Midterm Statistics

Minimum Value	56.1
Maximum Value	93.8
Range	37.7
Average	78.72
Median	79.15
Standard Deviation	9.51

Final Exam

- **Time:** Tuesday, December 12th at 1PM (Tentative Subject to Change)
- **Location:** LWSN 1106

Cryptography
CS 555

Week 8:

- DES, 3DES, AES
- One Way Functions

Readings: Katz and Lindell Chapter 7

Feistel Networks and Substitution Permutation Networks

- Used to construct block-ciphers
- DES: Feistel Network
- AES: Substitution Permutation Network

CS 555: Week 8: Topic 1: DES, 3DES, AES

Data Encryption Standard

- Developed in 1970s by IBM (with help from NSA)
- Adopted in 1977 as Federal Information Processing Standard (US)
- Data Encryption Standard (DES): 16-round Feistel Network.
- Key Length: 56 bits
  - Vulnerable to brute-force attacks in modern times
  - 1.5 hours at 14 trillion keys/second (e.g., Antminer S9)

DES Round

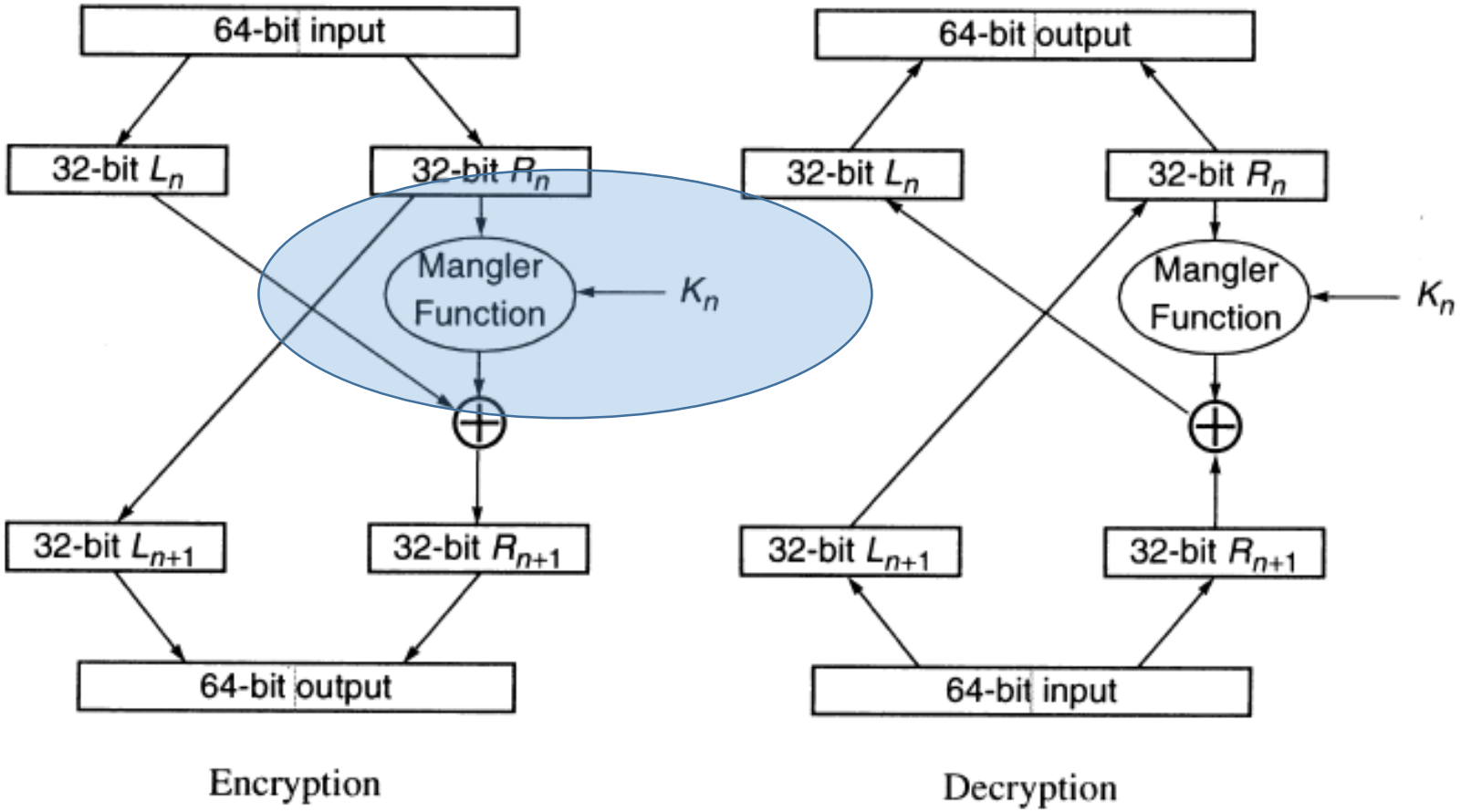


Figure 3-6. DES Round

DES Security

- Best Known attack is brute-force 2^{56}
  - Except under unrealistic conditions (e.g., 2^{43} known plaintexts)
- Brute force is not too difficult on modern hardware

- Attack can be accelerated further after precomputation
  - Output is a few terabytes
  - Subsequently keys are cracked in 2^{38} DES evaluations (minutes)
- Precomputation costs amortize over number of DES keys cracked

- Even in 1970 there were objections to the short key length for DES

Double DES

- Let $F_k(x)$ denote the DES block cipher
- A new block cipher F' with a key $k = (k_1, k_2)$ of length 2n can be defined by

$$F'_k(x) = F_{k_2}\left(F_{k_1}(x)\right)$$

- Can you think of an attack better than brute-force?

Meet in the Middle Attack

$$F'_k(x) = F_{k_2}\left(F_{k_1}(x)\right)$$

Goal: Given (x, c $= F'_k(x)$) try to find secret key k in time and space $O(n2^n)$.

- **Solution?**
  - **Key Observation**

$$F_{k_1}(x) = F_{k_2}^{-1}(c)$$

  - Compute $F_K^{-1}(c)$ and $F_K(x)$ for each potential key K and store

$$\left(K, F_K^{-1}(c)\right) \text{ and } \left(K, F_K\ (x)\right)$$

  - Sort each list of pairs (by $F_K^{-1}(c)$ or $F_K\ (x)$) to find K_1 and K_2.

Triple DES Variant 1

- Let $F_k(x)$ denote the DES block cipher
- A new block cipher F' with a key $k = (k_1, k_2, k_3)$ of length 2n can be defined by

$$F'_k(x) = F_{k_3}\left(F_{k_2}^{-1}\left(F_{k_1}(x)\right)\right)$$

- Meet-in-the-Middle Attack Requires time $\Omega(2^{2n})$ and space $\Omega(2^{2n})$

Triple DES

Allows backward compatibility with DES by setting $k_1=k_2=k_3$

- Let $F_k(x)$ denote the DES block cipher
- A new block cipher F' with a key $k = (k_1, k_2, k_3)$ of length 2n can be defined by

$$F'_k(x) = F_{k_3}\left(F_{k_2}^{-1}\left(F_{k_1}(x)\right)\right)$$

- Meet-in-the-Middle Attack Requires time $\Omega(2^{2n})$ and space $\Omega(2^{2n})$

Triple DES

$$F'_k(x) = F_{k_3}\left(F^{-1}_{k_2}\left(F_{k_1}(x)\right)\right)$$

- Standardized in 1999
- Still widely used, but it is relatively slow (three block cipher operations)
  - Now viewed as ``weak cipher" by OpenSSL
- Current gold standard: AES

Advanced Encryption Standard (AES)

- (1997) US National Institute of Standards and Technology (NIST) announces competition for new block cipher to replace DES

- Fifteen algorithms were submitted from all over the world
  - Analyzed by NIST
- Contestants given a chance to break competitors schemes

- October, 2000 NIST announces a winner Rijndael
  - Vincent Rijmen and Joan Daemen
  - No serious vulnerabilities found in four other finalists
  - Rijndael was selected for efficiency, hardware performance, flexibility etc…

Advanced Encryption Standard

- **Block Size:** 128 bits (viewed as 4x4 byte array)
- **Key Size:** 128, 192 or 256
- First public cipher approved by NSA for Top Secret information

- (2009) Attack on 11 round version of AES
  - recovers 256-bit key in time 2^{70}
  - But AES is 14 round (with 256 bit key) so the attack doesn't apply in practice
- (2009) Attack on 192-bit and 256 bit version of AES
  - recovers 256-bit key in time $2^{99.5}$.

AES Attacks?

- Side channel attacks affect a few specific implementations
  - But, this is not a weakness of AES itself
  - Timing attack on OpenSSL's implementation AES encryption (2005, Bernstein)

CS 555: Week 8: Topic 1: One Way Functions

What are the minimal assumptions necessary for symmetric key-cryptography?

One-Way Functions (OWFs)

$$f(x) = y$$

Definition: A function f: $\{0,1\}^* \rightarrow \{0,1\}^*$ is one way if it is

1. **(Easy to compute)** There is a polynomial time algorithm (in |x|) for computing f(x).
2. **(Hard to Invert)** Select $x \leftarrow \{0,1\}^n$ uniformly at random and give the attacker input 1^n, f(x). The probability that a PPT attacker outputs x' such that $f(x') = f(x)$ is negligible.

One-Way Functions (OWFs)

$$\mathrm{f}(x) = y$$

Key Takeaway: One-Way Functions is a necessary and sufficient assumption for most of symmetric key cryptography.

- From OWFs we can construct PRGs, PRFs, Authenticated Encryption
- From eavesdropping secure encryption (weakest) notion we can construct OWFs

One-Way Functions (OWFs)

$$f(x) = y$$

Remarks:

- **A function that is not one-way is not necessarily always easy to invert (even often)**
- **Any such function can be inverted in time 2^n (brute force)**
- **Length-preserving OWF: |f(x)| = |x|**
- **One way permutation: Length-preserving + one-to-one**

One-Way Functions (OWFs)

$$\mathrm{f}(x) = y$$

Remarks:

1. **f(x) does not necessarily hide all information about x.**
2. **If f(x) is one way then so is $\mathbf{f}'(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \parallel \boldsymbol{LSB}(\boldsymbol{x})$.**

One-Way Functions (OWFs)

$$\mathrm{f}(x) = y$$

Remarks:

1. **Actually we usually consider a family of one-way functions**

$$f_I : \{0,1\}^I \to \{0,1\}^I$$

Candidate One-Way Functions (OWFs)

$$f_{p,g}(x) = [g^x \bmod p]$$

(Discrete Logarithm Problem)

Note: The existence of OWFs implies $\mathrm{P} \neq NP$ so we cannot be *absolutely certain* that they do exist.

Hard Core Predicates

- Recall that a one-way function f may potentially reveal lots of information about input

- **Example**: $f(x_1,x_2)=(x_1,g(x_2))$, where g is a one-way function.
- **Claim**: f is one-way (even if $f(x_1,x_2)$ reveals half of the input bits!)

Hard Core Predicates

Definition: A predicate $\mathrm{hc}: \{0,1\}^* \to \{0,1\}$ is called a hard-core predicate of a function f if

1. (Easy to Compute) hc can be computed in polynomial time
2. (Hard to Guess) For all PPT attacker A there is a negligible function negl such that we have

$$\mathbf{Pr}_{x \leftarrow \{0,1\}^n}[A(1^n, f(x)) = \mathrm{hc}(x)] \leq \frac{1}{2} + negl(n)$$

Attempt 1: Hard-Core Predicate

Consider the predicate

$$\mathrm{hc}(\mathrm{x}) = \oplus_{i=1}^{n} x_i$$

Hope: hc is hard core predicate for any OWF.

Counter-example:

$$\mathrm{f}(\mathrm{x}) = (\mathrm{g}(\mathrm{x}), \oplus_{i=1}^{n} x_i)$$

Trivial Hard-Core Predicate

Consider the function

$$f(x_1,\ldots,x_n) = x_1,\ldots,x_{n-1}$$

f has a trivial hard core predicate

$$\mathrm{hc}(\mathrm{x}) = x_n$$

Not useful for crypto applications (e.g., f is not a OWF)

Attempt 3: Hard-Core Predicate

Consider the predicate

$$\mathrm{hc}(\mathrm{x},\mathrm{r}) = \bigoplus_{i=1}^{n} x_i r_i$$

(the bits r_1,…, r_n will be selected uniformly at random)

Goldreich-Levin Theorem: (Assume OWFs exist) For any OWF f, hc is a hard-core predicate of g(x,r)=(f(x),r).

Using Hard-Core Predicates

Theorem: Given a one-way-permutation f and a hard-core predicate hc we can construct a PRG G with expansion factor $\ell(n) = n + 1$.

Construction:

$$G(s) = f(s) \parallel \mathrm{hc}(s)$$

Intuition: f(s) is actually uniformly distributed

- s is random
- f(s) is a permutation
- Last bit is hard to predict given f(s) (since hc is hard-core for f)

Arbitrary Expansion

Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = n + 1$. Then for any polynomial p(.) there is a PRG with expansion factor p(n).

Construction:

- $G(x) = y||b.$ (n+1 bits)
- $G^1(x) = G(y)||b$ (n+2 bits)
- $G^{i+1}(x) = G(y)||b$ where $G^i(x) = y||b$ (n+2 bits)

Any Beyond

Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = n + 1$. Then for any polynomial p(.) there is a PRG with expansion factor p(n).

Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = 2n$. Then there is a secure PRF.

Theorem: Suppose that there is a secure PRF then there is a strong pseudorandom permutation.

Any Beyond

Corollary: If one-way functions exist then PRGs, PRFs and strong PRPs all exist.

Corollary: If one-way functions exist then there exist CCA-secure encryption schemes and secure MACs.

PRFs from PRGs

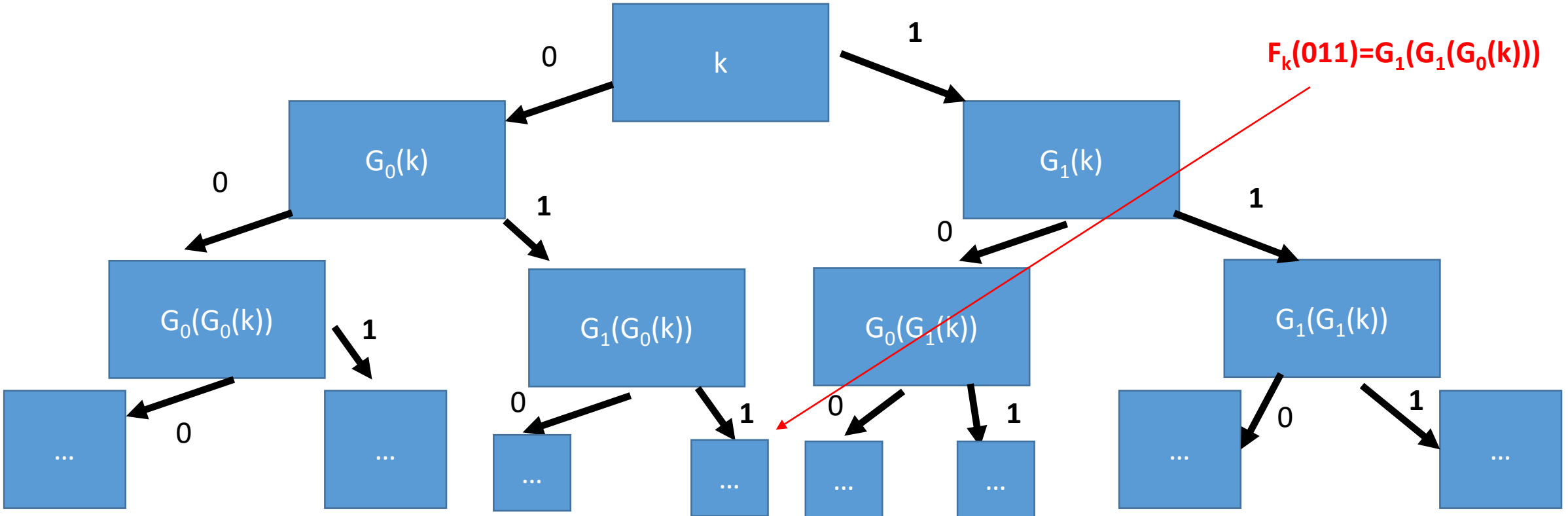
Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = 2n$. Then there is a secure PRF.

Let $G(x) = G_0(x)||G_1(x)$ (first/last n bits of output)

$$F_K(x_1, \ldots, x_n) = G_{x_n}\left(\ldots\left(G_{x_2}\left(G_{x_1}(K)\right)\right)\ldots\right)$$

PRFs from PRGs

Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = 2n$. Then there is a secure PRF.



PRFs from PRGs

Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = 2n$. Then there is a secure PRF.

Proof:

Claim 1: For any t(n) and any PPT attacker A we have

$$\left|\boldsymbol{Pr}\left[\boldsymbol{A}\left(\boldsymbol{r_1} \parallel \cdots \parallel \boldsymbol{r_{t(n)}}\right)\right] - \boldsymbol{Pr}\left[\boldsymbol{A}\left(\boldsymbol{G}(\boldsymbol{s_1}) \parallel \cdots \parallel \boldsymbol{G}\left(\boldsymbol{s_{t(n)}}\right)\right)\right]\right| < \boldsymbol{negl}(\boldsymbol{n})$$

PRFs from PRGs

Claim 1: For any t(n) and any PPT attacker A we have

$$\left|\boldsymbol{Pr}\left[\boldsymbol{A}\left(\boldsymbol{r_1} \parallel \cdots \parallel \boldsymbol{r_{t(n)}}\right)\right] - \boldsymbol{Pr}\left[\boldsymbol{A}\left(\boldsymbol{G}(\boldsymbol{s_1}) \parallel \cdots \parallel \boldsymbol{G}\left(\boldsymbol{s_{t(n)}}\right)\right)\right]\right| < \boldsymbol{negl}(\boldsymbol{n})$$

Proof by Hybrids: Fix j

$$\boldsymbol{Adv_j} = \left|\boldsymbol{Pr}\left[\boldsymbol{A}\left(\boldsymbol{r_1} \parallel \cdots \parallel \boldsymbol{r_{j+1}} \parallel \boldsymbol{G}\left(\boldsymbol{s_{j+2}}\right) \ldots \parallel \boldsymbol{G}\left(\boldsymbol{s_{t(n)}}\right)\right)\right]$$

PRFs from PRGs

Claim 1: For any t(n) and any PPT attacker A we have

$$\left|Pr\left[A\left(r_1 \| \cdots \| r_{t(n)}\right)\right] - Pr\left[A\left(G(s_1) \| \cdots \| G\left(s_{t(n)}\right)\right)\right]\right| < negl(n)$$

Proof

$$\left|Pr\left[A\left(r_1 \| \cdots \| r_{t(n)}\right)\right] - Pr\left[A\left(G(s_1) \| \cdots \| G\left(s_{t(n)}\right)\right)\right]\right|$$

$$\leq \sum_{j<t(n)} Adv_j$$

$$\leq t(n) \times negl(n) = negl(n)$$

PRFs from PRGs

Claim 1: For any t(n) and any PPT attacker A we have

$$\left|Pr\left[A\left(r_1 \| \cdots \| r_{t(n)}\right)\right] - Pr\left[A\left(G(s_1) \| \cdots \| G\left(s_{t(n)}\right)\right)\right]\right| < negl(n)$$

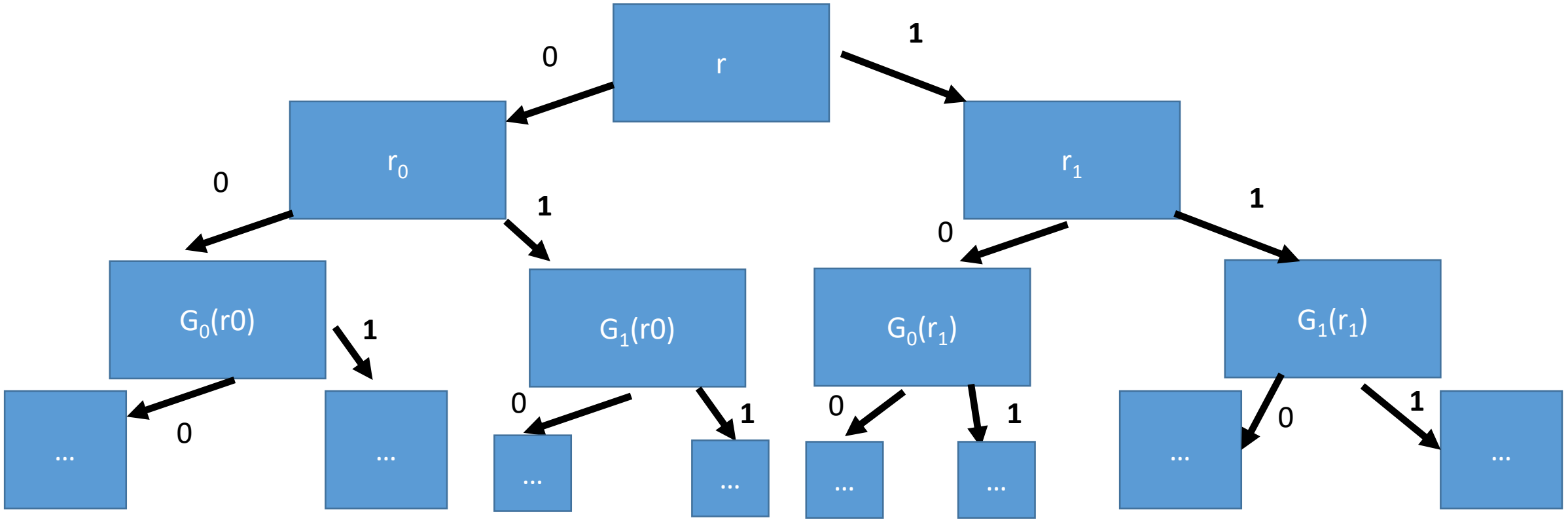
Proof

$$\left|Pr\left[A\left(r_1 \| \cdots \| r_{t(n)}\right)\right] - Pr\left[A\left(G(s_1) \| \cdots \| G\left(s_{t(n)}\right)\right)\right]\right|$$

$$\leq \sum_{j<t(n)} Adv_j$$

$$\leq t(n) \times negl(n) = negl(n)$$

Hybrid H_1



From OWFs (Recap)

Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = n + 1$. Then for any polynomial p(.) there is a PRG with expansion factor p(n).

Theorem: Suppose that there is a PRG G with expansion factor $\ell(n) = 2n$. Then there is a secure PRF.

Theorem: Suppose that there is a secure PRF then there is a strong pseudorandom permutation.

From OWFs (Recap)

Corollary: If one-way functions exist then PRGs, PRFs and strong PRPs all exist.

Corollary: If one-way functions exist then there exist CCA-secure encryption schemes and secure MACs.

Are OWFs Necessary for Private Key Crypto

- Previous results show that OWFs are sufficient.
- Can we build Private Key Crypto from weaker assumptions?
- **Short Answer:** No, OWFs are also *necessary* for most private-key crypto primitives

PRGs → OWFs

Proposition 7.28: If PRGs exist then so do OWFs.

Proof: Let G be a secure PRG with expansion factor $\ell(n) = 2n$.

Question: why can we assume that we have an PRG with expansion 2n?

Answer: Last class we showed that a PRG with expansion factor $\ell(n) = n + 1$. Implies the existence of a PRG with expansion p(n) for any polynomial.

PRGs → OWFs

Proposition 7.28: If PRGs exist then so do OWFs.

Proof: Let G be a secure PRG with expansion factor $\ell(n) = 2n$.

Claim: G is also a OWF!

(Easy to Compute?) ✓

(Hard to Invert?)

Intuition: If we can invert G(x) then we can distinguish G(x) from a random string.

PRGs → OWFs

Proposition 7.28: If PRGs exist then so do OWFs.

Proof: Let G be a secure PRG with expansion factor $\ell(n) = 2n$.

Claim 1: Any PPT A, given G(s), cannot find s except with negligible probability.

Reduction: Assume (for contradiction) that A can invert G(s) with non-negligible probability p(n).

Distinguisher D(y): Simulate A(y)

Output 1 if and only if A(y) outputs x s.t. G(x)=y.

PRGs → OWFs

Proposition 7.28: If PRGs exist then so do OWFs.

Proof: Let G be a secure PRG with expansion factor $\ell(n) = 2n$.

Claim 1: Any PPT A, given G(s), cannot find s except with negligible probability.

Intuition for Reduction: If we can find x s.t. G(x)=y then y is not random.

Fact: Select a random 2n bit string y. Then (whp) there does not exist x such that G(x)=y.

Why not?

PRGs → OWFs

Proposition 7.28: If PRGs exist then so do OWFs.

Proof: Let G be a secure PRG with expansion factor $\ell(n) = 2n$.

Claim 1: Any PPT A, given G(s), cannot find s except with negligible probability.

Intuition: If we can invert G(x) then we can distinguish G(x) from a random string.

Fact: Select a random 2n bit string y. Then (whp) there does not exist x such that G(x)=y.

- Why not? Simple counting argument, 2^{2n} possible y's and 2^n x's.
- Probability there exists such an x is at most 2^{-n} (for a random y)

What other assumptions imply OWFs?

- PRGs → OWFs
- (Easy Extension) PRFs → PRGs → OWFs

- Does secure crypto scheme imply OWFs?
  - CCA-secure? (Strongest)
  - CPA-Secure? (Weaker)
  - EAV-secure? (Weakest)
    - As long as the plaintext is longer than the secret key
  - Perfect Secrecy? X (Guarantee is information theoretic)

EAV-Secure Crypto → OWFs

Proposition 7.29: If there exists a EAV-secure private-key encryption scheme that encrypts messages twice as long as its key, then a one-way function exists.

Recap: EAV-secure.

- Attacker picks two plaintexts m_0, m_1 and is given $c=Enc_K(m_b)$ for random bit b.
- Attacker attempts to guess b.
- No ability to request additional encryptions (chosen-plaintext attacks)
- In fact, no ability to observe any additional encryptions

EAV-Secure Crypto → OWFs

Proposition 7.29: If there exists a EAV-secure private-key encryption scheme that encrypts messages twice as long as its key, then a one-way function exists.

Reduction: $\boldsymbol{f(m,k,r) = Enc_k(m;r)\|m}$.

Input: 4n bits

(For simplicity assume that $\mathbf{Enc}_k$ accepts n bits of randomness)

Claim: f is a OWF

EAV-Secure Crypto → OWFs

Proposition 7.29: If there exists a EAV-secure private-key encryption scheme that encrypts messages twice as long as its key, then a one-way function exists.

Reduction: $\boldsymbol{f(m,k,r) = Enc_k(m;r)\|m}$.

Claim: f is a OWF

Reduction: If attacker A can invert f, then attacker A' can break EAV-security as follows. Given $c=Enc_k(m_b;r)$ run $A(c\|m_0)$. If A outputs (m',k',r') such that $f(m',k',r') = c\|m_0$ then output 0; otherwise 1;

MACs→ OWFs

In particular, given a MAC that satisfies MAC security (Definition 4.2) against an attacker who sees an arbitrary (polynomial) number of message/tag pairs.

Conclusions: OWFs are necessary and sufficient for all (non-trivial) private key cryptography.

→OWFs are a minimal assumption for private-key crypto.

Public Key Crypto/Hashing?

- OWFs are known to be necessary
- Not known (or believed) to be sufficient.

Computational Indistinguishability

- Consider two distributions X_ℓ and Y_ℓ (e.g., over strings of length ℓ).
- Let D be a distinguisher that attempts to guess whether a string s came from distribution X_ℓ or Y_ℓ.

The advantage of a distinguisher D is

$$Adv_{D,\ell} = \left| Pr_{s \leftarrow X_\ell}[D(s) = 1] - Pr_{s \leftarrow Y_\ell}[D(s) = 1] \right|$$

Definition: We say that an ensemble of distributions $\{X_n\}_{n \in \mathbb{N}}$ and $\{Y_n\}_{n \in \mathbb{N}}$ are computationally indistinguishable if for all PPT distinguishers D, there is a negligible function negl(n), such that we have

$$Adv_{D,n} \leq negl(n)$$

Computational Indistinguishability

The advantage of a distinguisher D is

$$Adv_{D,\ell} = \left| Pr_{s \leftarrow \mathrm{X}_\ell}[D(s) = 1] - Pr_{s \leftarrow \mathrm{Y}_\ell}[D(s) = 1] \right|$$

- Looks similar to definition of PRGs
  - X_n is distribution $G(U_n)$ and
  - Y_n is uniform distribution $U_{\ell(n)}$ over strings of length $\ell(n)$.

Computational Indistinguishability

Definition: We say that an ensemble of distributions $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable if for all PPT distinguishers D, there is a negligible function negl(n), such that we have

$$Adv_{D,n} \leq negl(n)$$

Theorem 7.32: Let t(n) be a polynomial and let $P_n = X_n^{t(n)}$ and $Q_n = Y_n^{t(n)}$ then the ensembles $\{P_n\}_{n\in\mathbb{N}}$ and $\{Q_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable

Computational Indistinguishability

Definition: We say that an ensemble of distributions $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable if for all PPT distinguishers D, there is a negligible function negl(n), such that we have

$$Adv_{D,n} \leq negl(n)$$

Fact: Let $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ be computationally indistinguishable and let $\{Z_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ be computationally indistinguishable

Then

$\{X_n\}_{n\in\mathbb{N}}$ and $\{Z_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable