Randomness in Cryptography March 25, 2013

Lecture 11: Key Derivation without entropy loss

Lecturer: Yevgeniy Dodis *Scribe: Abhishek Samanta*

In reality, perfect source of randomness is hard to find. So, for real life applications, an imperfect source X of min-entropy k is converted into usable m-bit cryptographic key for some underlying application P. If P has security δ (against some class of attackes) with uniform random m-bit key, our goal is to design a key derivation function (KDF) h that allows us to use $R = h(x)$ as the key for P and results in comparable $\delta' \approx \delta$. This lower bound is known to be tight in general. In todays class we explore new areas to design KDFs with less waste for important special classes of sources of X and applications P.

1 Last Class

Before delving into technical details, let us refresh our memory with some important definitions and few important results we proved in last lecture.
Definition 1 ($\mathbf{H}_2$ Condenser) We say that an effcient function Cond : $\{0,1\}^n \times \{0,1\}^v \to \{0,1\}^m$ is a $(\frac{k}{n} \to \frac{m-d}{m})_2$-condenser if for $\mathbf{H}_2(X) \geq k$ and uniformly random S we have $\mathbf{H}_2(Cond(X;S)|S) \geq m - d$.

Theorem 1 *If an application P is (T, ε)-secure and (T, σ)-square secure (in the ideal model) and Cond is $(\frac{k}{n} \to \frac{m-d}{m})_2$-condenser, then using $R = Cond(X;S)$ as a key makes P (T, ε')-secure in the $(k,n)_2$-real model, where*

$$\varepsilon' \leq \varepsilon + \sqrt{\sigma \cdot (2^d - 1)}.$$

Lemma 1 *Universal hash function $h_s : \{0,1\}^n \to \{0,1\}^m$ is $(\frac{k}{n} \to \frac{m-d}{m})_2$-condenser where,*

$$2^d - 1 = 2^{m-k}.$$

Corollary 2 *If key derivation function (KDF) is universal hash function, then*

$$\varepsilon' \leq \varepsilon + \sqrt{\sigma \cdot 2^{m-k}}$$

Remark 1 *For square friendly applications, $\sigma \approx \varepsilon$, thus,*

$$\varepsilon' \approx \sqrt{\varepsilon \cdot 2^{m-k}}.$$

Remark 2 *For square friendly applications $\sigma \approx \varepsilon$. So, with entropy loss of $\log \frac{1}{\varepsilon}$ ($k = m + \log \frac{1}{\varepsilon}$), we get,*

$$\varepsilon' \approx 2 \cdot \varepsilon.$$

With no entropy loss ($k = m$), $\varepsilon' \approx \sqrt{\varepsilon}$.

2 Key derivation without entropy waste

2.1 Heuristic bound

In practice, one would typically use so called cryptographic hash function h, such as SHA or MD5, for key derivation. The reason behind this is the common belief that cryptographic hash functions achieve excellent security $\delta' \approx \delta$, when $k \approx m$. This can be easily justified in the random oracle model; assuming the KDF h is a random oracle which can be evaluated on at most q points (where, q is the upper bound of the attacker's running time), one can upper bound $\delta' \leq \delta + q/2^k$, where $q/2^k$ is the probability the attacker evaluates $h(X)$, where X is a source. In turn, in time q the attacker can also test about q out of 2^m possible m-bit keys, and hence achieve advantage $q/2^m$. This means that the ideal security δ of P cannot be lower than $q/2^m$ for most applications P. Thus, $q \leq \delta \cdot 2^m$. Plugging this bound on q in the bound of $\delta' \leq \delta + q/2^k$ above,we get that using a random oracle (RO) as a KDF achieves "real security",

$$\delta' \leq \delta_{RO} \overset{def}{=} \delta + \delta \cdot 2^{m-k} \tag{1}$$

In particular, $\delta' < 2\delta$ even when k = m. For example, to derive a 128-bit key for a CBC-MAC with security $\delta \approx \delta' \approx 2^{-64}$, one needs $k \approx 128$ bits of min-entropy.

Main questions Can one find reasonable application scenarios where one can design a provably-secure KDF achieving "real security" $\delta' \approx \delta$ when $k \approx m$ (matching the heuristic bound in Equation (1))? More generally, for a given (class of) applications P,
(A) What is the best (provably) achievable security δ' when $k = m$?
(B) What is the smallest (provable) entropy threshold k to achieve security $\delta' = O(\delta)$?

2.2 Using Leftover Hash Lemma (LHL)

In theory, the cleanest way to design a general KDF is by using famous Leftover Hash Lemma (LHL) [4], which achieves security $\varepsilon = \sqrt{2^{m-k}}$. This gives the following very general bound on δ' for all applications P,

$$\delta' \leq \delta_{ALL} \overset{def}{=} \delta + \sqrt{2^{m-k}} \tag{2}$$

As we can see, this provable (and very general) bound is much worse than the heuristic bound in Equation (1). In particular, we get no meaningful security when $k = m$ (giving no answer to Question (A)), and must assume $k \geq m + 2\log(1/\delta)$ to ensure that $\delta' = O(\delta)$ for Question (B). For example, to derive a 128-bit key for a CBC-MAC with security $\delta \approx \delta' \approx 2^{-64}$, one needs $k \approx 256$ bits of min-entropy.

2.3 Using square friendly(SF) applications

The idea here is that for SF applications one can argue that the derived key $R = h_s(X)$ is still "good enough" for P despite not being statistically close to U_m (given s). Intuitively, while any traditional application P demands that the expectation (over the uniform distribution $r \leftarrow U_m$) of the attacker's advantage $f(r)$ on key r is at most δ, square-friendly applications additionally require that the expected value of $f^2(r)$ is also bounded by δ. Additionally, for all such square-friendly applications P, it was shown that universal (and thus also the stronger pairwise independent) hash functions $\{h_s\}$ yield the following improved bound on the security δ' of the derived key $R = h_s(X)$,

$$\delta' \leq \delta_{SQF} \overset{def}{=} \delta + \sqrt{\delta \cdot 2^{m-k}} \tag{3}$$

This provable and still relatively general bound lies somewhere in between the idealized bound Equation (1) and the fully generic bound Equation (2): in particular, Equation (3) achieves security $\delta' \approx \delta$ when $k = m$ (giving partial answer to Question (A)), or, alternatively, we get full security $\delta' = O(\delta)$ provided $k \geq m + \log(1/\delta)$ (giving a partial answer to Question (B)). For example, to derive a 128-bit key for a CBC-MAC having ideal security $\delta = 2 - 64$, we can either settle for much lower security $\delta' \approx 2^{-32}$ with $k = 128$, or get full security $\delta' \approx 2^{-64}$ with $k = 192$. However, both bounds are still far from the expected bound $\delta' \approx 2^{-64}$ with $k = 128$, raising the question if further improvements are possible. But, unfortunately this bound is tight, for SF applications. Consider the following counter example,

2.3.1 Counter example P

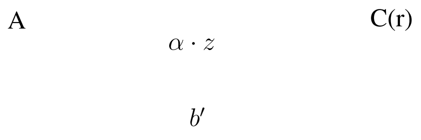

Figure 1: Counter example to show that Equation (3) is tight

Let us consider an SF application P as follows,

- The challenger has a random source r and is represented by $C(r)$ and the attacker is called A
- The challenger ($C(r)$) flips a coin $\alpha \in \{0, 1\}$, s.t. $Pr(\alpha = 1) = \sqrt{\delta}$
- Challenger flips $b \in \{0, 1\}$.
- The challenger generates z to send to the attacker. z can have two values as follows,
  - If $b = 0$, $z = r$.
  - Otherwise, $z = U$ (fresh uniformly random variable)
- The challenger sends $\alpha \cdot z$.
- The attacker in return sends back b'. The attacker wins iff $b = b'$.

Note that, for the above application ideal security is 0.

Claim 1 *If σ is the square security of the above mentioned application P, then $\sigma \leq \delta$.*

Proof: It is to be noted that, if $\alpha = 0$, challenger sends 0 to the attacker. So, if attacker receives 0, it best for the attacker to simply output a random guess $b' \leftarrow \{0,1\}$. If it receives some $r \in \{0,1\}^m$, then it outputs 1 if $Pr_X(h_s(X) = r) \geq 2^{-m}$ and 0, otherwise.So, the attacker can win the game iff $\alpha = 1$. Thus, for all r and A,

$$f(r) \leq \sqrt{\delta}, \text{ where } f(r) \text{ is advantage of } A$$
$$\Rightarrow f^2(r) \leq \delta$$

□

Note 1 *SRT bound [1] implies that using a universal hash function $\{h_s : \{0,1\}^n \to \{0,1\}^m\}$ as a key derivation function (KDF), there exists an efficiently samplable (polynomial in n) distribution X, and a (generally inefficient) distinguisher D, s.t. $\Delta_D((S, h_s(X)), (S, U)) \geq \sqrt{2^{k-m}}$*

Thus, for above mentioned P,

$$\delta' \leq \delta + \sqrt{\delta \cdot 2^{m-k}}$$

Definition 2 ($\mathbf{H}_\infty$-condenser) A function $Cond : \{0,1\}^n \times \{0,1\}^v \to \{0,1\}^m$ is $(\frac{k}{n} \xrightarrow{\varepsilon} \frac{m-d}{m})_\infty$-condenser ($(k, d, \varepsilon)$-condenser) if for all (n, k)-source X, and a uniformly random and independent seed $S \leftarrow \{0,1\}^v$, the joint distribution $(S, Cond(X, S)) \overset{\varepsilon}{\approx} (S, Y)$ such that $\mathbf{H}_\infty(Y|S) \geq m - d$, where Y is a random variable. ♢

Note 2 *$d = 0$, generalizes extractor.*

We can think of our (k, d, ε)-condenser as a way to hash 2^k items (out of a universe of size 2^n) into 2^m bins, so that the load (number of items per bin) is not too much larger than the expected 2^{k-m} for "most" of the bins. More concretely, it boils down to analyzing a version of average-load: if we choose a random item (and a random hash function from the family) then the probability that the item lands in a bin with more than $2^d \cdot 2^{k-m}$ items should be at most ε.

2.4 Using unpredictability applications

Definition 3 (Unpredictability extractor) We say that a function $D : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}$ is a δ-distinguisher if $Pr[D(U_m) = 1] \leq \delta$, where U_m is uniform random over $\{0,1\}^m$. A function $UExt : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is (k, δ, ε)-unpredictability extractor if for any (n, k)-source X and any δ-distinguisher D, we have $Pr[D(UExt(X; S), S) = 1] \leq \varepsilon$ where S is uniform over $\{0,1\}^d$. ♢

Definition 4 (Condenser) A function $Cond : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, l, ε)-condenser if for all (n, k)-sources X, and a uniformly random and independent seed S over $\{0,1\}^d$, the joint distribution $(S, Cond(X; S))$ is ε-statistically-close to some joint distribution (S, Y) such that, for all $S \in \{0,1\}^d$, $\mathbf{H}_\infty(Y|S = s) \geq m - l$. ♢

Lemma 3 *(Condenser $\Rightarrow$ UExt). Any (k, l, ε)-condenser is a $(k, \delta, \varepsilon*)$-UExt where $\varepsilon* = \varepsilon + 2^l \cdot \delta$*

Proof: Let $Cond : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$be a (k, l, ε)-condenser and let X be an (n, k)-source. Let S be uniform over $\{0,1\}^d$, so that, by definition, there is a joint distribution (S, Y) which has statistical distance at most ε from $(S, Cond(X; S))$ such that $\mathbf{H}_\infty(Y|S = s) \geq m - l$ for all $s \in \{0,1\}^d$. Therefore,for any δ-distinguisher D, we have,

$$\begin{aligned}
Pr[D(Cond(X;S),S) = 1] &\leq \varepsilon + Pr[D(Y,S) = 1] \\
&= \varepsilon + \sum_{y,s} Pr[S = s] \cdot Pr[Y = y|S = s] \cdot Pr[D(y,s) = 1] \\
&\leq \varepsilon + \sum_{y,s} 2^{-d} 2^{\mathbf{H}_\infty(Y|S=s)} \cdot Pr[D(y,s) = 1] \\
&\leq \varepsilon + 2^l \cdot \sum_{y,s} 2^{-(m+d)} \cdot Pr[D(y,s) = 1] \\
&\leq \varepsilon + 2^l \cdot \delta
\end{aligned}$$

□

Definition 5 (Balanced Hashing). Let $h = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^d}$ be a hash function family. For $\mathcal{X} \subseteq \{0,1\}^n$, $s \in \{0,1\}^d$, $x \in \mathcal{X}$ we define $Load_{\mathcal{X}}(x, s) = |\{x' \in \mathcal{X} : h_s(x') = h_s(x)\}|$. We say that the family h is (k, t, ε)-balanced if for all $\mathcal{X} \subseteq \{0,1\}^n$ of size $|\mathcal{X}| = 2^k$, we have,

$$Pr[Load_{\mathcal{X}}(X, s) > t \cdot 2^{k-m}] \leq \varepsilon,$$

where S, X are uniformly random and independent over $\{0,1\}^d$, $\mathcal{X}$, respectively. ◇

Lemma 4 *(Balanced ⇒ Condenser). Let $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^d}$ be a (k, t, ε)-balanced hash function family. Then the function $Cond : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ defined by $Cond(x; s) = h_s(x)$ is a (k, l, ε)-condenser for $l = \log(t)$.*

Proof: Without loss of generality, we can restrict ourselves to showing that Cond satisfies the condenser definition for every flat source X which is uniformly random over some subset $\mathcal{X} \subseteq \{0,1\}^n$, $|\mathcal{X}| = 2^k$. Let us take such a source X over the set $\mathcal{X}$, and define a modified hash family $\tilde{\mathrm{h}} = \{\tilde{\mathrm{h}}_s : \mathcal{X} \to \{0,1\}^m\}_{s\in\{0,1\}^d}$, which depends on $\mathcal{X}$ and essentially "rebalances" h on the set $\mathcal{X}$. In particular, for every pair (s, x) such that $Load^h_{\mathcal{X}}(x, s) \leq t \cdot 2^{k-m}$, we set $\tilde{\mathrm{h}}_s(x) = h_s(x)$, and for all other pairs (s, x) we define $\tilde{\mathrm{h}}_s(x)$ in such a way that $Load^{\tilde{\mathrm{h}}}_{\mathcal{X}} \leq t \cdot 2^{k-m}$(the super-script is used to denote the hash function with respect to which we are computing the load). It is easy to see that this "re-balancing" is always possible. We use the re-balanced hash function $\tilde{\mathrm{h}}$ to define a joint distribution (S, Y) by choosing S uniformly at random over $\{0,1\}^d$, choosing X uniformly/independently over $\mathcal{X}$ and setting $Y =\tilde{\mathrm{h}}_S(X)$. It is easy to check that the statistical distance between $(S, Cond(X; S))$ and (S, Y) is at most $Pr[h_S(X) \neq\tilde{\mathrm{h}}_S(X)] \leq Pr[Load^h_{\mathcal{X}}(X, S) > t \cdot 2^{k-m}] \leq \varepsilon$. Furthermore, for every $s \in \{0,1\}^d$, we have,

$$\begin{aligned}
\mathbf{H}_\infty(Y|S=s) &= -\log(\max_y Pr[Y=y|S=s]) \\
&= -\log(\max_y Pr[X \in \tilde{\mathrm{h}}_s^{-1}(y)]) \\
&\geq -\log(t \cdot 2^{k-m}/2^k) \\
&= m - \log t
\end{aligned}$$

Thus, $Cond$ is a $(k, l = \log t, \varepsilon)$-condenser. □

Lemma 5 *(UExt ⇒ balanced). Let* $UExt : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k, δ, ε)*-UExt for some,* $\varepsilon > \delta > 0$*. Then the hash family* $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^d}$ *defined by* $h_s(x) = UExt(x; s)$ *is* $(k, \varepsilon/\delta, \varepsilon)$*-balanced.*

Proof: Let, $t = \varepsilon/\delta$ and assume that $\mathcal{H}$ is not (k, t, ε)-balanced. Then there exists some set $\mathcal{X} \subseteq \{0,1\}^n$, $|\mathcal{X}| = 2^k$, s.t. $\varepsilon' = Pr[Load_{\mathcal{X}}(X, S) > t \cdot 2^{k-m}] > \varepsilon$, where X is uniform over $\mathcal{X}$ and S is uniform over $\{0,1\}^d$. Let $\mathcal{X}_s \subseteq \mathcal{X}$ be defined by $\mathcal{X}_s = \{x \in \mathcal{X} : Load_{\mathcal{X}}(X, S) > t \cdot 2^{k-m}\}$ and let $\varepsilon_s \overset{def}{=} |\mathcal{X}|/2^k$. By definition $\varepsilon' = \sum_s 2^{-d}\varepsilon_s$. Define $\mathcal{Y}_s \subseteq \{0,1\}^m$ via $\mathcal{Y}_s = h_s(\mathcal{X}_s)$. Now by definition, each $y \in \mathcal{Y}_s$ has atleast $t \cdot 2^{k-m}$ pre-images in $\mathcal{X}_s$, and therefore $\delta_s \overset{def}{=} |\mathcal{Y}_s|/2^m \leq |\mathcal{X}_f|/(t \cdot 2^{k-m} \cdot 2^m) \leq \varepsilon_s/t$ and $\delta = \sum_s 2^{-d} \cdot \delta_s \leq \varepsilon'/t$.

Define the distinguisher D via $D(y, s) = 1$ iff $y \in \mathcal{Y}_s$. The D is a δ-distinguisher for $\delta \leq \varepsilon'/t \leq \varepsilon/t$, but $Pr[D(h_S(X), S) = 1] = \varepsilon' \geq \varepsilon$. Thus, $UExt$ is not a $(k, \varepsilon/t, \varepsilon)$-UExt. □

Summary Taking Lemma 3, Lemma 4, and Lemma 5, together, we see that they are close to tight. In particular, for any $\varepsilon > \delta > 0$, we get,

$$\begin{aligned}
(k, \delta, \varepsilon) - UExt &\Rightarrow (k, \varepsilon/\delta, \varepsilon) - balanced \text{ [Using Lemma 5]} \\
&\Rightarrow (k, \log(\varepsilon/\delta), \varepsilon) - Cond \text{ [Using Lemma 4]} \\
&\Rightarrow (k, \delta, 2 \cdot \varepsilon) - UExt \text{ [Using Lemma 3]}
\end{aligned}$$

2.4.1 Constructing Unpredictability Extractors

Theorem 2 *There exists an efficient* (k, δ, ε)*-unpredictablity extractor* $UExt : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for the following parameters,*

- *When* $k = m$ *(no entropy loss), we get* $\varepsilon = (1 + \log(1/\delta)) \cdot \delta$
- *When* $k \geq m + \log\log(1/\delta) + 4$*, we get* $\varepsilon = 3 \cdot \delta$
- *In general,* $\varepsilon = O(1 + 2^{m-k} \cdot \log(1/\delta)) \cdot \delta$

In all cases, the function UExt is simply a $(\log(1/\delta) + O(1))$*-wise independent hash function and the seed length is* $d = O(n \log(1/\delta))$

We prove Theorem 2 by constructing "good" balanced hash functions and using our connections between balanced hashing and unpredictability extractors.

Lemma 6 *Let $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^k\}$ be $(t+1)$-wise independent. Then it is (k,t,ε)-balanced where $\varepsilon \le (\frac{e}{t})^t$ and e is the base of the natural logarithm.*

Proof: Fix any set $\mathcal{X} \subseteq \{0,1\}^n$ of size $|\mathcal{X}| = 2^k$. Let X be uniform over $\mathcal{X}$ and S be uniform/independent over $\{0,1\}^d$. Then

$$\begin{aligned}
Pr[Load_{\mathcal{X}}(X,S) > t] &\le Pr[\exists \mathcal{C} \subseteq \mathcal{X}, |\mathcal{C}| = t, \forall x' \in \mathcal{C} : h_S(x') = h_S(X) \wedge x' \neq X] \\
&\le \sum_{\mathcal{X} \subseteq \mathcal{X}, |\mathcal{C}|=t} Pr[\forall x' \in \mathcal{C} : h_S(x') = h_S(X) \wedge x' \neq X] \\
&\le \binom{2^k}{t} 2^{-tk} \\
&\le \left(\frac{e \cdot 2^k}{t}\right)^t \cdot 2^{-tk} \\
&\le \left(\frac{e}{t}\right)^t
\end{aligned}$$

□

Corollary 7 *For any $0 < \varepsilon < 2^{-2e}$, any $\delta > 0$, a $(\log(1/\varepsilon)+1)$-wise independent hash family $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^k\}_{s \in \{0,1\}^d}$ is, $(k+\log(1/\varepsilon), \varepsilon)$-balanced, $(k, \log\log(1/\varepsilon), \varepsilon)$-condenser, $(k, \delta, \log(1/\varepsilon)\cdot\delta + \varepsilon)$-UExt. Setting $\delta = \varepsilon$, we get a $(k, \delta, (1+\log(1/\delta))\cdot\delta)$-UExt*

Proof: Set $t = \log(1/\varepsilon)$ in Lemma 6 and notice that $(\frac{e}{t})^t \le 2^{-t} \le \varepsilon$ as long as $t \ge 2e$. □

This establishes part(1) of Theorem 2. Next we look at a more general case where k may be larger than m. This also covers the case $k = m$ but gets a somewhat weaker bound. It also requires a more complex tail bound for q-wise independent variables.

Lemma 8 *Let $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in S}$ be $(q+1)$-wise independent. Then, for any $\alpha > 0$, it is $(k, 1+\alpha, \varepsilon)$-balanced where $\varepsilon \le 8 \cdot \left(\frac{q \cdot 2^{k-m} + q^2}{(\alpha \cdot 2^{k-m} - 1)^2}\right)^{q/2}$.*

Proof: Let $\mathcal{X} \subseteq \{0,1\}^n$ be a set of size $|\mathcal{X}| = 2^k$, X be uniform over $\mathcal{X}$, and S be uniform/independent over $\{0,1\}^d$. Define the indicator random variables $C(x^*, x)$ to be 1 if $h_S(x) = h_S(x^*)$ and 0, otherwise. Then,

$$\begin{aligned}
Pr[Load_{\mathcal{X}}(X,S) > (1+\alpha)\cdot 2^{k-m}] &= \sum_{x^* \in \mathcal{X}} Pr[X = x^*] \cdot Pr[Load_{\mathcal{X}}(x^*, S) > (1+\alpha)\cdot 2^{k-m}] \\
&= 2^{-k} \cdot \sum_{x^* \in \mathcal{X}} Pr\left[\sum_{x \in \mathcal{X} \setminus \{x^*\}} C(x^*, x) + 1 > (1+\alpha)\cdot 2^{k-m}\right] \\
&\le 8 \cdot \left(\frac{q \cdot 2^{k-m} + q^2}{(\alpha \cdot 2^{k-m} - 1)^2}\right)^{q/2}
\end{aligned}$$

Where the last line follows from the tail inequality [3] with random variables $\{C(x^*, x)\}_{x \in \mathcal{X} \setminus \{x^*\}}$ which are q-wise independent and have expected value $\mu = \mathbb{E}[\sum_{x \in \mathcal{X} \setminus \{x*\}} C(x^*, x)] = (2^k - 1) \cdot 2^{-m} \le 2^{k-m}$, and by setting $A = (1+\alpha)\cdot 2^{k-m} - 1 - \mu \ge \alpha \cdot 2^{k-m} - 1$; recall that $C(x^*, x^*)$ is always 1 and $C(x^*, x)$ for $x \neq x^*$ is 1 with probability 2^{-m}. □

Corollary 9 *For any* $0 < \varepsilon < 2^{-7}$*,* $k \geq m + \log\log(1/\varepsilon) + 4$*, a* $(\log(1/\varepsilon) + 4)$*-wise independent hash function family* $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^d}$ *is,* $(k, 2, \varepsilon)$*-balanced,* $(k, 1, \varepsilon)$*-condenser,* $(k, \delta, 2\delta + \varepsilon)$*-UExt for any* $\delta > 0$*. Setting* $\delta = \varepsilon$*, it is a* $(k, \delta, 3\delta)$*-UExt.*

Proof: Set $q = \log(1/\varepsilon) + 3$, $\alpha = 1$ and $2^{k-m} = 5q$. Then we apply Lemma 8,

$$8 \cdot \left(\frac{q \cdot 2^{k-m} + q^2}{(\alpha \cdot 2^{k-m} - 1)^2}\right)^{q/2} \leq 8 \cdot \left(\frac{6 \cdot q^2}{(5q-1)^2}\right)^{q/2} \leq 8\left(\frac{1}{4}\right)^{q/2} \leq 8(2^{-q}) \leq \varepsilon.$$

The second step assumes $q > 10$ meaning that $\varepsilon < 2^{-7}$. □

The above corollary establishes part (2) of Theorem 2. The next corollary gives us a general bound which establishes part (3) of the theorem. Asymptotically it implies both Corollary 7 and Corollary 9 but with worse constants.

Corollary 10 *For any* $\varepsilon > 0$ *and* $q = \log(1/\varepsilon) + 3$*, a* $(q+1)$*-wise independent hash function family* $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^d}$ *is* $(k, 1 + \alpha, \varepsilon)$*-balanced for*

$$\alpha = 4 \cdot \sqrt{q \cdot 2^{m-k} + (q \cdot 2^{m-k})^2} = O(2^{m-k} \cdot \log(1/\varepsilon) + 1).$$

By setting $\delta = \varepsilon$*, a* $(\log(1/\delta) + 4)$*-wise independent hash function is a* $(k, \delta, O(1 + 2^{m-k} \cdot \log 1/\delta) \cdot \delta)$*-UExt.*

Proof: The first part follows from Lemma 8 by noting that,

$$8 \cdot \left(\frac{q \cdot 2^{k-m} + q^2}{(\alpha \cdot 2^{k-m} - 1)^2}\right)^{q/2} \leq 8 \cdot \left(\frac{6 \cdot q^2}{(5q-1)^2}\right)^{q/2} \leq 8\left(\frac{1}{4}\right)^{q/2} \leq \varepsilon.$$

For the 2^{nd} part, we can consider two cases. If $q \cdot 2^{m-k} \leq 1$, then $\alpha \leq 4\sqrt{2}$ and we are done. Else, $\alpha \leq 4\sqrt{2} \cdot (q \cdot 2^{m-k}) = 4\sqrt{2}(\log(1/\varepsilon) + 3) \cdot 2^{m-k}$. □

References

[1] Dana Dachman-Soled, Rosario Gennaro, Hugo Krawczyk, and Tal Malkin. *Computational Extractors and Pseudorandomness* in TCC 2012

[2] Yevgeniy Dodis, Krzysztof Pietrzak, Daniel Wichs. *Key derivation without entropy waste*

[3] M. Bellare and J. Rompel. *Randomness-effcient oblivious sampling* In 35th Annual Symposium on Foundations of Computer Science, pages 276-287. IEEE, 1994.

[4] J. Hastad, R. Impagliazzo, L.A. Levin, and M. Luby. *Construction of pseudorandom generator from any one-way function.* In SIAM Journal on Computing, 28(4):1364-1396, 1999.