CS6222 Cryptography

Topic: Generalized Goldreich-Levin lemma, pairwise independent hash — Date: Oct 10, 2024

Lecturer: Wei-Kai Lin (TA: Arup Sarker) — Scriber: Shiyu Li, Han Yang

1 Exercise 1

Suppose $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function (OWF), define $f' : \{0,1\}^{2n} \to \{0,1\}^{2n}, h : \{0,1\}^{2n} \to \{0,1\}$ to be the following:

$$f'(x,r) := f(x)||r, h(x,r) := x \odot r)$$

Is $g(x,r) := f'(x,r)||h(x,r)$ a OWF?

Yes. If f is a one-way permutation (OWP), g is a PRG and therefore a OWF. To really see g is a OWF when f is a OWF, we use a reduction that is similar to the one in the proof of Goldreich-Levin Lemma.

Proof. Suppose there exists NUPPT A and polynomial p s.t. for inf. many $n \in \mathbb{N}$

$$\Pr_{x,r}[y := g(x,r), g(A(1^{2n}, y)) = y] \geq \frac{1}{2} + \frac{1}{p(n)}.$$

To invert $y := g(x,r)$, the construction of $B(1^n, y)$ is as follows:

1. Choose pairwise independent strings $r_1, ..., r_m \in \{0,1\}^n$ and pairwise independent bits $b_1, ..., b_m \in \{0,1\}$.
2. For each $i = 1, ..., n$, do the following
   (a) For $j = 1, ..., m$
      i. $(x'_j, r'_j) \leftarrow A(y||r_j||b_j)$
   (b) Let x'_i be the majority of $\{x'_j\}_{j\in[m]}$
3. Output $x' = x'_1 x'_2 ... x'_n$

□

2 Generalized Goldreich-Levin Lemma

In the reduction, I am not really using this y except for keeping this y directly to A. That is a key property in the reduction that makes us being able to generalize this reduction to other kind of functions or capabilities. There is a generalized version of Goldreich-Levin Lemma. It is totally about this reduction.

Lemma 1 (Generalized Goldreich-Levin Lemma)**.** *Suppose there is an algorithm A and $\alpha > 0$ s.t.*

$$\Pr_{x,r}[A_x(r) = x \odot r] = \frac{1}{2} + \alpha,$$

$$GD_a(x) = \begin{cases} 1 & x = a \\ 0 & o.w. \end{cases}$$

Then

$$\Pr_{x,\ randomness\ of\ R}[GD_x\left(R^{A_x(\cdot)}(1^n)\right)=1]\geq\frac{1}{poly(n,1/\alpha)},$$

where R runs in time $poly(n,\frac{1}{\alpha})$.

This reduction is roughly a learning algorithm. Here the reduction begins with no information, the only thing it can do is just calling an algorithm A with some knowledge in an oracle way. And you should view this oracle as a black box you do not really know. But you can use random queries to extract the information out. Learning algorithms are extracting information out of some phenomena. It is a specialized form of learning. In learning algorithms, we also care about the number of sample we use, the number of query we ask, and the amount of randomness we use.

Here the reduction tries to minimize the number of randomness we are using, the number of guess we are making. I think this kind of algorithm is inspired by early stage of learning, you need to make some guess to make efficient learning. The better guess will probably not be uniformly random, but correlated but pairwise independent.

3 Exercise 2

Suppose $f:\{0,1\}^n\to\{0,1\}^n$ is a OWF, define $f':\{0,1\}^{3n}\to\{0,1\}^{3n}, h':\{0,1\}^{3n}\to\{0,1\}, f'':\{0,1\}^{3n}\to\{0,1\}^{3n+1}$ as follows

$$\begin{aligned}f'(x,r_1,r_2)&:=f(x)||r_1||r_2,\\ h'(x,r_1,r_2)&:=x\odot r_2,\\ f''(x,r_1,r_2)&:=f(x)||r_1||r_2||x\odot r_1.\end{aligned}$$

Clearly f' is a OWF, h' is a hard-core predicate (HCP) w.r.t. f', f'' is a OWF. Is h' a HCP w.r.t. f''?

Yes. h' a HCP w.r.t. f''.

This can be further extended:

$$\begin{aligned}f'''(x,r_1,r_2,r_3)&:=f(x)||r_1||r_2||r_3||x\odot r_1||x\odot r_2\\ h''(x,r_1,r_2,r_3)&:=x\odot r_3.\end{aligned}$$

h'' is a HCP w.r.t. f''.

This hard-core bits will continue for $O(\log n)$ times. That means, suppose f is a OWF, define

$$\begin{aligned}f^{(k)}(x,r_1,...,r_k)&:=f(x)||r_1||...||r_k||x\odot r_1||...||x\odot r_k,\\ h^{(k)}(x,r_1,...,r_k)&:=x\odot r_k.\end{aligned}$$

$h^{(k)}$ is a HCP w.r.t. $f^{(k)}$ as long as $k\leq O(\log n)$.

High-level idea: An adversary can't guess $\log n$ random bits in polynomial time.

4 Pairwise Independent Hash Functions

Definition 2 (Pairwise Independent Hash)**.** A family of functions $H = \{h : \{0,1\}^n \to \{0,1\}^m\}$ is pairwise independent if the following two conditions hold when $h \leftarrow H$ is a function chosen uniformly at random from H:

1. For all $x \in \{0,1\}^n$, the random variable $h(x)$ is uniform in $\{0,1\}^m$.
2. For all $x_1 \neq x_2 \in \{0,1\}^n$, the random variables $h(x_1)$ and $h(x_2)$ are independent.

Example: Pairwise Independent Hash from Linear Mapping. For any finite field F, define H to be the following set:

$$H := \{h_{a,b} : h_{a,b}(x) = ax + b, a, b \in F\}.$$

H is a pairwise independent hash family.

Example: Pairwise Independent Hash from Matrix.

$$H := \{h_M(x) := M \odot x\}, M \in Z_2^{m \times n}, x \in \{0,1\}^n.$$

H is a pairwise independent hash family.