One-Way Functions

CS 276: Introduction to Cryptography

Sanjam Garg

January 29, 2026

Overview

Hard Concentrated Bit: The Question

Question

Is it possible to concentrate the strength of a one-way function into one bit? In particular, given a one-way function f, does there exist one bit that can be computed efficiently from the input x, but is hard to compute given $f(x)$?

Hard Concentrated Bit: The Question

Question

Is it possible to concentrate the strength of a one-way function into one bit? In particular, given a one-way function f, does there exist one bit that can be computed efficiently from the input x, but is hard to compute given $f(x)$?

Why This Matters

- Hard bits are useful for constructing pseudorandom generators
- Needed for other cryptographic primitives
- The answer is subtle - not every OWF has a hard bit, but we can always construct one that does

Definition

Definition 1 (Hard Concentrated Bit)

For one-way function $f : \{0,1\}^n \to \{0,1\}^n$, a function $B : \{0,1\}^n \to \{0,1\}$ is a **hard concentrated bit** if:

- B is computable in polynomial time
- $\forall$ non-uniform PPT $\mathcal{A}$:

$$\Pr_x[\mathcal{A}(1^n, f(x)) = B(x)] \leq \frac{1}{2} + \mathrm{negl}(n)$$

Definition

Definition 1 (Hard Concentrated Bit)

For one-way function $f : \{0,1\}^n \to \{0,1\}^n$, a function $B : \{0,1\}^n \to \{0,1\}$ is a **hard concentrated bit** if:

- B is computable in polynomial time
- $\forall$ non-uniform PPT $\mathcal{A}$:

$$\Pr_x[\mathcal{A}(1^n, f(x)) = B(x)] \leq \frac{1}{2} + \mathsf{negl}(n)$$

Intuition

Even given $f(x)$, adversary cannot predict $B(x)$ better than random guessing.

Simple Example: Information-Theoretic

Construction

Given one-way function f, define:

$$g(b, x) = 0 \| f(x)$$

$$B(b, x) = b$$

Simple Example: Information-Theoretic

Construction

Given one-way function f, define:

$$g(b, x) = 0 \| f(x)$$

$$B(b, x) = b$$

Why This Works

- $g(b, x)$ reveals **nothing** about b (information-theoretically)
- Adversary cannot predict b better than $1/2$
- But this is **information-theoretic** security

Simple Example: Information-Theoretic

Construction

Given one-way function f, define:

$$g(b, x) = 0 \| f(x)$$

$$B(b, x) = b$$

Why This Works

- $g(b, x)$ reveals **nothing** about b (information-theoretically)
- Adversary cannot predict b better than $1/2$
- But this is **information-theoretic** security

What We Want

Computational hardness - adversary could theoretically compute it but it's computationally infeasible. The Goldreich-Levin theorem shows we can achieve this.

Open Question

Unknown

Does every one-way function have a hard concentrated bit? **Unknown!**

Open Question

Unknown

Does every one-way function have a hard concentrated bit? **Unknown!**

What We Know

- We can always **construct** a one-way function with a hard bit
- But we may not be able to find a hard bit for an **arbitrary** one-way function
- This remains an open problem in cryptography

Hard Concentrated Bit for OWPs

Theorem 2

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function. Define:

$$g(x, r) = f(x) \| r$$

where $|x| = |r| = n$. Then g is one-way and has hard concentrated bit:

$$B(x, r) = \sum_{i=1}^{n} x_i r_i \bmod 2$$

Hard Concentrated Bit for OWPs

Theorem 2

Let $f : \{0,1\}^n \to \{0,1\}^n$ *be a one-way function. Define:*

$$g(x, r) = f(x) \| r$$

where $|x| = |r| = n$*. Then* g *is one-way and has hard concentrated bit:*

$$B(x, r) = \sum_{i=1}^{n} x_i r_i \bmod 2$$

Key Insight

- $B(x, r)$ is the **inner product** modulo 2
- Even given $f(x)$ and r, hard to predict $B(x, r)$
- Proof uses **Goldreich-Levin** technique

Proof Strategy: Three Cases

We present the proof in three steps, each progressively more complex:

1. **Super Simple**: Adversary predicts perfectly ($\Pr = 1$)
2. **Simple**: Adversary predicts with $\Pr \geq 3/4 + \epsilon$
3. **Real Case**: Adversary predicts with $\Pr \geq 1/2 + \epsilon$

Case 1: Super Simple

Setup

Suppose adversary $\mathcal{A}$ guesses $B(\cdot)$ with perfect accuracy:

$$\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)] = 1$$

Case 1: Super Simple

Setup

Suppose adversary $\mathcal{A}$ guesses $B(\cdot)$ with perfect accuracy:

$$\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)] = 1$$

Construction of $\mathcal{B}$

Let e^i denote the one-hot n-bit string (only i-th bit is 1).

$\mathcal{B}$ on input $f(x)\|r$:

- For $i = 1$ to n:
  - $x_i' \leftarrow \mathcal{A}(1^{2n}, f(x)\|e^i)$
- Output $x_1' \cdots x_n'\|r$

Case 1: Why This Works

Key Observation

Observe that $B(x, e^i) = \sum_{j=1}^{n} x_j e^i_j = x_i$.

Case 1: Why This Works

Key Observation

Observe that $B(x, e^i) = \sum_{j=1}^{n} x_j e^i_j = x_i$.

Success Probability

Since $\mathcal{A}$ predicts perfectly:

$$\Pr_x \left[\mathcal{A}(1^{2n}, f(x) \| e^i) = x_i \right] = \Pr_x \left[\mathcal{A}(1^{2n}, f(x) \| e^i) = B(x, e^i) \right] = 1$$

Case 1: Why This Works

Key Observation

Observe that $B(x, e^i) = \sum_{j=1}^{n} x_j e^i_j = x_i$.

Success Probability

Since $\mathcal{A}$ predicts perfectly:

$$\Pr_x\left[\mathcal{A}(1^{2n}, f(x)\|e^i) = x_i\right] = \Pr_x\left[\mathcal{A}(1^{2n}, f(x)\|e^i) = B(x, e^i)\right] = 1$$

Conclusion

Hence $\Pr_{x,r}[\mathcal{B}(1^{2n}, g(x, r)) = (x, r)] = 1$.
$\mathcal{B}$ inverts g with perfect accuracy!

Case 2: Simple Case

Setup

$$\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x, r)) = B(x, r)] \geq \frac{3}{4} + \epsilon(n)$$

where $\epsilon(\cdot)$ is non-negligible.

Case 2: Simple Case

Setup

$$\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)] \geq \frac{3}{4} + \epsilon(n)$$

where $\epsilon(\cdot)$ is non-negligible.

Key Technique

We can no longer use the super simple algorithm. Instead:

- Use two calls: $B(x,s) \oplus B(x, s \oplus e^i) = x_i$
- Take majority over many trials
- Use Chebyshev's inequality

Case 2: Key Identity

Why $B(x,s) \oplus B(x, s \oplus e^i) = x_i$?

$$\begin{aligned} B(x,s) \oplus B(x, s \oplus e^i) &= \sum_j x_j s_j \oplus \sum_j x_j (s_j \oplus e^i_j) \\ &= \sum_{j \neq i} (x_j s_j \oplus x_j s_j) \oplus x_i s_i \oplus x_i (s_i \oplus 1) \\ &= x_i \end{aligned}$$

Case 2: Key Identity

Why $B(x, s) \oplus B(x, s \oplus e^i) = x_i$?

$$\begin{aligned} B(x, s) \oplus B(x, s \oplus e^i) &= \sum_j x_j s_j \oplus \sum_j x_j (s_j \oplus e^i_j) \\ &= \sum_{j \neq i} (x_j s_j \oplus x_j s_j) \oplus x_i s_i \oplus x_i (s_i \oplus 1) \\ &= x_i \end{aligned}$$

Intuition

- XORing with e^i flips only the i-th bit of s
- This isolates x_i in the inner product
- All other terms cancel out

Case 2: Algorithm

Algorithm $\mathcal{B}$ on input $f(x)\|r$:

1. For $i = 1$ to n:
   - For $t = 1$ to $T = \frac{n}{2\epsilon(n)^2}$:
     - Sample $s \leftarrow \{0,1\}^n$ uniformly
     - $x_i^t \leftarrow \mathcal{A}(f(x)\|s) \oplus \mathcal{A}(f(x)\|(s \oplus e^i))$
   - $x_i' \leftarrow$ majority of $\{x_i^1, \ldots, x_i^T\}$
2. Output $x_1' \cdots x_n' \| r$

Case 2: Proof - Good x's

Define Event E

Let E denote the event that $\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)$.

Case 2: Proof - Good x's

Define Event E

Let E denote the event that $\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)$.

Define Good Set

$$G := \left\{ x \mid \Pr_r[E] \geq \frac{3}{4} + \frac{\epsilon(n)}{2} \right\}$$

Case 2: Proof - Good x's

Define Event E

Let E denote the event that $\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)$.

Define Good Set

$$G := \left\{ x \mid \Pr_r [E] \geq \frac{3}{4} + \frac{\epsilon(n)}{2} \right\}$$

Claim: G is Large

$$\Pr_{x \leftarrow \{0,1\}^n} [x \in G] \geq \frac{\epsilon(n)}{2}$$

Case 2: Proof - G is Large

Proof by Contradiction

Assume $\Pr[x \in G] < \frac{\epsilon(n)}{2}$. Then:

$$\begin{aligned}
\frac{3}{4} + \epsilon(n) &\le \Pr_{x,r}[E] \\
&= \Pr_x[x \in G] \Pr_r[E \mid x \in G] + \Pr_x[x \notin G] \Pr_r[E \mid x \notin G] \\
&< \frac{\epsilon(n)}{2} \cdot 1 + 1 \cdot \left(\frac{3}{4} + \frac{\epsilon(n)}{2} \right) \\
&= \frac{3}{4} + \epsilon(n)
\end{aligned}$$

Case 2: Proof - G is Large

Proof by Contradiction

Assume $\Pr[x \in G] < \frac{\epsilon(n)}{2}$. Then:

$$\begin{aligned}
\frac{3}{4} + \epsilon(n) &\leq \Pr_{x,r}[E] \\
&= \Pr_x[x \in G] \Pr_r[E \mid x \in G] + \Pr_x[x \notin G] \Pr_r[E \mid x \notin G] \\
&< \frac{\epsilon(n)}{2} \cdot 1 + 1 \cdot \left(\frac{3}{4} + \frac{\epsilon(n)}{2} \right) \\
&= \frac{3}{4} + \epsilon(n)
\end{aligned}$$

Contradiction!

This is a contradiction, so we must have:

$$\Pr[x \in G] \geq \frac{\epsilon(n)}{2}$$

Case 2: Proof - Success Probability (Part 1)

For Fixed $x \in G$

$$
\begin{aligned}
&\Pr_s\left[\mathcal{A}(f(x), s) \oplus \mathcal{A}(f(x), s \oplus e^i) = x_i\right] \\
&= \Pr_s[\text{Both correct}] + \Pr_s[\text{Both wrong}] \\
&\geq \Pr_s[\text{Both correct}] = 1 - \Pr_s[\text{Either wrong}] \\
&\geq 1 - 2 \cdot \Pr_s[\mathcal{A} \text{ is wrong}] \\
&\geq 1 - 2\left(\frac{1}{4} - \frac{\epsilon(n)}{2}\right) = \frac{1}{2} + \epsilon(n)
\end{aligned}
$$

Case 2: Proof - Success Probability (Part 1)

For Fixed $x \in G$

$$
\begin{aligned}
&\Pr_s\left[\mathcal{A}(f(x), s) \oplus \mathcal{A}(f(x), s \oplus e^i) = x_i\right] \\
&= \Pr_s[\text{Both correct}] + \Pr_s[\text{Both wrong}] \\
&\geq \Pr_s[\text{Both correct}] = 1 - \Pr_s[\text{Either wrong}] \\
&\geq 1 - 2 \cdot \Pr_s[\mathcal{A} \text{ is wrong}] \\
&\geq 1 - 2\left(\frac{1}{4} - \frac{\epsilon(n)}{2}\right) = \frac{1}{2} + \epsilon(n)
\end{aligned}
$$

Key Point

For $x \in G$, each trial succeeds with probability $\geq \frac{1}{2} + \epsilon(n)$.

Case 2: Proof - Success Probability (Part 2)

Chebyshev's Inequality

Let $X_1, \ldots, X_m$ be i.i.d. random variables with $\Pr[X_i = 1] = p$. Then:

$$\Pr\left[\left|\sum_{i=1}^{m} X_i - pm\right| \geq \delta m\right] \leq \frac{1}{4\delta^2 m}$$

Case 2: Proof - Success Probability (Part 2)

Chebyshev's Inequality

Let $X_1, \ldots, X_m$ be i.i.d. random variables with $\Pr[X_i = 1] = p$. Then:

$$\Pr\left[\left|\sum_{i=1}^{m} X_i - pm\right| \geq \delta m\right] \leq \frac{1}{4\delta^2 m}$$

Applying Chebyshev's Inequality

Let Y_i^t be indicator that $x_i^t = x_i$. Trials are independent with $\Pr[Y_i^t = 1] \geq \frac{1}{2} + \epsilon(n)$.

$$\Pr[x_i' \neq x_i] = \Pr\left[\sum_{t=1}^{T} Y_i^t \leq \frac{T}{2}\right]$$

$$= \Pr\left[\left|\sum_{t=1}^{T} Y_i^t - \left(\tfrac{1}{2} + \epsilon(n)\right) T\right| \geq \epsilon(n) T\right]$$

Case 2: Proof - Success Probability (Part 2 cont.)

Applying Chebyshev (bound)

With $\delta = \epsilon(n)$ and $m = T$ in Chebyshev:

$$\Pr[x_i' \neq x_i] \leq \frac{1}{4\epsilon(n)^2 T} = \frac{1}{2n}$$

Case 2: Proof - Final Success Probability

Final Calculation

$$
\begin{aligned}
&\Pr_{x,r}[\mathcal{B}(1^{2n}, g(x,r)) = (x,r)] \\
&\geq \Pr_{x}[x \in G] \cdot \Pr[\text{all bits correct} \mid x \in G] \\
&\geq \frac{\epsilon(n)}{2} \cdot \left(1 - \sum_{i=1}^{n} \Pr[x_i' \neq x_i \mid x \in G]\right) \\
&\geq \frac{\epsilon(n)}{2} \cdot \left(1 - n \cdot \frac{1}{2n}\right) = \frac{\epsilon(n)}{4}
\end{aligned}
$$

Case 2: Proof - Final Success Probability

Final Calculation

$$\begin{aligned}
&\Pr_{x,r}[\mathcal{B}(1^{2n}, g(x,r)) = (x,r)] \\
&\geq \Pr_x[x \in G] \cdot \Pr[\text{all bits correct} \mid x \in G] \\
&\geq \frac{\epsilon(n)}{2} \cdot \left(1 - \sum_{i=1}^{n} \Pr[x_i' \neq x_i \mid x \in G]\right) \\
&\geq \frac{\epsilon(n)}{2} \cdot \left(1 - n \cdot \frac{1}{2n}\right) = \frac{\epsilon(n)}{4}
\end{aligned}$$

Conclusion

Since $\epsilon(n)$ is non-negligible, $\frac{\epsilon(n)}{4}$ is also non-negligible. This contradicts the one-wayness of g!

Case 3: Real Case

Setup

$$\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)] \geq \frac{1}{2} + \epsilon(n)$$

where $\epsilon(\cdot)$ is non-negligible.

Case 3: Real Case

Setup

$$\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)] \geq \frac{1}{2} + \epsilon(n)$$

where $\epsilon(\cdot)$ is non-negligible.

Key Challenge

We cannot make two related calls to $\mathcal{A}$ as in the simple case since we can't argue that both calls will be correct with high enough probability.

Case 3: Real Case

Setup

$$\Pr_{x,r}[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r)] \geq \frac{1}{2} + \epsilon(n)$$

where $\epsilon(\cdot)$ is non-negligible.

Key Challenge

We cannot make two related calls to $\mathcal{A}$ as in the simple case since we can't argue that both calls will be correct with high enough probability.

The "Magic"

The key idea is to guess $\log T$ values. With probability $1/T$, all guesses are correct. Then by using XOR combinations, we get T pairwise independent samples.

Case 3: Algorithm

Algorithm $\mathcal{B}$ on input $f(x)\|r$:

1. $T = \frac{2n}{\epsilon(n)^2}$
2. For $\ell = 1$ to $\log T$:
   - Sample $s_\ell \leftarrow \{0,1\}^n$ uniformly
   - Sample $b_\ell \leftarrow \{0,1\}$ uniformly
3. For $i = 1$ to n:
   - For all $L \subseteq \{1, 2, \ldots, \log T\}$:
     - $S_L := \bigoplus_{j \in L} s_j$
     - $B_L := \bigoplus_{j \in L} b_j$
     - $x_i^L \leftarrow B_L \oplus \mathcal{A}(f(x)\|S_L \oplus e^i)$
   - $x_i' \leftarrow$ majority of $\{x_i^{\emptyset}, \ldots, x_i^{[\log T]}\}$
4. Output $x_1' \cdots x_n' \| r$

Case 3: Key Property

Event F: All Guesses Correct

With probability $\frac{1}{T}$, we have $b_\ell = B(x, s_\ell)$ for all ℓ.

Case 3: If F Happens, Then $B_L = B(x, S_L)$

Derivation

$$\begin{aligned}
B(x, S_L) &= \sum_{k=1}^{n} x_k \left(\bigoplus_{j \in L} s_j \right)_k \\
&= \sum_{k=1}^{n} x_k \sum_{j \in L} (s_j)_k \\
&= \sum_{j \in L} \sum_{k=1}^{n} x_k (s_j)_k \\
&= \sum_{j \in L} B(x, s_j) \\
&= \sum_{j \in L} b_j = B_L
\end{aligned}$$

Case 3: Proof - Good x's

Define Good Set

$$G := \left\{ x \mid \Pr_r \left[\mathcal{A}(1^{2n}, g(x, r)) = B(x, r) \right] \geq \frac{1}{2} + \frac{\epsilon(n)}{2} \right\}$$

Case 3: Proof - Good x's

Define Good Set

$$G := \left\{ x \mid \Pr_r \left[\mathcal{A}(1^{2n}, g(x,r)) = B(x,r) \right] \geq \frac{1}{2} + \frac{\epsilon(n)}{2} \right\}$$

Claim: G is Large

$$\Pr_{x \leftarrow \{0,1\}^n} [x \in G] \geq \frac{\epsilon(n)}{2}$$

Case 3: Proof - Good x's

Define Good Set

$$G := \left\{ x \mid \Pr_r \left[\mathcal{A}(1^{2n}, g(x, r)) = B(x, r) \right] \geq \frac{1}{2} + \frac{\epsilon(n)}{2} \right\}$$

Claim: G is Large

$$\Pr_{x \leftarrow \{0,1\}^n} [x \in G] \geq \frac{\epsilon(n)}{2}$$

Proof

Same argument as in Case 2 (by contradiction).

Case 3: Proof - Pairwise Independence

Key Observation

Given $\{b_\ell = B(x, s_\ell)\}_\ell$ and $x \in G$:

$$
\begin{aligned}
&\Pr_r\left[B_L \oplus \mathcal{A}(f(x) \| S_L \oplus e^i) = x_i\right] \\
&= \Pr_r\left[B(x, S_L) \oplus \mathcal{A}(f(x) \| S_L \oplus e^i) = x_i\right] \\
&= \Pr_r\left[\mathcal{A}(f(x) \| S_L \oplus e^i) = B(x, S_L \oplus e^i)\right] \\
&\geq \frac{1}{2} + \frac{\epsilon(n)}{2}
\end{aligned}
$$

Case 3: Proof - Pairwise Independence

Key Observation

Given $\{b_\ell = B(x, s_\ell)\}_\ell$ and $x \in G$:

$$\begin{aligned}
&\Pr_r \left[B_L \oplus \mathcal{A}(f(x) \| S_L \oplus e^i) = x_i\right] \\
&= \Pr_r \left[B(x, S_L) \oplus \mathcal{A}(f(x) \| S_L \oplus e^i) = x_i\right] \\
&= \Pr_r \left[\mathcal{A}(f(x) \| S_L \oplus e^i) = B(x, S_L \oplus e^i)\right] \\
&\geq \frac{1}{2} + \frac{\epsilon(n)}{2}
\end{aligned}$$

Pairwise Independence

The events Y_i^L (that $x_i^L = x_i$) for different L are **pairwise independent**.

Case 3: Proof - Chebyshev Bound

By Chebyshev's Inequality

$$\Pr[x_i' \neq x_i] \leq \frac{1}{4(\epsilon(n)/2)^2 T} = \frac{1}{2n}$$

Case 3: Final Success Probability

Completing the Proof

$$
\begin{aligned}
&\Pr_{x,r}[\mathcal{B}(1^{2n}, g(x,r)) = (x,r)] \\
&\geq \Pr[F] \cdot \Pr_x[x \in G] \cdot \Pr[\text{all bits correct} \mid x \in G \wedge F] \\
&\geq \frac{1}{T} \cdot \frac{\epsilon(n)}{2} \cdot \left(1 - \sum_{i=1}^{n} \Pr[x_i' \neq x_i \mid x \in G \wedge F]\right) \\
&\geq \frac{\epsilon(n)^2}{2n} \cdot \frac{\epsilon(n)}{2} \cdot \left(1 - n \cdot \frac{1}{2n}\right) \\
&= \frac{\epsilon(n)^3}{8n}
\end{aligned}
$$

Case 3: Conclusion

Conclusion

Since $\epsilon(n)$ is non-negligible, $\frac{\epsilon(n)^3}{8n}$ is also non-negligible. This contradicts the one-wayness of g, completing the proof.

Key Takeaways

1. **One-way functions** are the weakest cryptographic primitive
   - Easy to compute, hard to invert
   - Existence implies $P \neq NP$
2. **Robustness**: Can modify OWFs in many ways (fix values, etc.)
3. **Brittleness**: Some operations break one-wayness
   - Composition doesn't always work
   - Dropping bits can break security
4. **Hardness Amplification**: Weak OWFs $\Rightarrow$ Strong OWFs
5. **Levin's OWF**: Explicit construction, future-proof
6. **Hard Bits**: Can extract hard bits from OWFs (for OWPs)

Next Steps

- One-way functions are the foundation
- Next: Building more powerful primitives from OWFs
  - Pseudorandom generators
  - Pseudorandom functions
  - Encryption schemes
- Questions?