COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

Recap

Discrete Log

Factoring

Discrete Log

Let $\mathbf{p}$ be a large integer (maybe prime)

Given $\mathbf{g} \in \mathbb{Z}_{\mathbf{p}}^{*}$, $\mathbf{a} \in \mathbb{Z}$, easy to compute $\mathbf{g^a \bmod p}$

- Time $\mathbf{poly(\log a, \log p)}$

However, no known efficient ways to recover $\mathbf{a}$ from $\mathbf{g}$ and $\mathbf{g^a \bmod p}$

Applications of Discrete Log

One-way functions

Collision resistance

- Key space = $\mathbf{G}^2$, $\mathbf{G}$ has prime order $\mathbf{p}$
- Domain: $\mathbb{Z}_p{}^2$
- Range: $\mathbf{G}$
- $\mathbf{H(\ (g,h),\ (x,y)\ ) = g^x h^y}$

Blum-Micali PRG

Let $\mathbf{G}=\mathbb{Z}_p^*$

Let $\mathbf{g}$ be a generator of $\mathbf{G}$

Let $\mathbf{h}$:$\mathbf{G} \rightarrow \{0,1\}$ be $\mathbf{h(x) = 1}$ **if** $0<x<(p-1)/2$

Seed space: $\mathbb{Z}_p^*$

Algorithm:

- Let $\mathbf{x_0}$ be seed
- For $\mathbf{i=0,\ldots}$
    - Let $x_{i+1} = g^{x_i} \bmod p$
    - Output $h(x_i)$

Stronger Assumptions on Groups

Sometimes, the discrete log assumption is not enough

Instead, define stronger assumptions on groups

Computational Diffie-Hellman:

- Given (g,g^a,g^b), compute g^{ab}

Decisional Diffie-Hellman:

- Distinguish (g,g^a,g^b,g^c) from (g,g^a,g^b,g^{ab})

Hard Problems on Groups

Increasing Difficulty

DLog:

- Given (g, g^a), compute a

CDH:

- Given (g, g^a, g^b), compute g^{ab}

DDH:

- Distinguish (g, g^a, g^b, g^c) from (g, g^a, g^b, g^{ab})

Stronger Assumptions

Naor-Reingold PRF

Domain: $\{0,1\}^n$

Key space: $\mathbb{Z}_p^{n+1}$

Range: $\mathbb{G}$

$$F(\ (a,b_1,b_2,\ldots,b_n),\ x\ ) = g^{a\ b_1^{x1}\ b_2^{x2}\ \ldots\ b_n^{xn}}$$

Theorem: If the discrete log assumption holds on $\mathbb{G}$**,** then the Naor-Reingold PRF is secure

Integer Factorization

Given an integer **N**, factor **N** into its prime factors

Studied for centuries, presumed computationally difficult

- Grade school algorithm: $O(N^{1/2})$
- Much better algorithms:

$$\exp\left(C\, (\log N)^{1/3}\, (\log \log N)^{2/3} \right)$$

- However, all require super-polynomial time

One-way Functions From Factoring

P_λ = {λ-bit primes}

$F: P_\lambda^2 \rightarrow \{0,1\}^{2\lambda}$
$F(p,q) = p \times q$

Trivial Theorem: If factoring assumption holds, then **F** is one-way

Another OWF

Fix a large integer $\mathbf{N = pq}$

- Primes $\mathbf{p,q}$ random, unknown

$$F_N(x) = x^2 \bmod N$$

Theorem: If the factoring assumption holds, then F is one-way: given $\mathbf{y}$, computationally infeasible to compute an x such that $x^2 = y \bmod N$

Chinese Remainder Theorem

Let $\mathbf{N} = \mathbf{pq}$ for co-prime $\mathbf{p,q}$

Let $\mathbf{x} \in \mathbb{Z}_\mathbf{p}$, $\mathbf{y} \in \mathbb{Z}_\mathbf{q}$

Then there exists a unique integer $\mathbf{z} \in \mathbb{Z}_\mathbf{N}$ such that

- $\mathbf{x = z \bmod p}$, and
- $\mathbf{y = z \bmod q}$

Proof: $\mathbf{z = [py(p^{-1} \bmod q)+qx(q^{-1} \bmod p)] \bmod N}$

Chinese Remainder Theorem

Let $\mathbf{N = pqr\ldots}$ for co-prime $\mathbf{p,q,r}$

Let $\mathbf{x}\in\mathbb{Z}_{\mathbf{p}}$, $\mathbf{y}\in\mathbb{Z}_{\mathbf{q}}$, $\mathbf{w}\in\mathbb{Z}_{\mathbf{r}}$

Then there exists a unique integer $\mathbf{z}\in\mathbb{Z}_{\mathbf{N}}$ such that

- $\mathbf{x = z \bmod p}$
- $\mathbf{y = z \bmod q}$
- $\mathbf{w = z \bmod r}$

Proof: $\mathbf{z = [(pr\ldots)y((pr\ldots)^{-1} \bmod q)}$
$\mathbf{+(qr\ldots)x((qr\ldots)^{-1} \bmod p)}$
$\mathbf{+(pq\ldots)w((pq\ldots)^{-1} \bmod r + \ldots\ ] \bmod N}$

Today

More constructions from factoring

One-way permutations

Hardcore bits

Collision Resistance from Factoring

Let $N=pq$, y a QR mod N
Suppose -1 is not a QR mod N

Hashing key: (N,y)
Domain: $\{1,\dots,(N-1)/2\}\times\{0,1\}$
Range: $\{1,\dots,(N-1)/2\}$

$H(\ (N,y),\ (x,b)\ )$: Let $z = y^b x^2 \bmod N$
- If $z\in\{1,\dots,(N-1)/2\}$, output z
- Else, output $-z \bmod N\ \in\{1,\dots,(N-1)/2\}$

Theorem: If the factoring assumption holds, $\mathbf{H}$ is collision resistant

Proof:

- Collision means $(x_0,b_0)\neq(x_1,b_1)$ s.t.

$$y^{b0}\ x_0^{\ 2} = \pm\ y^{b1}\ x_1^{\ 2} \bmod N$$

- If $b_0=b_1$, then $x_0\neq x_1$, but $x_0^{\ 2}=\pm x_1^{\ 2} \bmod N$
  - $x_0^{\ 2}=\pm x_1^{\ 2}$ mod $\mathbf{N}$ not possible. Why?
  - $x_0\neq -x_1$ since $x_0,x_1\in\{1,\ldots,(N-1)/2\}$
  - $\mathrm{GCD}(x_0-x_1,N)$ will give factor

- If $b_0\neq b_1$, then $(x_0/x_1)^2 = \pm y^{\pm 1} \bmod N$
  - (x_0/x_1) or (x_1/x_0) is a square root of $\pm y$
  - $-y$ case not possible. Why?

Choosing **N**

How to choose **N** so that **-1** is not a QR?

By CRT, need to choose **p,q** such that -1 is not a QR mod **p** or mod **q**

Fact: if $\mathbf{p = 3 \bmod 4}$, then **-1** is not a QR mod **p**
Fact: if $\mathbf{p = 1 \bmod 4}$, then **-1** is a QR mod **p**

Is Composite N Necessary for SQ to be hard?

Let p be a prime, and suppose $p = 3 \bmod 4$

Given a QR x mod p, how to compute square root?

Hint: recall Fermat: $x^{p-1}=1$ mod p for all $x\neq 0$

Hint: what is $x^{(p+1)/2}$ mod p?

Solving Quadratic Equations

In general, solving quadratic equations is:

- Easy over prime moduli
- As hard as factoring over composite moduli

Other Powers?

What about $\mathbf{x \rightarrow x^4 \bmod N}$? $\mathbf{x \rightarrow x^6 \bmod N}$?

The function $\mathbf{x \rightarrow x^3 \bmod N}$ appears quite different

- Suppose $\mathbf{3}$ is relatively prime to $\mathbf{p-1}$ and $\mathbf{q-1}$
- Then $\mathbf{x \rightarrow x^3 \bmod p}$ is injective for $\mathbf{x \neq 0}$
  - Let $\mathbf{a}$ be such that $\mathbf{3a = 1 \bmod p-1}$
  - $\mathbf{(x^3)^a = x^{1+k(p-1)} = x(x^{p-1})^k = x \bmod p}$
- By CRT, $\mathbf{x \rightarrow x^3 \bmod N}$ is injective for $\mathbf{x} \in \mathbb{Z}_{\mathbf{N}}^{\mathbf{*}}$

$x^3 \bmod N$

What does injectivity mean?

Cannot base of factoring:

Adapt alg for square roots:

- Choose a random $z \bmod N$
- Compute $y = z^3 \bmod N$
- Run inverter on y to get a cube root x
- Let $p = GCD(z-x, N)$, $q = N/p$

RSA Problem

Given

- $\mathbf{N = pq}$,
- $\mathbf{e}$ such that $\mathbf{GCD(e,p-1)=GCD(e,q-1)=1}$,
- $\mathbf{y=x^e \bmod N}$ for a random $\mathbf{x}$

Find $\mathbf{x}$

Injectivity means cannot base hardness on factoring, but still conjectured to be hard

One-way permutations

A one-way function that is also a permutation

Examples:
- The RSA function $\mathbf{x} \rightarrow \mathbf{x^e \bmod N}$
- Almost: discrete exponentiation: $\mathbf{x} \rightarrow \mathbf{g^x}$

Hardcore Bits

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

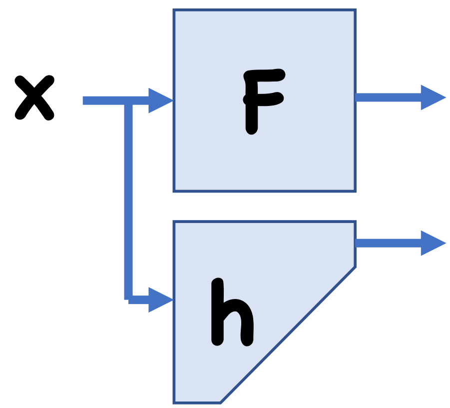
Definition: A function **h:x→{0,1}** is a "hardcore bit" for **F** if the following two distributions are computationally indistinguishable:

- **(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)**

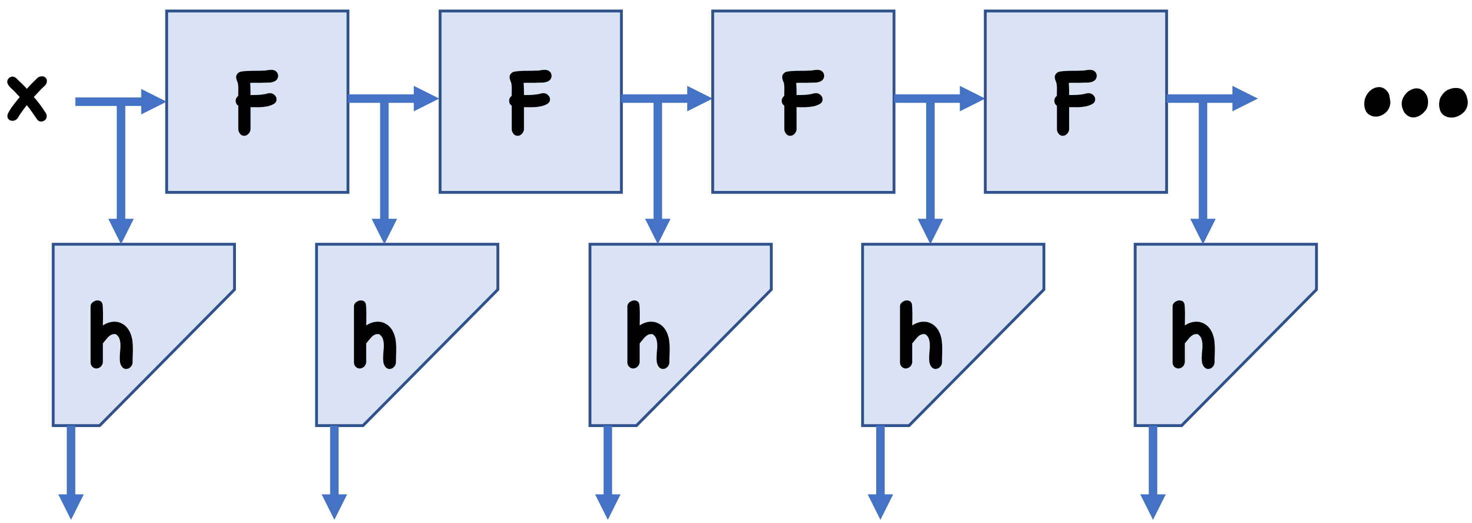
Application: PRGs

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



Theorem: If **h** is a hc bit for **F** and **F** is a OWP, then
G(x) = (F(x), h(x)) is a secure PRG

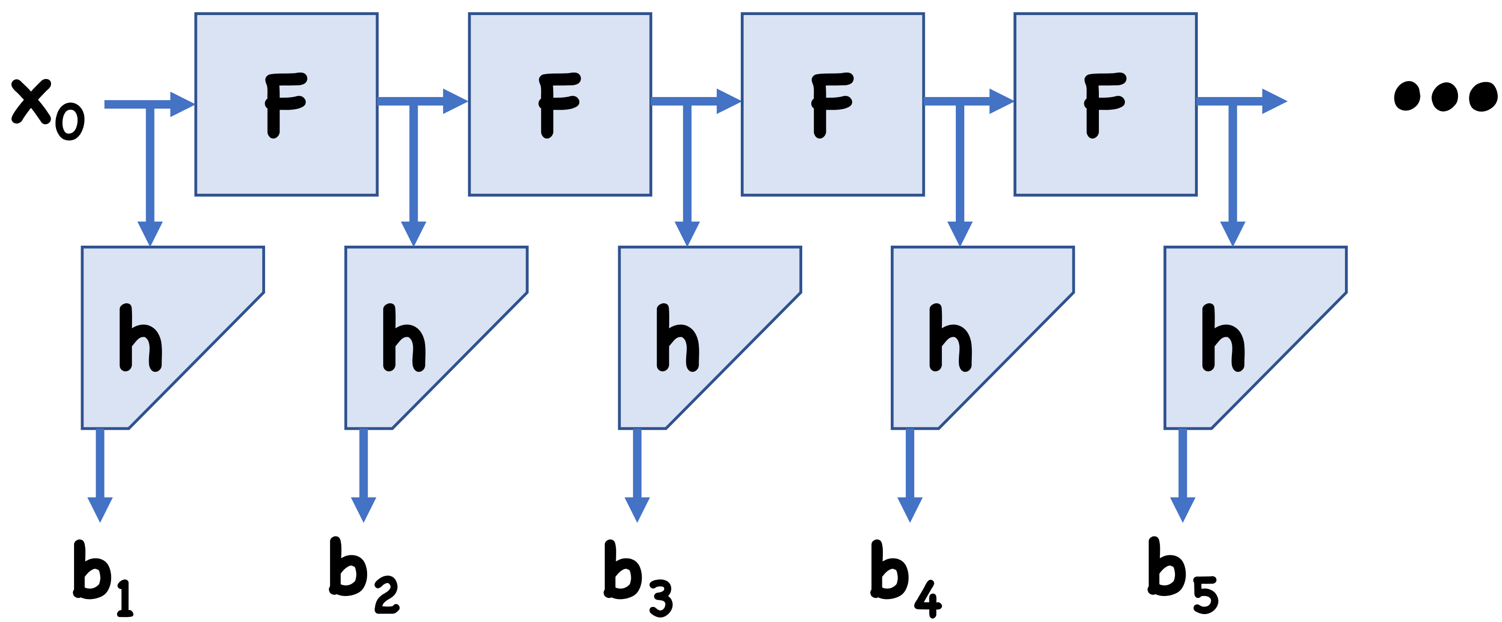
Application: PRGs



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)), ...)** is a secure PRG

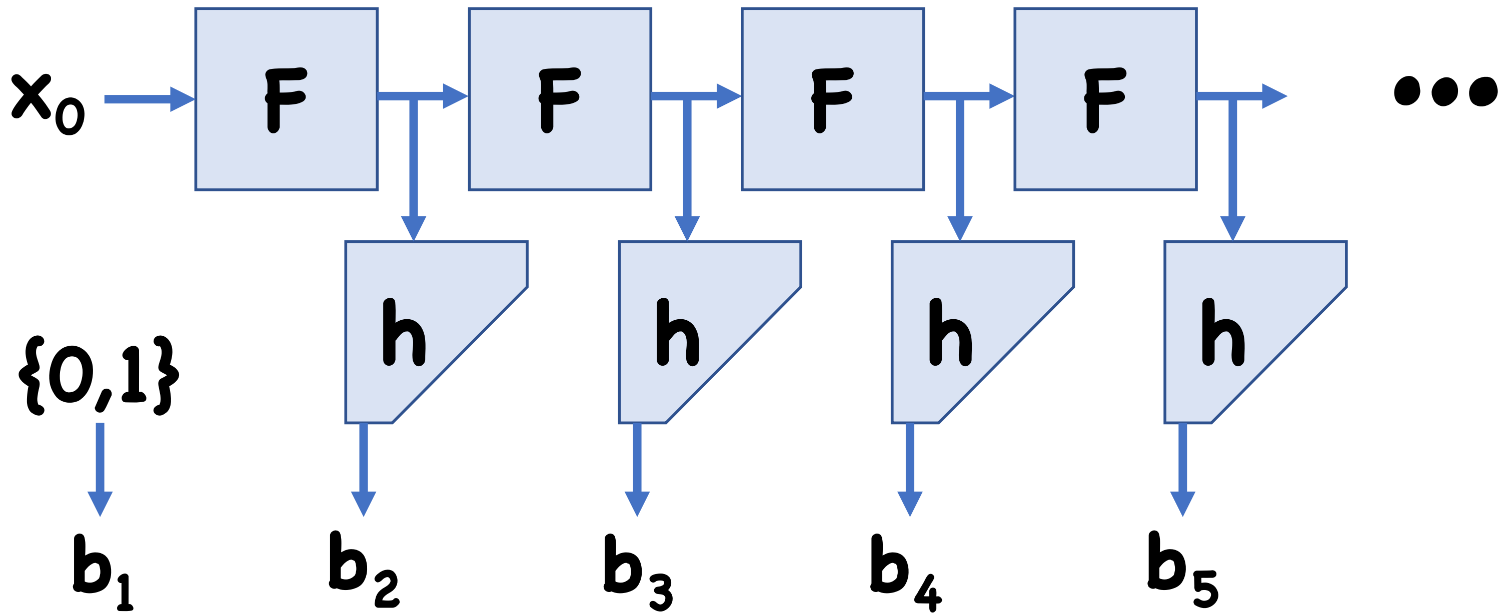
Proof by Hybrids

Hybrid 0:



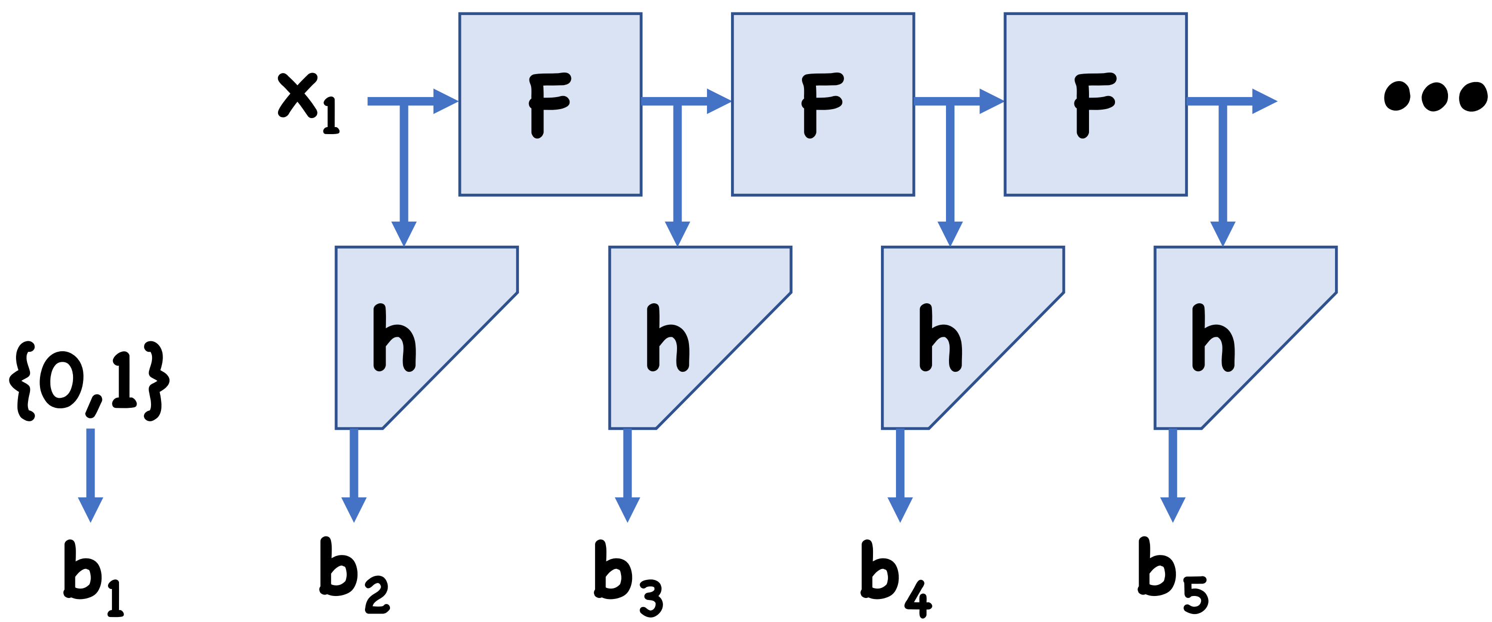
Proof by Hybrids

Hybrid 1:



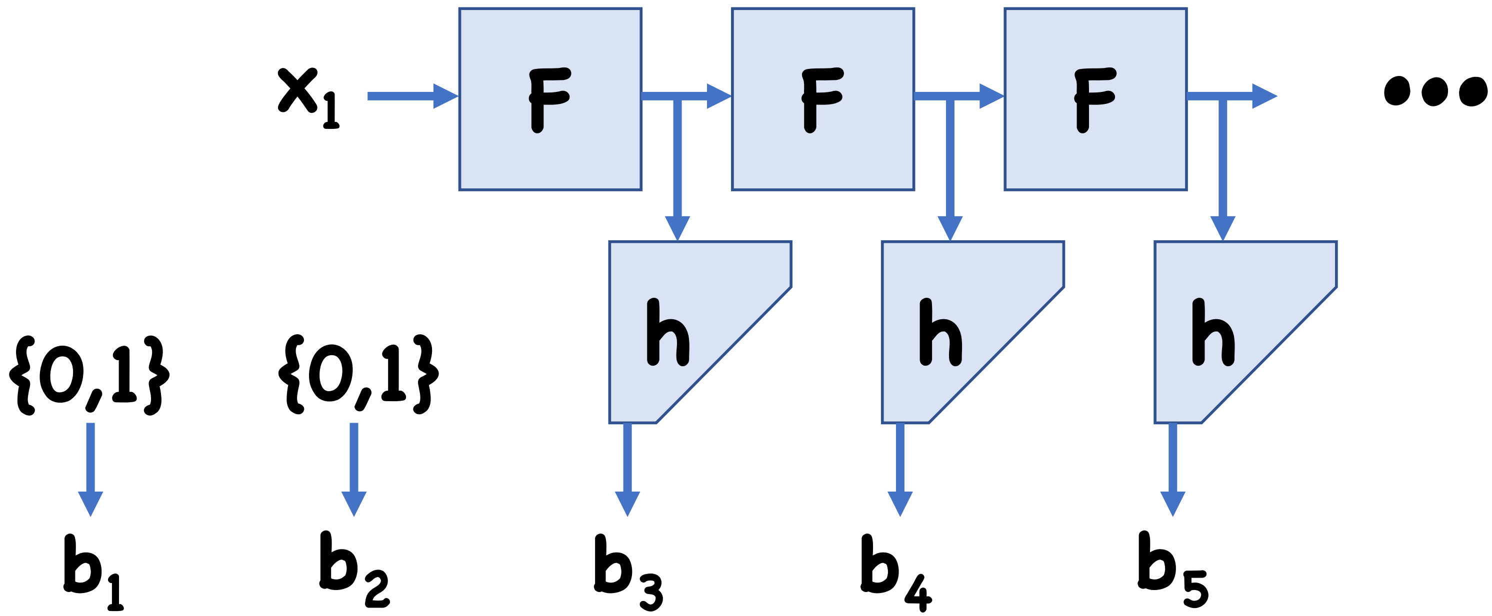
Proof by Hybrids

Hybrid 1:



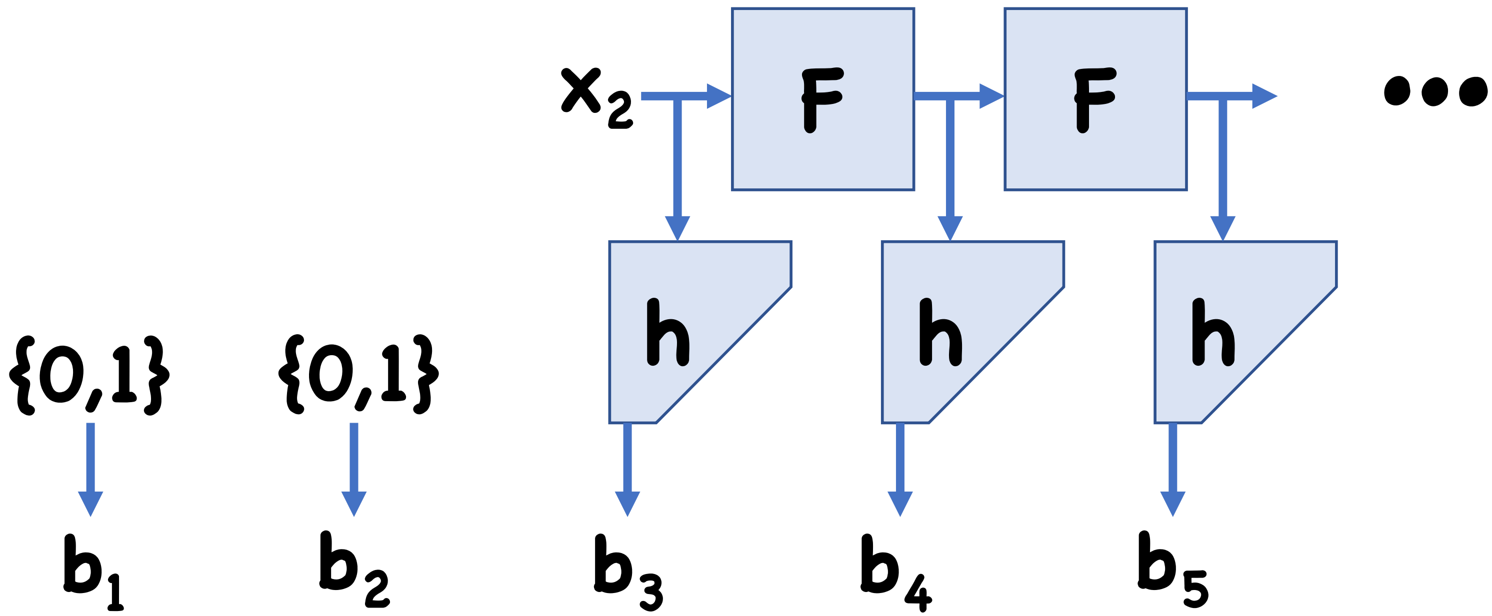
Proof by Hybrids

Hybrid 2:



Proof by Hybrids

Hybrid 2:



Proof by Hybrids

Hybrid 2:

$$\begin{array}{cccccc} \{0,1\} & \{0,1\} & \{0,1\} & \{0,1\} & \{0,1\} & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \cdots \\ b_1 & b_2 & b_3 & b_4 & b_5 & \end{array}$$

Examples of Hardcore Bits

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

For $\mathbf{x} \in \mathbf{Z_N}$, define $\mathbf{Half(x)}$ as $\mathbf{1}$ iff $\mathbf{0 \leq x < N/2}$

Theorem: Let $\mathbf{p}$ be a prime, and $\mathbf{F: Z_p^* \rightarrow Z_p^*}$ be $\mathbf{F(x) = g^x \bmod p}$, for some generator $\mathbf{g}$

Half is a hardcore bit for $\mathbf{F}$ (assume $\mathbf{F}$ is one-way)

Proof Sketch

Need to show: if there is a PPT adversary that can predict **Half(x)** given **F(x)** with non-negligible advantage, then there is a PPT adversary that can compute **x** given **F(x)** with non-negligible probability

Will instead show: if there is a PPT adversary that can predict **Half(x)** given **F(x)** with certanty, then there is a PPT adversary that can compute **x** given **F(x)** with certainty

Inverter

Let  be an adversary that predicts **Half(x)**:

Pr[ **(F(x)) = Half(x)] = 1**

Given $y=F(x)$, do the following:

- Let $x' = 0$
- Run $b_1 \leftarrow$  (y)
- If $b_1 = 1$, (meaning x is odd)
  - $x' \mathrel{+}= p/2$,
  - $y_1 \leftarrow (y/g^{p/2})^2 = g^{2(x-p/2)}$ (so $x_1 \leftarrow 2(x-p/2)$)
- Else (meaning x is even)
  - $y_1 \leftarrow y^2 = g^{2x}$ (so $x_1 \leftarrow 2x$)
- Run $b_2 \leftarrow$  (y_1)
- ...

Extending to Non-perfect Adversary

Couple problems with low-advantage adversaries:

- Distribution of $\mathbf{x}_1$, … not random, adversary not guaranteed to work on these
    - $\mathbf{x}_1$ is even
    - $\mathbf{x}_2$ is divisible by 4
    - …
- Extremely unlikely all $\mathbf{b}_i$ are correct
    - If any $\mathbf{b}_i$ is wrong, get completely wrong answer

Boosting Advantage?

Idea 1: run adversary multiple times

Random Self Reduction

Suppose given Dlog instance $\mathbf{y=g^x}$

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 $\mathbf{y} \rightarrow \mathbf{y'}$

- From solution to $\mathbf{y'}$, can compute solution to $\mathbf{y}$
- $\mathbf{y'}$ is uniformly random

Dlog random self reduction:

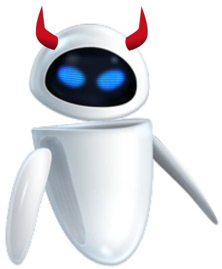
- Choose random $\mathbf{z}$
- Let $\mathbf{y'} \leftarrow \mathbf{y \times g^z}$
- Run adversary on $\mathbf{y'}$ to get Dlog $\mathbf{x'}$
- $\mathbf{x} =$?

Boosting Dlog advantage

Theorem: Let  be a PPT Dlog adversary with non-negligible advantage **ε**.

Then there is a PPT Dlog adversary  that, for any instance **y**, outputs the Dlog with probability 1-negl

Proof

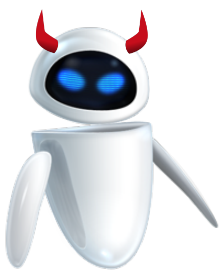


$y_1 = y \times g^{z1}$

x_1

If $g^{x1-z1} = y$, Output $x_1 - z_1$

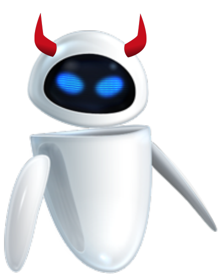
$y = g^x$



$y_2 = y \times g^{z2}$

x_2

If $g^{x2-z2} = y$, Output $x_2 - z_2$



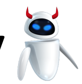
$y_3 = y \times g^{z3}$

x_3

If $g^{x3-z3} = y$, Output $x_3 - z_3$

…

Analysis

In iteration $\mathbf{i}$, probability  outputs Dlog of $\mathbf{y_i}$: $\boldsymbol{\varepsilon}$

- If so, then $\mathbf{y \times g^{zi} = y_i = g^{xi}}$, so $\mathbf{y = g^{xi-zi}}$
- Therefore, $\mathbf{Pr[}$ succeeds in iter $\mathbf{i] = \varepsilon}$

Pr[ succeeds**] = 1 − Pr[**  fails**]**

$\mathbf{= 1 - (Pr[}$ fails in iter $\mathbf{i])^t}$

$$\mathbf{= 1 - (1-\varepsilon)^t}$$

By setting $\mathbf{t = (1/\varepsilon)\times\lambda}$, success prob is

$$\mathbf{\approx 1 - e^{-\lambda} = 1\text{-}negl}$$

Random Self Reduction RSA Func?

$F(x) = x^e \bmod N$

Theorem: Let **N** be a product of two large primes **p,q**, and $\mathbf{F}: Z_N^* \rightarrow Z_N^*$ be $\mathbf{F(x) = x^e \bmod N}$ 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 $\mathbf{F}: Z_N^* \rightarrow Z_N^*$ be $\mathbf{F(x) = x^2 \bmod N}$

Lsb and Half are hardcore bits for **F** (assuming factoring)

Is Half Hardcore for any OWF?

No

Proof?

- Given hardcore bit **h**
- Start with any OWF **F**, construct a OWF **F'** such that **h** is not hardcore for **F**

General HC Bits

Given any OWF F, can construct another OWF $\mathsf{F'}$ that has a HC bit

Yao's Method

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

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

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

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

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

Adversary $A(y) = A_1(y) || A_2(y) || \ldots$

$$\Pr[A(F(x)) = x] \geq 1/2$$

Yao's Method

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

Claim: $\exists\, \mathbf{i}$ such that $\forall$ PPT $\mathbf{A}$

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

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

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

Yao's XOR lemma $\Rightarrow$ $\mathbf{h}$ is hardcore for $\mathbf{F'}$

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 $h(r,x) = \langle r,x \rangle = \Sigma r_i x_i \bmod 2$

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'**

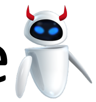
Proof Sketch:

First attempt: suppose  predicts $\mathbf{<x,r>}$ given $\mathbf{r,F(x)}$ with certainty

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

Algorithm: $\mathbf{x_i} \leftarrow$  $\mathbf{(e_i,\ F(x)\ )}$

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

Second attempt: suppose  predicts **<x,r>** given **r,F(x)** with prob **3/4 + ε**

Claim: For an **ε/2** fraction of **x**,  predicts **<x,r>** given **r,F(x)** for a random **r** with prob **3/4 + ε/2**

Call such **x** “good”

For rest of proof, assume we are given a “good” **x**

For "good" $\mathbf{x}$,  predicts $\mathbf{\langle x,r\rangle}$ given $\mathbf{r, F(x)}$ for a random $\mathbf{r}$ with prob $\mathbf{3/4 + \varepsilon/2}$

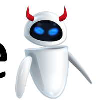
Want to perform $\mathbf{x_i} \leftarrow$  $\mathbf{(e_i,\ F(x)\ )}$ attack like before

- Problem:  might not work on $\mathbf{e_i}$

Solution: Random Self Reduction

- Choose random $\mathbf{r}$
- $\mathbf{b_0} \leftarrow$  $\mathbf{(r,\ F(x)\ )}$
- $\mathbf{b_1} \leftarrow$  $\mathbf{(r\oplus e_i,\ F(x)\ )}$
- $\mathbf{Pr[x=b_0\oplus b_1] = 1/2 + \varepsilon}$
- 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 **<x,r>** given **r,F(x)** with prob **1/2 + ε**

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

Next Time

OWF imply most crypto we’ve seen so far

- Namely, PRG → PRFs

But not everything

- No collision resistance