Computational Security & Birthday Bounds

HW1 due

Big idea: Breaking security (distinguishing 2 libs) might be possible in principle, just very hard

(clarify: how hard?)

Last time:

$$\mathcal{L}_{\text{left}}: \boxed{\begin{array}{l} \underline{\text{haystack}(x \in \{0,1\}^\lambda):} \\ \quad \text{return false} \end{array}}$$

$$\mathcal{L}_{\text{right}}: \boxed{\begin{array}{l} \text{needle} \leftarrow \{0,1\}^\lambda \\ \underline{\text{haystack}(x \in \{0,1\}^\lambda):} \\ \quad \text{return } x \overset{?}{=} \text{needle} \end{array}}$$

Def: **Advantage** of adversary A is:

$$\Big| \Pr[A \diamond \mathcal{L}_{\text{left}} \Rightarrow 1] - \Pr[A \diamond \mathcal{L}_{\text{right}} \Rightarrow 1] \Big|$$

Ex: (haystack) If A makes 1 query, then advantage is $\le 1/2^\lambda$

If A makes 20 queries, then advantage is $\le 20/2^\lambda$

Q: how big of advantage should we care about??

Def: A function f is negligible if for every polynomial $p(x)$, $\lim_{\lambda \to \infty} p(\lambda) \cdot f(\lambda) \to 0$

Idea: Suppose A (poly-time) with some advantage f. If I repeat attack p times, resulting attack is still poly-time (if p is poly) and maybe has advantage $p \cdot f$, which should be negligible

Ex: $1/2^\lambda$ is negligible $\qquad \frac{\lambda^c}{2^\lambda} = \frac{2^{c \log \lambda}}{2^\lambda} = 2^{\overbrace{c\log\lambda - \lambda}^{\text{goes to } -\infty}}$

$1/\lambda^2$ is not negligible $\qquad \left(\frac{\lambda^{100}}{\lambda^2} \not\to 0 \right)$ a polynomial

Def: $\mathcal{L}_1 \approx \mathcal{L}_2$ (libs are <u>indistinguishable</u>)

if for all polynomial-time A,

the advantage of A is negligible

as a function of λ

Ex: the two "haystack" libraries are indistinguishable.

If A is poly-time, then # of queries it makes is polynomial in λ $\Rightarrow$ Advantage $\leq \frac{p(\lambda)}{2^\lambda}$ = negl.

Def: Write $f \approx g$ if $|f(\lambda) - g(\lambda)|$ is negligible

(libraries are indist. if $\forall A$: $\Pr[A \diamond \mathcal{L}_1 \Rightarrow 1] \approx \Pr[A \diamond \mathcal{L}_2 \Rightarrow 1]$)

Birthday Bounds

$\mathcal{L}_{\text{samp-L}}$

samp():
  $r \leftarrow \{0,1\}^\lambda$
  return r

samp w/ replacement

$\mathcal{L}_{\text{samp-R}}$

$R = \emptyset$
samp():
  $r \leftarrow \{0,1\}^\lambda \setminus R$
  add r to R
  return r

samp w/o replacement

Claim: $\mathcal{L}_{\text{samp-L}} \approx \mathcal{L}_{\text{samp-R}}$, more precisely,
Advantage of A is $\dfrac{q(q-1)}{2^{\lambda+1}}$ if A makes q queries

(If q is polynomial func of λ, advantage is negl.)

In Notes: advantage of A $\leq$ $\Pr[\mathcal{L}_{\text{samp-L}}$ repeats an output$]$

Here: $\leq \dfrac{q(q-1)}{2^{\lambda+1}}$

$\Pr[\text{repeated output}] = 1 - \Pr[\text{all outputs unique}]$

(call outputs of SAMP: $r_1, r_2, \ldots, r_q$)

$$\Pr[\text{all outputs unique}] = \Pr[r_2 \neq r_1] \cdot \Pr[r_3 \notin \{r_1, r_2\} \mid r_1 \neq r_2] \cdot \Pr[r_4 \notin \{r_1 .. r_3\} \mid \{r_1 .. r_3\} \text{ distinct}] \cdots$$

$$= \left(1 - \frac{1}{2^\lambda}\right)\left(1 - \frac{2}{2^\lambda}\right)\left(1 - \frac{3}{2^\lambda}\right) \cdots \left(1 - \frac{q-1}{2^\lambda}\right)$$

$$= \prod_{i=1}^{q-1}\left(1 - \frac{i}{2^\lambda}\right) \geq 1 - \sum_{i=1}^{q-1} \frac{i}{2^\lambda}$$

$$(1-x)(1-y) = 1 - x - y + xy$$

$$\geq 1 - x - y \quad \text{if } x, y > 0$$

$$= 1 - (x+y)$$

generally:

$$\prod(1-x_i) \geq 1 - \sum x_i \quad \text{if } x_i\text{'s} > 0$$

$$= 1 - \frac{1}{2^\lambda}\left[\frac{(q-1)q}{2}\right] = 1 - \frac{(q-1)q}{2^{\lambda+1}}$$