\tilde{\lambda}_n$ where $\lambda_n - 1/n \leq \tilde{\lambda}_n \leq \lambda_n + 1/n$