Today

- Weak vs. Strong One-Way functions
- Universal One-Way functions
- Collections of One-Way Functions
- In search of an example (I)
- Trapdoor Function
- Collection of Trapdoor Functions
- In search of an example (II)

Weak One Way Functions

- **Weak One Way Function:**
  \[ \exists \text{ polynomial } Q \text{ s.t.
  } \forall \text{ PPT algorithms } A, \forall n \text{ sufficiently large,}
  \Pr( A(f(x)) \neq x' \text{ such that } f(x)=f(x')) \geq 1/Q(n) \]
  probability is taken over x s.t. |x|=n and A's coins

- **Intuitively:** there is some hard-core polynomial fraction set of instances on which it is always hard to invert f

Weak OWF = Strong OWF

- **Theorem:** one-way functions exist iff weak one-way functions exist
- **Proof:** in class
Problems used in Cryptography tend to be

- Not known to be NP-hard P nor in BPP, for example:
  - Factoring Integers
  - Graph Isomorphism
  - Decoding of Random Linear Codes
  - Approximating size of the shortest vector in a lattice to within sqrt(n) of the optimal for dimension n

What to do till can prove a particular function is OWF

- **Universal One-Way Function Theorem:**
  There exists (and we can construct it) a polynomial computable function which is one-way if and only if any one-way function exists.

  - **Proof:** Goldreich's book, section 2.4.1
  - **Idea:** Show OWF $\Rightarrow \exists$ OWF which can evaluate in quadratic time by some algorithm.

    - Set $f_{universal}(\text{desc}(M),x) = (\text{desc}(M), M(x))$ where
      - $\text{desc}(M) =$ description of algorithm $M$
      - $M(x) =$ output of $M$ on $x$ if $M$ on $x$ runs in $O(|x|^2)$ and $x$ otherwise.

Collections of One Way Functions

**Definition:** $F = \{f_i:D_i\rightarrow R_i\}_{i \in I}$ where $I$ is a set of indices, and $D_i$, $R_i$ are finite sets.

- **Easy to Evaluate:** $\exists$ PPT algorithm $A$ s.t.
  $$A(i,x) = f_i(x)$$ for all $i,x$

- **Hard to Invert:** $\forall$ PPT $B$, $\forall$ sufficiently large $k$,
  $\text{prob} ( B(i,f_i(x)) = x \text{ s.t } f_i(x) = f_i(x')) < \text{neg}(k)$
  $|i|=n$, $i$ in $I$, $x$ in $D_i$, coins of $B$

- **Generation:** $\exists$ PPT (in $n$) algorithm $G$ that selects a random $f_i$ in $F$ such that $|i|=n$.
  $G$ takes as input $n$, the security parameter.