One-Way Functions

One-Way Functions

- A one-way function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ is easy to compute, yet hard to invert.
- Easy to formalize: simply require that f be computable in polynomial time.
- Infeasible for any probabilistic polynomial-time algorithm to invert f—that is, to find a preimage of a given value y

One-Way Functions

- The inverting experiment $\textbf{\textit{Invert}}_{A,f(n)}$

1. Choose uniform $x \in \{0,1\}^n$, and compute $y := f(x)$.
2. A is given 1^n and y as input, and outputs x^0.
3. The output of the experiment is defined to be 1 if $f(x^0) = y$, and 0 otherwise.

DEFINITION 8.1 *A function $f : \{0,1\}^* \to \{0,1\}^*$ is* one-way *if the following two conditions hold:*

1. **(Easy to compute:)** *There exists a polynomial-time algorithm M_f computing f; that is, $M_f(x) = f(x)$ for all x.*
2. **(Hard to invert:)** *For every probabilistic polynomial-time algorithm $\mathcal{A}$, there is a negligible function* negl *such that*

$$\Pr[\mathsf{Invert}_{\mathcal{A},f}(n) = 1] \leq \mathsf{negl}(n).$$

Exponential-time inversion

- Any one-way function can be inverted at any point y in exponential time, by simply trying all values $x \in \{0,1\}^n$ until a value x is found such that $f(x) = y$.
- Thus, the existence of one-way functions is inherently an assumption about **computational complexity and computational hardness**

Hard-core predicate

- A hard-core predicate of a one-way function f is a predicate b (i.e., a function whose output is a single bit) which is easy to compute (as a function of x) but is hard to compute given f(x).
- In formal terms, there is no probabilistic polynomial-time (PPT) algorithm that computes b(x) from f(x) with probability significantly greater than one half over random choice of x.