Welcome to CS 355! – Spring 2026

"Advanced" Topics in Cryptography
↑ a.k.a. we have preregs: - CS255 or equiv.
- interest and enthusiasm :)

not to scare you!

Clarification: CS 355 is a "topics" class. Normally, taught by Dan Boneh's PhD students

Last year was an exception: Dan himself taught a course on ZK proofs.

This year: back to normal...

2025-26 Instructors

- Trisha Datta
- Aditi Partap
- Ro Nema ← me!

TA: Jasdeep Sidhu

Today's Agenda

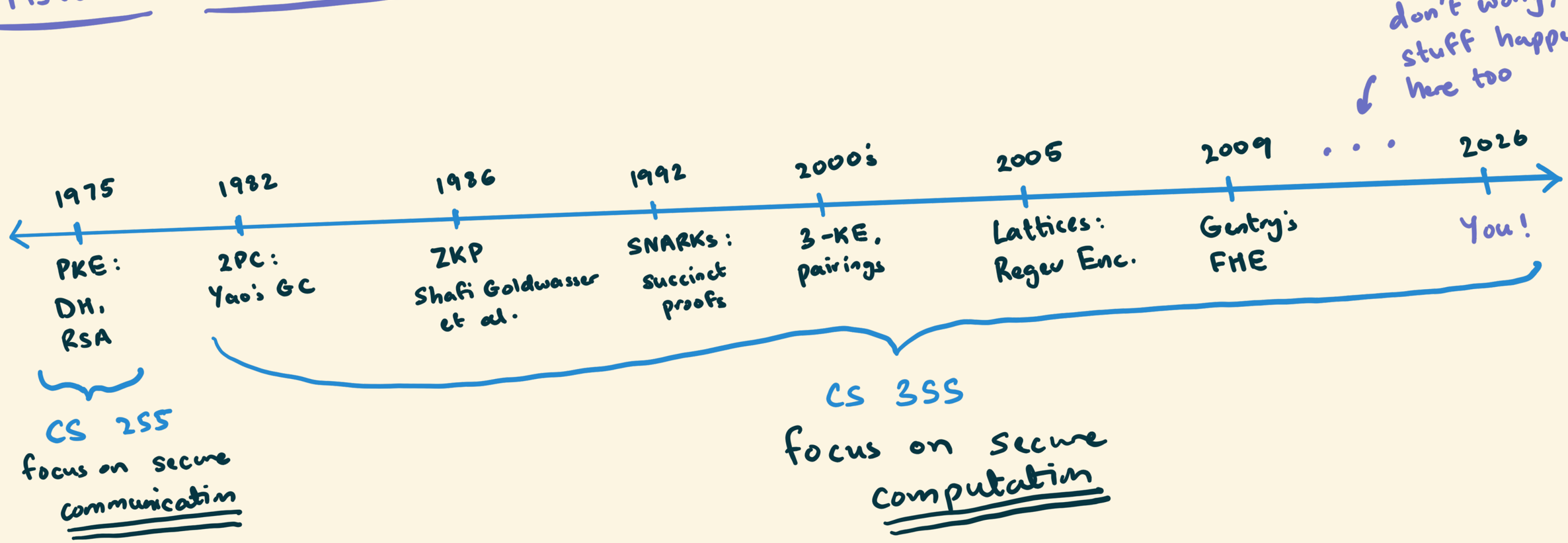
- About CS 355
- Course Logistics
- One-Way Functions (OWFs)
- Pseudorandom Generators (PRGs)
- Hardcore Bits

Course Goals

first advanced crypto course:
- learn techniques and formalisms
- understand open research problems
- prepare you for research!

or ... last advanced crypto course:
- applications of cutting-edge crypto
- digest new papers
- prepare you to use crypto to "change the world"!

Historical Perspective



About CS355

Topics

1. Foundations
2. Cryptanalysis
3. Elliptic Curve Cryptography
4. Zero Knowledge Proofs
5. Multiparty Computation
6. Lattice-based Cryptography

Why Take CS355?

- Cryptography is about the balance of power and privacy
- Widely deployed
- Beautiful math!

Why NOT Take CS 355?

- We assume mathematical maturity
  - should not be your first proof-based course
  - possible preregs.: CS 255, 265, 254; MATH 120, ...
- can be time-consuming

Course Logistics

Website: cs355.stanford.edu (we don't use Canvas)

Communication: ↙ don't spoil problems!

- Ed: Q&A, announcements
- Anonymous Feedback Form: welcome throughout the quarter :)
- cs355@cs.stanford.edu: email all 3 instructors

↑ you can also privately message us on Ed.

Lectures:

- in-person, not recorded
- attendance highly encouraged
- notes available after each class
- no textbook; supplemental readings available

↖ but there is Boneh & Shoup: cryptobook.us

Office Hours:

- see course calendar: both TA and Instructor have OH
- group work environment

Problem Sets :

- due every 2 weeks (5 total : each 20% of total grade)
- due on Fridays, 6 pm on Gradescope
- LaTeX mandatory. Template available. Overleaf.com works well!
- PSet 1 is out! Due Apr 10

- 3 late days total, at most 1 day for PSet 5
- No exams :)

Collaboration, AI, Academic Honesty

- encouraged to work together BUT write solutions by yourself

- list collaborators (obviously, no penalty for collaborating)

- AI / LLMs, in this class, are considered knowledgeable (if fallible) "friends" outside of this class (they are not collaborators). Thus, you are allowed to interact with them the same way you would interact with such friends

OK: build intuition, get ideas, learn a topic, et cetra

NOT OK: solve an entire problem, write up the solution, rob you of the joy of learning :(

Hardness & Reductions

- hardness and reductions are asymptotic

- honest parties are "efficient" (some polynomial in λ) ← tunable security parameter

- attacks are "inefficient" (super-polynomial or even exponential in λ)
  - e.g. decryption with secret key is <u>efficient</u>
    BUT <u>intractable/inefficient</u> without it

- Security ~~Proofs~~ Reductions!

Assumption.

Problem is hard ⇒ Cryptosystem is secure

Problem is easy (i.e., efficient solution to problem) ⇐ (efficient) Attack on cryptosystem

⇕

e.g. factoring, discrete log, CDH, DDH, "AES is a block cipher"

This week: Build crypto from one of the most minimal assumptions "One-Way Functions"

We'll show: OWFs => PRGs => PRFs => Block ciphers

Today: OWF => PRG_{+1} : PRGs with only 1-bit stretch

One-Way Functions (OWFs)

Let $\mathcal{X} := \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{Y} := \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ be families of finite sets indexed by sec. param. λ (natural numbers)

↖ input space ↖ output space

(e.g. $X_\lambda = \{0,1\}^\lambda$, $Y_\lambda = \{0,1\}^{p(\lambda)}$ for some poly. p)

Family of functions: $F = \{f_\lambda : X_\lambda \to Y_\lambda\}_{\lambda \in \mathbb{N}}$

intuitively, f_λ is "easy to compute" BUT "hard to invert"

we're leaving some stuff out to simplify the presentation for this course

Formally,

Def. F is a <u>one-way function family</u> if

(1) for all λ, f_λ is deterministic and computable in polynomial time
"easy to compute"

(2) for all <u>probabilistic polynomial-time</u> (PPT) algorithms $\mathcal{A}$,
(in λ; efficient)

there exists a "negligible" function $\text{negl}(\cdot)$ such that

$$\Pr_{x \xleftarrow{\$} X_\lambda,\ \mathcal{A}}\left[f_\lambda(\mathcal{A}(f_\lambda(x))) = f_\lambda(x) \right] \leq \text{negl}(\lambda)$$

- $\mathcal{A}$'s randomness
- "hard to invert" (on average)
- what does this mean??

Q: this formulation vs. $x = \mathcal{A}(f_\lambda(x))$.
What's the difference and why?

For the remainder of the course:
we'll drop the λ index when it's clear...
(means λ is "fixed" for the sake of presentation)

Negligible Function: formalizing "very small" in cryptography when dealing with asymptotics

A function g is negligible if

$$g(x) \in O(x^{-c}) \quad \forall c \in \mathbb{N}$$

← this is little-o not Big-O! although in this case it doesn't matter...

"asymptotically smaller than the inverse of any polynomial"

Examples. $2^{-\lambda}$, $2^{-\lambda/2}$, $2^{-\sqrt{\lambda}}$ are all negligible.

$$\| \quad \lambda^{\log_\lambda(2^{-\lambda})} = \lambda^{-\lambda \log_\lambda(2)} = \lambda^{-\lambda \cdot \frac{\log_2(2)}{\log_2(\lambda)}} = \lambda^{\underbrace{-\frac{\lambda}{\log_2(\lambda)}}}$$

← super-constant

$\Rightarrow O(\lambda^{-c}) \ \forall c \in \mathbb{N}$, grows slower than any polynomial

What does "$f(\lambda) \leq \text{negl}(\lambda)$" mean?

→ abuse of notation... just means f is negligible in λ

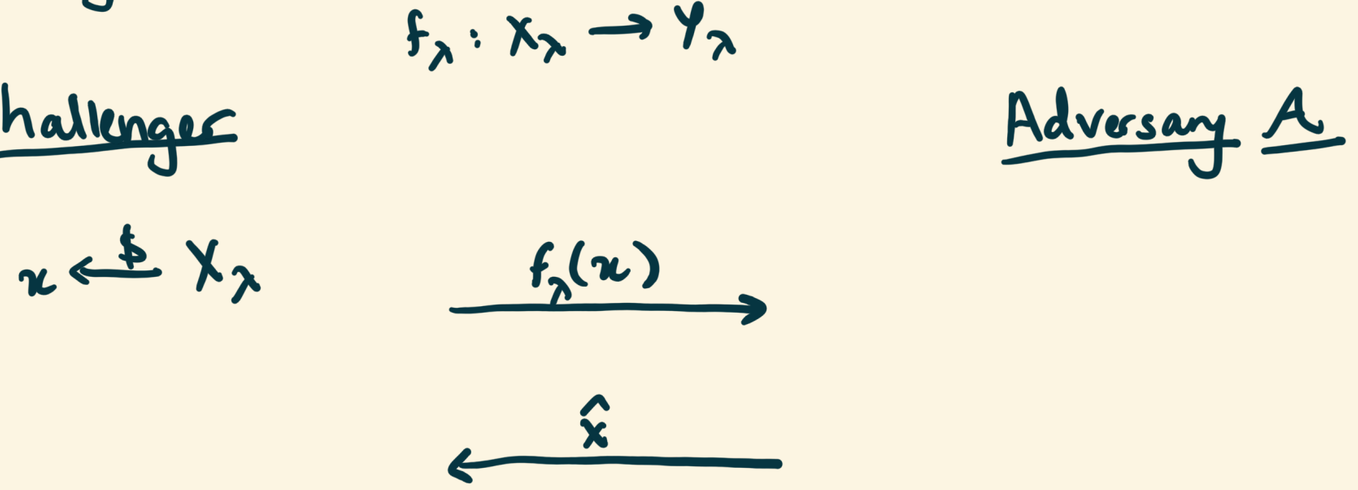
Helpful identities. (1) $f \leq \text{negl}$, $g \leq \text{poly} \Rightarrow f \cdot g \leq \text{negl}$

(2) $f > \text{negl}$, $g \leq \text{poly} \Rightarrow f/g > \text{negl}$

← not negl

(2) "Hard to invert" as a game:

$f_\lambda : X_\lambda \to Y_\lambda$



$$\mathrm{OWFAdv}[A,F](\lambda) = \Pr_{A}\left[f_\lambda(x) = f_\lambda(\hat{x})\right]$$

F is a one-way function family if $\forall$ PPT A,
there exists negligible function negl s.t.

$$\mathrm{OWFAdv}[A,F](\lambda) \leq \mathrm{negl}(\lambda)$$

Q: Can F be one-way if $X_\lambda = \{0,1\}^{\log(\lambda)}$?

A: No! Regardless of f_λ, consider (PPT) A that outputs $\hat{x} \xleftarrow{\$} X_\lambda$.

It has advantage $\geq 1/\lambda \not\leq \mathrm{negl}(\lambda)$

Candidate OWFs

(1) $X_\lambda = (x_1, x_2)$ where x_1, x_2 are λ-bit primes

$Y_\lambda = N$

$f_\lambda(x_1, x_2) := x_1 \cdot x_2$

Factoring

(2) $f_{p,g}(x) := g^x \pmod p$

p is λ-bit prime, $g \in \mathbb{Z}_p^*$, $x \in \mathbb{Z}_{p-1}$

Modular Exponentiation or DLOG

(3) $f(x_1, \ldots, x_n, S \subseteq [n]) = (x_1, \ldots, x_n, \sum_{i \in S} x_i)$

Subset Sum

(4) "Levin's One-Way Function": f_L is OWF $\iff \exists$ OWF

Pseudo-Random Generators

$G := \{ G_\lambda : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)} \}$ Takes random input and generates longer input that "looks random"

How do we formalize "looks random"?

The notion of (Computational) Indistinguishability!

Def. G is a PRG family if $\forall$ PPT $\mathcal{A}$,

$$\left| \Pr_{\mathcal{A}}[\mathcal{A}(G_\lambda(s)) = 1 : s \xleftarrow{\$} \{0,1\}^\lambda] - \Pr_{\mathcal{A}}[\mathcal{A}(r) = 1 : r \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}] \right| \leq \mathrm{negl}(\lambda)$$

$$\mathsf{PRGAdv}[\mathcal{A}, G](\lambda)$$

Cannot distinguish between $G(s)$ and random $\ell(\lambda)$-bit string

↑ random λ-bit string

OWFs $\Rightarrow$ PRG with $\ell(\lambda) = \lambda + 1$

Only one-bit stretch.

Assume OWF $f: \{0,1\}^\lambda \to \{0,1\}^\lambda$

From now on, we're sweeping the family formalization under the rug. We'll drop the λ subscript as well.

First Attempt.

Just leak the first bit of input

$g(x) = x_1 \| f(x)$

$= x_1 \| x_2 \| \cdots \| x_\lambda$ $\|$ is concatenation $\quad 1 \| 0 = 10$

g is not necessarily a PRG...

Turns out it's still one-way! Why?

Pf. By contradiction.

Suppose g is not one-way.

Then $\exists$ PPT $\mathcal{A}$ s.t.

$$\Pr[g(\mathcal{A}(g(x))) = g(x)] > \text{negl}(\lambda)$$

We will use $\mathcal{A}$ to construct PPT $\mathcal{B}$ s.t.

$$\Pr[f(\mathcal{B}(f(x))) = f(x)] > \text{negl}(\lambda)$$

$\mathcal{B}(y)$: Sample $b \xleftarrow{\$} \{0,1\}$;
Output $\mathcal{A}(b \| y)$.

Intuition. b = first bit of input with prob. $1/2$.
If that's the case, $\mathcal{A}$ outputs inverse with non-negl. prob and $\mathcal{B}$ "wins".

Formally, $\Pr[f(\mathcal{B}(f(x))) = f(x)]$

$\geq \Pr[f(\mathcal{B}(f(x))) = f(x) \wedge b = x_1]$

$= \Pr[f(\mathcal{B}(f(x))) = f(x) \mid b = x_1] \cdot \Pr[b = x_1]$

$= \Pr[f(\mathcal{A}(x_1 \| f(x))) = f(x)] \cdot 1/2$

$\geq \Pr[g(\mathcal{A}(g(x))) = g(x)] \cdot 1/2 > \text{negl}(\lambda)$

$> \text{negl}(\lambda)$ $\Rightarrow$ f is not one-way. $\lightning$

What went wrong? x_1 can be easy to guess given the output...
Can we derive a bit that is hard to guess?
We would like a predicate on the input that hides it even when given the output of the OWF!

Def. An efficient predicate $B : \{0,1\}^\lambda \longrightarrow \{0,1\}$ is "hard-core" for a OWF $f : \{0,1\}^\lambda \to Y$ if for all PPT $\mathcal{A}$,

$$\Pr\left[\mathcal{A}(f(x)) = B(x)\right] \leq \frac{1}{2} + \text{negl}(\lambda)$$

↖ not much better than random guessing

$$\Big|\Pr\left[\mathcal{A}(f(x)) = B(x)\right] - \frac{1}{2}\Big| \leq \text{negl}(\lambda)$$

$$!!$$

$$\text{HCBAdv}[\mathcal{A}, B, f]$$

We also call B a Hard-core Bit (HCB) of f

Theorem [Goldreich-Levin 89]. "Every OWF can be tranformed into another OWF with a HCB"

Idea. Take a random linear combination of the bits of x

Where is the randomness coming from? From the input!

Let f be a OWF.

Define $f'(x,r) := (f(x), r)$ $\quad |r| = |x| = \lambda$

f' is also a OWF. Why?

Claim. $B(x,r) := \sum_{i \in \lambda} x_i \cdot r_i \pmod 2$ is a HCB of f'.

Proof won't be covered in class. Resources will be available.
Involves non-trivial probability arguments. Proof usually built up by relaxing assumptions on the error in reduction.

In class: given HCB we can construct a PRG with 1-bit stretch!

Okay, I lied... We'll show something similar but very close.
We'll also assume our OWF is a permutation a.k.a. **OWP**
"one-way permutation"

Given OWP f and its HCB B*,

* if f is OWP, then f' is also OWP as defined in Goldreich-Levin Thm.

Construct PRG,

$$G(x) = f(x) \| B(x)$$

Thm. $f : \{0,1\}^\lambda \to \{0,1\}^\lambda$ OWP with HCB B

$$\Downarrow$$

$G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$ is a PRG.

Pf. Suppose not.

$\Rightarrow \exists$ PPT (Adv.) $\mathcal{A}$ s.t. $\text{PRGAdv}[\mathcal{A}, G] > \text{negl}(\lambda)$

i.e., (1) $\left| \Pr_{x,\mathcal{A}}[\mathcal{A}(f(x)\|B(x)) = 1] - \Pr_{r,\mathcal{A}}[\mathcal{A}(r) = 1] \right| > \varepsilon(\lambda)$ where $\varepsilon(\lambda)$ is a non-negl function.

in words, $\mathcal{A}$ can distinguish between random and $f(x)\|B(x)$

How do we use this? Notice that we can write $r \in \{0,1\}^{\lambda+1}$ as

$$r = y \| b \quad \text{where} \quad y \in \{0,1\}^\lambda, \; b \in \{0,1\}$$

Since f is a permutation,

$$r = f(x) \| b \quad \text{for some } x.$$

Moreover $x \xleftarrow{\$} \{0,1\}^\lambda$, $b \xleftarrow{\$} \{0,1\}$ $\Rightarrow r$ is uniformly random in $\{0,1\}^{\lambda+1}$

So, $\Pr_{r,\mathcal{A}}[\mathcal{A}(r)=1] = \Pr_{x,b,\mathcal{A}}[\mathcal{A}(f(x)\|b)=1]$

↖ when $b = B(x)$, this is just the first term

We can simplify: $\Pr[\mathcal{A}(f(x)\|b)=1] = \Pr[b=B(x)]\cdot\Pr[\mathcal{A}(f(x)\|b)=1]$
$+ \Pr[b=\overline{B(x)}]\cdot\Pr[\mathcal{A}(f(x)\|b]=1]$

$= \frac{1}{2}\cdot\Pr[\mathcal{A}(f(x)\|B(x)=1] + \frac{1}{2}\cdot\Pr[\mathcal{A}(f(x)\|\overline{B(x)})=1]$

Plugging this in ①,

① : $\left|\Pr[\mathcal{A}(f(x)\|B(x))=1] - \frac{1}{2}\Pr[\mathcal{A}(f(x)\|B(x)] = 1 - \frac{1}{2}\Pr[\mathcal{A}(f(x)\|\overline{B(x)})=1]\right| > \varepsilon(\lambda)$

$\Rightarrow \left|\Pr[\mathcal{A}(f(x)\|B(x))=1 - \Pr[\mathcal{A}(f(x)\|\overline{B(x)})=1]\right| > 2\varepsilon(\lambda)$ ↙ still non-negl.

What does this tell us? We can use $\mathcal{A}$ to distinguish if the HCB is flipped!

Construct adv. for HCB, $\mathcal{B}$ as follows:

$\mathcal{B}(y)$: Sample $b \xleftarrow{\$} \{0,1\}$
If $\mathcal{A}(y\|b)=1$: output b
Else : output $\overline{b}$

Claim. $\mathcal{B}$ "guesses" HCB with non-negl. probability

Pf. $\left|\Pr[\mathcal{B}(f(x)) = B(x)] - \frac{1}{2}\right| = \left|\Pr[\mathcal{B}(f(x)) = B(x) \mid b = B(x)] \cdot \Pr[b = B(x)]\right.$

$\left. + \Pr[\mathcal{B}(f(x)) = B(x) \mid b = \overline{B(x)}] \cdot \Pr[b = \overline{B(x)}] - 1/2\right|$

need to show this is non-negl.

$= \left|\Pr[\mathcal{A}(f(x) \| B(x)) = 1] \cdot \frac{1}{2} + \Pr[\mathcal{A}(f(x) \| \overline{B(x)}) = 0] \cdot \frac{1}{2} - \frac{1}{2}\right|$ (conditionally; $b = B(x)$, $b = \overline{B(x)}$)

$= \frac{1}{2}\left|\Pr[\mathcal{A}(f(x) \| B(x)) = 1] + (1 - \Pr[\mathcal{A}(f(x) \| \overline{B(x)}) = 1]) - 1\right|$

$= \frac{1}{2}\left|\Pr[\mathcal{A}(f(x) \| B(x) = 1] - \Pr[\mathcal{A}(f(x) \| \overline{B(x)}) = 1]\right|$

$> \frac{1}{2} \cdot (2\varepsilon(\lambda)) = \varepsilon(\lambda)$

Therefore, $\mathrm{HCBAdv}[B, \mathcal{B}, f] > \varepsilon(\lambda)$, non-negl. ↯

Using Goldreich-Levin transformation,

$$G(x, r) := f(x) \| r \| B(x, r) \qquad (x, r) \in \{0,1\}^{2\lambda}$$

$f(x) \| r$: $f'(x, r)$ is still OWF

$B(x, r)$: HCB of f'

G : 2λ bits to $2\lambda + 1$ bits!