EECS 575: Advanced Cryptography
Fall 2021
Lecture 6

Paul Grubbs

paulgrub@umich.edu

Beyster 4709

Agenda for this lecture

- Announcements
- Recap from last time
- Pseudorandom Generators (PRGs)
- PRGs with polynomial stretch
- A concrete PRG from hardness of discrete logarithms
  - ”Blum-Micali”

Agenda for this lecture

- Announcements
- Recap from last time
- Pseudorandom Generators (PRGs)
- PRGs with polynomial stretch
- A concrete PRG from hardness of discrete logarithms
  - ”Blum-Micali”

Announcements

- Homework 2 is due 9/21 (tomorrow)
- Lecture topic vote
- Course feedback - email/Canvas/Piazza
  - Speed of course - too fast/slow

Agenda for this lecture

- Announcements
- Recap from last time
- Pseudorandom Generators (PRGs)
- PRGs with polynomial stretch
- A concrete PRG from hardness of discrete logarithms
  - ”Blum-Micali”

Recap from last time

Computational indistinguishability

- Define $Adv^{dist}_{X,Y} = |Pr[A(X)=1] - Pr[A(Y)=1]|$

  dists

- Let $X = \{X_n\}_{n \in \mathbb{N}}$, $Y = \{Y_n\}_{n \in \mathbb{N}}$ be ensembles.

  [Each $|X_n| = |Y_n| = \text{poly}(n)$]

  $X \overset{c}{\equiv} Y$ (i.e. X and Y are comp. ind.)

  If $Adv^{dist}_{X_n,Y_n}(\mathcal{D}) = \text{negl}(n)$ $\forall$ nppt $\mathcal{D}$

Recap from last time

- Composition lemma:
  If $x \approx_c y$ then $B(x) \approx_c B(y)$.
- Hybrid lemma:
  Let $x^i = \{x_n^i\}_{n \in \mathbb{N}}$ where $i = [m]$, $m = \text{poly}(n)$.
  If $x^i \approx_c x^{i+1}$ for $i \in [m-1]$, then $x^1 \approx_c x^m$

Proofs in notes

Agenda for this lecture

- Announcements
- Recap from last time
- Pseudorandom Generators (PRGs)
- PRGs with polynomial stretch
- A concrete PRG from hardness of discrete logarithms
  - ”Blum-Micali”

Pseudorandom Generators (PRGs)

Recall OTP encryption. Perfectly secure, not very usable: key must be as long as msg!

PRG: OTP-like encryption, but from very short seed!

[Also many other uses.]

Pseudorandom Generators (PRGs)

A deterministic function $G: \{0,1\}^* \to \{0,1\}^*$ is a pseudorandom generator with output length $\ell(n) > n$ if:

- G can be computed by deterministic poly-time alg.
- $|G(x)| = \ell(|x|) > |x|$ [min value: $\ell(|x|) = |x|+1$]
- the ensemble $\{G(U_n)\} \approx_c U_{\ell(n)}$ [i.e., $G(U_n)$ is pseudorandom]

Question

Let G be a PRG. Must $H(x) = \overline{G(x)}$ be a PRG?

Three properties:

- H efficient? Yes.
- H expands? Yes.
- H pseudorandom? Yes [composition Lemma]

Agenda for this lecture

- Announcements
- Recap from last time
- Pseudorandom Generators (PRGs)
- PRGs => PRGs with polynomial stretch
- A concrete PRG from hardness of discrete logarithms
  - ”Blum-Micali”

Expanding a PRG

"Smallest" stretch is one bit. Can we do more?

Yes!

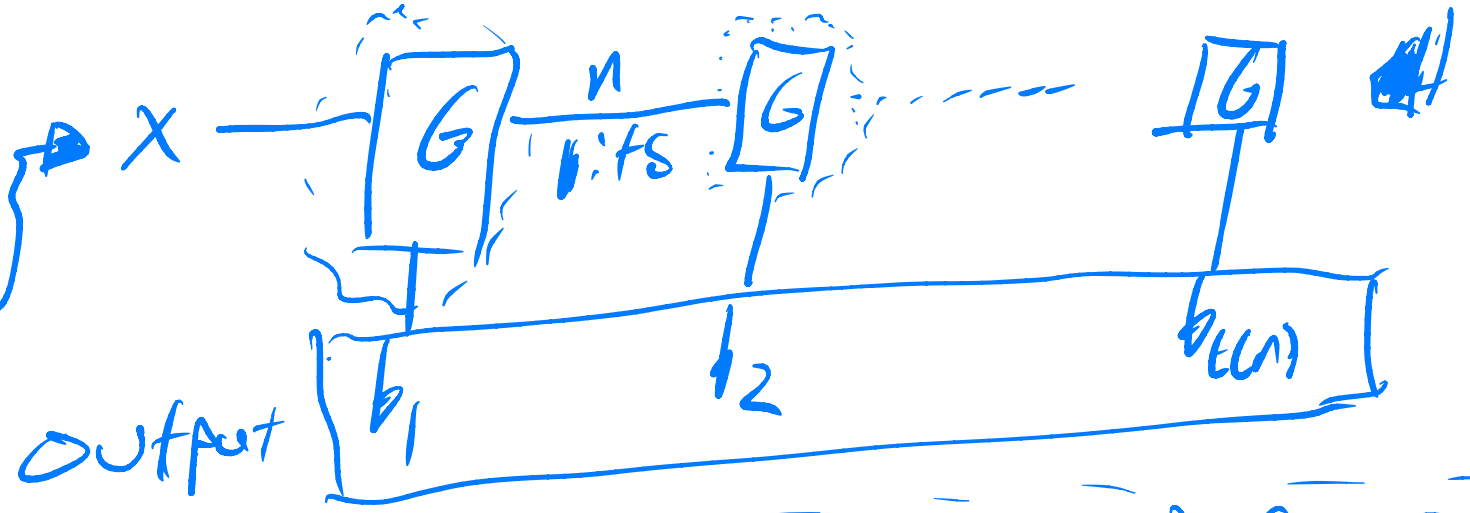
Theorem: Suppose $\exists$ PRG with one-bit stretch. Then $\forall$ polys $t(n)$, $\exists$ PRG with output $t(n)$

Expanding a PRG

Theorem: Suppose $\exists$ PRG with one-bit stretch. Then $\forall$ polys $t(n)$, $\exists$ PRG with output $t(n)$

Proof idea:

Let $G(x)$ be PRG with one-bit stretch. Define $G_t(x)$ as



Three properties:

- G_t efficient ✓
- G_t expands ✓
- G_t pseudorandom?

Intuition: Define sequence of hybrids: Each G is replaced with random bits.

Expanding a PRG

Define H_i as follows:

- $H_0(x) = G_t(U_n)$ ← our PRG
- $H_1(x) = U_1 \mid G_{t-1}(U_n)$
- $H_i = U_i \mid G_{t-i}(U_n)$
- $H_t = U_t$ ← uniform bits

Need to show $H^i \approx_c H^{i+1}$ $\forall i$. Then apply hybrid lemma.

Expanding a PRG

- $H_0(x) = G_t(U_n)$
- $H_1(x) = U_1 \| G_{t-1}(U_n)$
- $H_i = U_i \| G_{t-i}(U_n)$
- $H_t = U_t$

Need to show $H^i \approx_c H^{i+1}$.

Define efficient "simulator":

$S_i(y \in \{0,1\}^{n+1})$:

parse y as $x \| b$, $b \in \{0,1\}$

Return $U_{i-1} \| b \| G_{t-i}(x)$

Claim: $S_i(G(U_n)) = H_{i-1}$ and

$S_i(U_{n+1}) = H_i$.

Because $G(U_n) \approx_c U_{n+1}$ and S_i is efficient, $H_{i-1} \approx_c H_i$.

Apply hybrid lemma to get $G_{t(n)} \approx_c U_{t(n)}$. ☒

Agenda for this lecture

- Announcements
- Recap from last time
- Pseudorandom Generators (PRGs)
- PRGs => PRGs with polynomial stretch
- A concrete PRG from hardness of discrete logarithms
  - ”Blum-Micali”

The Discrete Logarithm Problem

Euler's theorem: Let G be a finite abelian group.
For all $a \in G$, $a^{|G|} = 1$ $\quad |\mathbb{Z}_p^*|$

Special case: Let p be prime. $\forall a$, $a^{(p-1)} = 1 \bmod p$

Let p be prime. Then $\mathbb{Z}_p^*$ is cyclic: $\exists g$ s.t.
$\{g^1, g^2, \ldots g^{p-1}\}$ is $\mathbb{Z}_p^*$

E.g. $\mathbb{Z}_5^*$ $\{3^1, 3^2 = 4 \bmod 5, 3^3 = 2 \bmod 5, 3^4 = 1 \bmod 5\}$

3 generates $\mathbb{Z}_5^*$

The Discrete Logarithm Problem

Let $S(1^n)$ be ppt that outputs prime p and generator g. [of $\mathbb{Z}_p^*$]

$\forall$ nuppt A,

$$\Pr_{\substack{(p,g)\leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_p^*}}\left[A(\underbrace{p,g}_{\text{gdu}},y) = \underbrace{\log_g y}_{x \text{ s.t. } g^x = y}\right] = \text{negl}(n)$$

Next time: design PRG from discrete log

Blum-Micali PRG