Randomness Extraction

We're going to construct a seeded extractor

<u>DEF</u> A function $\mathrm{Ext}: \underbrace{\{0,1\}^d}_{\text{SEED}} \times \underbrace{\{0,1\}^n}_{\text{SOURCE}} \to \{0,1\}^\ell$

is a (k,ε)-extractor if $\forall X \to H_\infty(X) \geq k$

$$SD((S, \mathrm{Ext}(S,X)); (S, U_\ell)) \leq \varepsilon$$

S is uniform but shorter than what we want

<u>THM</u> (Leftover hash lemma)

Let $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^\ell\}_{s \in \{0,1\}^d}$

one extractor for each seed

be pairwise independent.

Then $\mathrm{Ext}(s,k) = h_s(x)$ is (k,ε)-extractor

for $\underbrace{k \geq \ell + \log(2/\varepsilon) - 2}$

we want to get $\ell \approx k$ as much as possible

<u>PROOF</u>: We rely on a few lemmas.

<u>LEMMA</u>: Let Y be a random variable over $\mathcal{Y}$ such that

collision probability: the probability that extracting from two independent sources you get the same result

$$\mathrm{Col}(\mathcal{Y}) = \Pr[Y = Y']$$

$$= \sum_y \Pr[Y = y]^2$$

$$\leq \frac{1}{|\mathcal{Y}|}(1+4\varepsilon^2)$$ → if the collision is bounded by this quantity,

Then, $SD(Y, U) \leq \varepsilon$

the random variable is almost uniform

<u>EX</u> Let X be such that $H_\infty(X) \geq k$
Then $Col(X) \leq 2^{-k}$

<u>EX</u> Let $\mathcal{H}$ be a pairwise independent family as in THM.

$$Pr[h_s(X) = h_s(X') \wedge X \neq X] \leq 2^{-\ell}$$

Now let's prove LHL.

Idea: Use technical lemme with

$$Y = (S, h_s(X))$$
$$\mathcal{Y} = \{0,1\}^{d+\ell}$$
$$Y' = (S', h_{s'}(X'))$$

$$Col(Y) = Pr[Y = Y']$$
$$= Pr[S = S' \wedge h_s(X) = h_{s'}(X')]$$
$$= Pr[S = S'] \cdot Pr[h_s(X) = h_s(X')]$$
$$= Pr[S = S' \wedge \underbrace{h_s(X) = h_s(X')}_{\text{because } S = S'}]$$
$$= \underbrace{Pr[S = S']}_{} \cdot Pr[h_s(X) = h_s(X')]$$

$$\overset{\text{by Ex 1}}{=} 2^{-d} \cdot \Pr[h_S(X) = h_S(X')]$$

$$\leq 2^{-d}\left(\Pr[X = X'] + \Pr[h_S(X) = h_S(X') \wedge X \neq X']\right)$$

$$\leq 2^{-d}\left(2^{-k} + 2^{-\ell}\right)$$

$$\leq \frac{1}{2^{d+\ell}}\left(2^{\ell-k} + 1\right)$$

$$\leq \frac{1}{2^{d+\ell}}\left(2^{2-2\log(1/\varepsilon)} + 1\right)$$

$$= \frac{1}{|\Upsilon|}\left(1 + 4\varepsilon^2\right)$$

$\Rightarrow$ By LEMMA $SD(Y, U_{d+\ell}) \leq \varepsilon$

<u>Proof of LEMMA</u>: By def of SD

$$SD(Y;U) = \frac{1}{2}\sum_y \left| \Pr[Y=y] - \underbrace{\frac{1}{|\Upsilon|}}_{\Pr[U=y]} \right|$$

Define $q_y = \Pr[Y=y] - \frac{1}{|\Upsilon|}$

$$s_y = \begin{cases} 1 & \text{if } q_y \geq 0 \\ -1 & \text{if } q_y < 0 \end{cases}$$

$$SD(Y;U) = \frac{1}{2}\sum_y q_y \cdot s_y$$

$$\xrightarrow{\quad} \vec{q} = (q_y)_{y \in Y}$$

$$= \frac{1}{2} \langle \vec{q}, \vec{s} \rangle$$

$$\leq \frac{1}{2} \sqrt{\langle \vec{q}, \vec{q} \rangle \cdot \langle \vec{s} \cdot \vec{s} \rangle} \quad \text{(by CS)}$$

$$= \frac{1}{2} \sqrt{\sum_y q_y^2 \cdot |\Upsilon|}$$

Look at

$$\sum_y q_y^2 = \sum_y \left(\Pr[Y=y]^2 + \frac{1}{|\Upsilon|^2} - 2 \frac{\Pr[Y=y]}{|\Upsilon|} \right)$$

$$= \mathrm{Col}(Y) + \frac{1}{|\Upsilon|} - \frac{2}{|\Upsilon|} =$$

$$= \mathrm{Col}(Y) - \frac{1}{|\Upsilon|}$$

$$\leq (1 + 4\varepsilon^2) \cdot \frac{1}{|\Upsilon|} - \frac{1}{|\Upsilon|}$$

$$= \frac{4\varepsilon^2}{|\Upsilon|}$$

$$\Rightarrow SD(Y; U) \leq \frac{1}{2} \sqrt{\underbrace{\sum_y q_y^2}_{} \cdot |\Upsilon|}$$

$$\leq \frac{1}{2} \sqrt{\frac{4\varepsilon^2}{|\Upsilon|} |\Upsilon|} = \varepsilon.$$

Computational Security

Until now the adversery is all-powerful. Actually real life doesn't work like this, so we want to model a real-world attacker!

A real attacker would use a kind of turing machine - so we model the attacker as a Turing machine! ($\mathcal{A}$)

$\mathcal{A}$ is an efficient Turing machine, and as all machines it has a small probability of failure ($\mathcal{A}$ breaks encryption w.p. $\varepsilon \leq 2^{-80}$)

Take 1: CONCRETE SECURITY

No $\mathcal{A}$ Turing machine running for t steps can break Π w.p. better than $\varepsilon = 2^{-80}$

Take 2: ASYMPTOTIC SECURITY

Parametrize everything with SECURITY PARAMETERS $\lambda \in \mathbb{N}$

probabilistic polynomial-time

$\Rightarrow$ EFFICIENT : $\mathcal{A}$ is a PPT Turing machine with input 1^λ

DEF Turing machine $\mathcal{A}$ is PPT if its worst case running time is polynomial

$\Rightarrow \exists\, p(\lambda) = poly(\lambda)$ s.t.

$\forall x \in \{0,1\}^*, r \in \{0,1\}^*$ (r → random)

then $\mathcal{A}(1^\lambda, x, r)$ runs in time $p(\lambda)$

POLYNOMIAL: $f: \mathbb{N} \to \mathbb{N}$ is polynomial

$f(\lambda) = poly(\lambda)$ if $\exists\, c \in \mathbb{N}$ s.t.

$f(\lambda) = O(\lambda^c)$

TIND: Negligible in $\lambda \in \mathbb{N}$.