One-Way Functions as Collections: A collection of functions consists of an infinite set of indices, denoted $\bar{I}$, and a corresponding set of finite functions, denoted $\{f_i\}_{i \in \bar{I}}$. That is, for each $i \in \bar{I}$, the domain of the function f_i, denoted D_i, is a finite set.

A collection of functions $\{f_i : D_i \longrightarrow \{0,1\}^*\}_{i \in \bar{I}}$ is called strongly one-way if there exist three probabilistic polynomial-time algorithms I, D, and F such that the following two conditions hold:

(1) Easy to sample and compute: The output distribution of algorithm I on input 1^n is a random variable assigned values in the set $\bar{I} \cap \{0,1\}^n$. The output distribution of algorithm D on input $i \in \bar{I}$ is a random variable assigned values in D_i. On input $i \in \bar{I}$ and $x \in D_i$, algorithm F always outputs $f_i(x)$.

(2) Hard to invert: For every probabilistic polynomial-time algorithm A', every positive polynomial $p(\,)$, and all sufficiently large n's,

$$\Pr[A'(I_n, f_{I_n}(X_n)) \in f_{I_n}^{-1}(f_{I_n}(X_n))] < \frac{1}{p(n)},$$

where I_n is a random variable describing the output distribution of algorithm I on input 1^n, and X_n is a random variable describing the output of algorithm D on input random variable I_n.

Examples of one-way functions:

① Integer Factorization: Let $f_{mult}(x, y) = x \cdot y$ where $|x| = |y|$, and $x \cdot y$ denotes the string representing the integer resulting by multiplying the integers represented by the strings x and y. We can compute f_{mult} in polynomial time. Assuming the intractability of factoring that given the product of two uniformly chosen ~~n~~ n-bit-long primes, it's infeasible to find the prime factors, we can show that f_{mult} is strongly one way.

② The Subset-Sum Problem: Let $f_{ssum}(x_1, \dots x_n, I) = (x_1, \dots, x_n, \sum_{i \in I} x_i)$ where $|x_1| = \dots = |x_n| = n$, and $I \subseteq \{1, 2, \dots, n\}$ f_{ssum} is polynomial-time-computable. The fact that the subset-sum problem is NP-complete cannot serve as evidence to the one-wayness of f_{ssum}. The conjecture that f_{ssum} is one-way is based on the failure of known algorithms to handle random "high density" instances in which the length of the elements approximately equals their number, as in the definition of f_{ssum}.

Examples of One-Way Collections:

① The RSA Function: The RSA collection of functions has an index set consisting of pairs (N, e) where N is a product of two $(\frac{1}{2} \cdot \log_2 N)$-bit primes, denoted P and Q, and e is an integer smaller than N and relatively prime to $\phi(N) = (P-1)(Q-1)$. The function of index (N, e) has domain $\{1, \ldots, N\}$ and maps the domain element x to $x^e \bmod N$. Using the fact that e is relatively prime to $(P-1)(Q-1)$, we can show that the function is in fact a permutation over its domain. The RSA collection is a collection of permutations: $(e, \phi(N)) = 1$

$\Rightarrow$ there exists an integer d such that $ed \equiv 1 \pmod{\phi(N)}$.

Given $x \in \{1, 2, \ldots N\}$, let $ed = 1 + k(P-1)(Q-1)$ for some integer k. If $x \not\equiv 0 \pmod P$, we have

$$(x^d)^e \equiv x\left(x^{P-1}\right)^{k(Q-1)} \equiv x\left((x \pmod P)^{P-1}\right)^{k(Q-1)} \pmod P$$

$$\equiv x(1)^{k(Q-1)} \pmod P \equiv x \pmod P.$$

Also, $(x^d)^e \equiv x \pmod P$ if $x \equiv 0 \pmod P$

$\Rightarrow (x^d)^e \equiv x \pmod P$ for all x.

Similarly $(x^d)^e \equiv x \pmod Q$ for all x

P and Q are distinct primes $\Rightarrow (x^d)^e \equiv x \pmod N$

$\Rightarrow$ The RSA function is an onto function. Since the domain $\{1, 2, \ldots, N\}$ is finite $\Rightarrow$ RSA is a permutation.