CS6222 Cryptography

Topic: Min-Entropy and Leftover Hash Lemma | Date: October 17, 2024
Lecturer: Wei-Kai Lin (TA: Arup Sarker) | Scriber: Chase Fickes and Adarsha Poudel

---

1 Recap

We will start by briefly recalling an important definition from last class:

Definition 1 (Pairwise Independent Hash Family)**.** A family of hash functions $\mathcal{H} := \{h : \{0,1\}^n \to \{0,1\}^n\}$ is pairwise independent if, $\forall x$, $h(x)$ is chosen uniformly random such that, $\forall a \in \{0,1\}^n$, $\Pr_{h \leftarrow \mathcal{H}}[h(x) = a] = \frac{1}{2^n}$, and $\forall x_1 \neq x_2$, $(h(x_1), h(x_2)$ are independent variables.

A good example of this that we will refer to later is for the family of hash functions $\mathcal{H} := \{h_M := M \odot x\}, M \in \mathbb{Z}_2^{m \times n}$.

2 Motivation

Consider a OWF that is not a permutation such that, for example, each output has two input. This means that there are some values in the range that are not mapped to, which wastes some randomness of the OWF. Typically, if f is a OWF, then we can create a PRG g such that $g(x, r) = f(x)||r||x \odot r$, where $x \odot r$ was our hardcore bit.

To recover the randomness, we can construct a PRG g differently such that $g(x, M_1, M_2, r) = M_1||M_2||r||M_2 \odot f(x)||M_1 \odot x||x \odot r$. We can see instances of the example hash function we mentioned earlier that required matrix multiplication. Since these are pairwise independent functions, any two x are unlikely to map to the same output for $M_1 \odot x$, which allows us to recover the remaining randomness.

This is helpful, but it is currently not clear how many bits we need to include from the two hashes $M_1 \odot x$ and $M_2 \odot f(x)$ to fully recover the randomness, which is where the idea of entropy can help us.

3 Entropy

Entropy is a measure of how random or unpredictable our data is, and we define it in the following ways:

Definition 2 (Min-Entropy)**.** Suppose X is a random variable. We define the min-entropy of X to be the following:

$$H_\infty(X) := \min_x \left\{ \log \left(\frac{1}{\Pr[X = x]} \right) \right\}.$$

Definition 3 (Shannon Entropy)**.** Suppose X is a random variable. We define the min-entropy of X to be the following:

$$H(X) := \sum_a p_X(a)(-\log(p_X(a)) = \mathbb{E}_x[\log\left(\frac{1}{\Pr[X=x]}\right).$$

We note an example of applying the definition of entropy:

For $X = U_3||000$, we have that $H(X) = (\frac{1}{8} * -\log(\frac{1}{8})) * 8 = 3$. We can think of entropy as the number of random bits in the variable X.

Theorem 4 (Leftover Hash Lemma)**.** *Suppose $\mathcal{H} := \{h : \{0,1\}^n \to \{0,1\}^n\}$ is a pairwise independent has family and that there is a random variable X such that $H_\infty(X) \geq k \in \mathbb{N}$. Then, over the random space of X, for $h \leftarrow \mathcal{H}$, $\Delta[(h, h(x)), (h, U_m)] \leq \epsilon$ if $m = k - 2\log(\frac{1}{\epsilon})$ for all $\epsilon > 0$.*

We can apply the above definition in an example as follows: suppose X is a random variable and h is a hash from the leftover hash lemma such that $m = k - 2\log(\frac{1}{\epsilon})$. What is $\Pr_{h,y=U_m}[(h, y) \in \mathbf{Supp}(h, h(x))]$ equal to, where $\mathbf{Supp}(Z) := \{a|\Pr[Z = a] > 0\}$? The answer here is $1-\epsilon$, since, from the leftover hash lemma, we are given that (h, y) and (h, U_n) are statistically close.

Definition 5 (Weak Pseudo-Entropy Generator (PEG))**.** A function $F : \{0,1\}^n \to \{0,1\}^m$ is a weak PEG if, for $X \leftarrow U_n$, $\{F(X)\} \approx \{Y_n\}$ such that $H(Y_n) \geq k + \frac{1}{100n}$, and $H(F(X)) \leq k$.