Last Time

Hardcore Bits
Hardcore Bits

Let $F$ be a one-way function with domain $x$, range $y$

**Definition:** A function $h : x \rightarrow \{0, 1\}$ is a “hardcore bit” for $F$ if the following two distributions are computationally indistinguishable:
- $(F(x), h(x))$ for a random $x$
- $(F(x), b)$ for a random $x, b$

In other words, even given $F(x)$, hard to guess $h(x)$
Application: PRGs

Let $F$ be a one-way permutation with hardcore bit $h$

$$G(x) = (F(x), h(x))$$ is a secure PRG
Application: PRGs

Theorem: If $h$ is a hc bit for $F$ and $F$ is a OWP, then $G(x) = (h(x), h(F(x)), h(F(F(x))), \ldots )$ is a secure PRG
Examples of Hardcore Bits

Define $\text{lsb}(x)$ as the least significant bit of $x$

For $x \in \mathbb{Z}_N$, define $\text{Half}(x)$ as 1 iff $0 \leq x < N/2$
Theorem: Let $p$ be a prime, and $F: \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^*$ be $F(x) = g^x \mod p$, for some generator $g$

Half is a hardcore bit for $F$ (assume $F$ is one-way)

Theorem: Let $N$ be a product of two large primes $p, q$, and $F: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ be $F(x) = x^e \mod N$ for some $e$ relatively prime to $(p-1)(q-1)$

Lsb and Half are hardcore bits for $F$ (assuming RSA)

Theorem: Let $N$ be a product of two large primes $p, q$, and $F: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ be $F(x) = x^2 \mod N$

Lsb and Half are hardcore bits for $F$ (assuming factoring)
Random Self Reduction

Suppose given Dlog instance $y = g^x$

Have adversary that works for random Dlog instances
  • May not work for my particular instance

Nonetheless, want to use adversary to solve my instance
Random Self Reduction

Goal: randomize procedure that takes $y \rightarrow y'$
- From solution to $y'$, can compute solution to $y$
- $y'$ is uniformly random

Dlog random self reduction:
- Choose random $z$
- Let $y' \leftarrow y \times g^z$
- Run adversary on $y'$ to get Dlog $x'$
- $x = x' - z$
Today

Constructing PRPs with hardcore bits

Basing cryptography on on-way functions
Yao’s Method

Let $F$ be a OWF with domain $\{0,1\}^n$

**Claim:** $\exists i$ such that $\forall$ PPT $A$

$$\Pr[A(F(x)) = x_i] < 1 - 1/2^n$$

Proof: otherwise, $\forall i$, $\exists A_i$ s.t.

$$\Pr[A_i(F(x)) = x_i] \geq 1 - 1/2^n$$

Adversary $A(y) = A_1(y)||A_2(y)||...$

$$\Pr[A(F(x)) = x] \geq 1/2$$
Yao’s Method

Let $F$ be a OWF with domain $\{0,1\}^n$

Claim: $\exists i$ such that $\forall$ PPT $A$

$$\Pr[A(F(x)) = x_i] < 1 - 1/2^n$$

Let $F'(x^{(1)},...,x^{(t)}) = (F(x^{(1)}),...,F(x^{(t)}))$

$$h(x^{(1)},...,x^{(t)}) = x^{(1)}_i \oplus x^{(2)}_i \oplus ... \oplus x^{(t)}_i$$

Yao’s XOR lemma $\Rightarrow h$ is hardcore for $F'$
Goldreich Levin

Let $F$ be a OWF with domain $\{0,1\}^n$ and range $Y$

Let $F':\{0,1\}^{2n} \rightarrow \{0,1\}^n \times Y$ be:

$$F'(r,x) = r,F(x)$$

Define $h(r,x) = \langle r,x \rangle = \sum r_i x_i \mod 2$

**Theorem (Goldreich-Levin):** If $F$ is one-way, then $h$ is a hc bit for $F'$
Theorem (Goldreich-Levin): If $F$ is one-way, then $h$ is a hc bit for $F'$

Proof Sketch:

First attempt: suppose predicts $\langle x,r \rangle$ given $r,F(x)$ with certainty

Let $e_i = 0^{i-1}10^{n-i}$

Algorithm: $x_i \leftarrow (e_i, F(x))$
Theorem (Goldreich-Levin): If $F$ is one-way, then $h$ is a hc bit for $F'$

Second attempt: suppose predicts $\langle x,r \rangle$ given $r,F(x)$ with prob $3/4 + \varepsilon$

Claim: For an $\varepsilon/2$ fraction of $x$, predicts $\langle x,r \rangle$ given $r,F(x)$ for a random $r$ with prob $3/4 + \varepsilon/2$

Call such $x$ “good”

For rest of proof, assume we are given a “good” $x$
For "good" $x$, predicts $\langle x, r \rangle$ given $r, F(x)$ for a random $r$ with prob $3/4 + \varepsilon/2$

Want to perform $x_i \leftarrow \langle e_i, F(x) \rangle$ attack like before

- Problem: $e_i$ might not work on $e_i$

Solution: Random Self Reduction

- Choose random $r$
- $b_0 \leftarrow \langle r, F(x) \rangle$
- $b_1 \leftarrow \langle r \oplus e_i, F(x) \rangle$
- $Pr[x = b_0 \oplus b_1] = 1/2 + \varepsilon$
- Can increase accuracy by repeating multiple times
Theorem (Goldreich-Levin): If $F$ is one-way, then $h$ is a hc bit for $F'$

Second attempt: suppose predicts $\langle x, r \rangle$ given $r, F(x)$ with prob $1/2 + \varepsilon$

Can similarly define “good” $x$

Additional ideas required to get inverter
Summary

A hc bit for any OWF

Implies PRG from any OWP
• PRG from Dlog (Blum-Micali)
• PRG from Factoring
• PRG from RSA

Actually, can construct PRG from any OWF
• Proof beyond scope of course
So Far

OWP

TCR

CRH

OWF

PRG

Com

PRF

PRP

MAC

Enc

Auth

Enc

Plus arrows from everything to one-way functions
What’s Known

OWF

OWP

TCR

PRG

Com

PRF

PRP

MAC

Enc

Auth

Enc

Plus arrows from everything to one-way functions
What’s Known

OWP

CRH

OWF

PRG

Com

TCR

PRF

PRP

MAC

Auth

Enc

Enc
PRGs $\rightarrow$ PRFs

Today, we will show how to construct PRFs from PRGs

(Target collision resistance from one-way functions beyond scope of course)
First: Expanding Length of PRGs
A Different Approach
Advantage of Tree-based Approach

To expand $\lambda$ bits into $2^{h\lambda}$ bits, need $h$ levels

Can compute output locally:
• To compute $i$th chunk of $\lambda$ bits, only need $h$ PRG evaluations

In other words, can locally compute in logarithmic time
Advantage of Tree-based Approach

Theorem: For any logarithmic $h$, if $G$ is a secure PRG, then so is the tree-based PRG
Proof

Hybrid 0:
Proof

Hybrid 1:
Proof

Hybrid 2:
Proof

Hybrid 3:
Proof

Hybrid $t$: 

\[\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \]
Proof

What is $t$ in terms of $h$?

PRG adversary distinguishes Hybrid 0 from Hybrid $t$ with advantage $\varepsilon$

• $\exists i$ such that adversary distinguishes Hybrid $i-1$ from Hybrid $i$ with advantage $\varepsilon/t$
• Can use to construct adversary for $G$ with advantage $\varepsilon/t$
A PRF

Domain \( \{0, 1\}^n \)

Set \( h = n \)

\( F(k, x) \) is the \( x \)th block of \( \lambda \) bits

• Computation involves \( h \) evals of \( G \), so efficient
A PRF

$F(k,1)  F(k,2)  F(k,3)  F(k,4)  F(k,5)  F(k,6)  F(k,7)  F(k,8)  F(k,9)  F(k,10)  F(k,11)  F(k,12)  F(k,13)  F(k,14)$
Problem with Security Proof

Suppose I have a PRF adversary with advantage $\varepsilon$. In the proof, what is the advantage of the derived PRG adversary?
A Better Proof

Hybrid 0:
A Better Proof

Hybrid 1:
A Better Proof

Hybrid 2:
A Better Proof

Hybrid 3:
A Better Proof

Hybrid $h=n$: 

\[ \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \]
A Better Proof

Now if PRF adversary distinguishes Hybrid 0 from Hybrid $h=n$ with advantage $\epsilon$, $\exists i$ such that adversary distinguishes Hybrid $i-1$ from Hybrid $i$ with advantage $\epsilon/n$

- Non-negligible advantage

Not quite done: Distinguishing Hybrid $i-1$ from Hybrid $i$ does not immediately give a PRG distingusher

- Exponentially many PRG values changed!
A Better Proof

Hybrid $i-1$

Hybrid $i$
Key Observation:

Hybrid $i-1$

Adversary only queries polynomially many outputs
$\Rightarrow$ Only need to worry about polynomially many PRG instances in level $i$
A Better Proof

More Formally:

Given distinguisher $A$ for Hybrid $i-1$ and Hybrid $i$, can construct distinguisher $B$ for the following two oracles from $\{0,1\}^{i-1} \rightarrow \{0,1\}^{2^\lambda}$

- $H_0$: each output is a fresh random PRG sample
- $H_1$: each output is uniformly random

If $A$ makes $q$ queries, $B$ makes at most $q$ queries
A Better Proof

Now we have a distinguisher $B$ with advantage $\frac{\varepsilon}{n}$ that sees at most $q$ values, where either

- Each value is a random output of the PRG, or
- Each value is uniformly random

By introducing $q$ hybrids, can construct a PRG distinguisher with advantage $\frac{\varepsilon}{qn}$

$\Rightarrow$ non-negligible
What’s Known

What about OWP, CRH?
Generally Believed That...

Cannot construct OWP from OWF

Cannot construct CRH from OWF

Cannot construct CRH from OWP

Cannot construct OWP from CRH
Black Box Separations

How do we argue that you cannot build collision resistance from one-way functions?
• We generally believe both exist!

Observation: most natural constructions treat underlying objects as black boxes (don’t look at code, just input/output)

Maybe we can rule out such natural constructions
Black Box Separations

Present a world where one-way functions exist, but collision resistance does not

Hopefully, natural (black box) constructions make sense in this world
• Can construct PRGs, PRFs, PRPs, Auth-Enc, etc
Separating CRH from OWF

Starting point: random oracle model

Computation power is unlimited, but number of calls to random oracle is polynomial
Separating CRH from OWF

In ROM, despite unlimited computational power, one-way functions exist
• \( F(x) = H(x) \)
• Can only invert oracle by making exponentially-many calls

Unfortunately, collision resistant hashing exists too!
• \( F(x) = H(x) \)

To fix, also add collision finding oracle
Separating CRH from OWF
Separating CRH from OWF

What does $\text{CF}$ do?
- Takes as input a circuit $C$
- Circuit may have “oracle gates” that make calls to $H$ or $\text{CF}$
- Outputs a collision for $C$

Impossibility of Collision Resistance?
- Consider BB construction of CRHF from OWF
- Replace calls to OWF with $H$ queries
- Feed circuit computing CRHF to $\text{CF}$ to find collision
Separating CRH from OWF

So we have a world in which collision resistance does not exist

However, maybe **CF** can be used to invert **H**
- So maybe one-way functions don’t exist either

Must be careful in defining **CF**
- Random pair of colliding inputs will allow for inverting **H**
Separating CRH from OWF

Correct CF:
• Choose random input $\mathbf{x}$ to circuit
• Choose random input $\mathbf{y}$ that collides with $\mathbf{x}$

Note that $\mathbf{x}$ will sometimes equal $\mathbf{y}$. However, if circuit shrinks input, then with probability at least $\frac{1}{2}$ $\mathbf{x} \neq \mathbf{y}$

Careful analysis shows that $\mathbf{H}$ is still one-way
Next Time

Begin public key cryptography

Key agreement: how to exchange keys without ever meeting