of the ideal world makes it easy to understand the effect of such an attack. Considering our previous laundry list: the adversary clearly learns no more than $\mathcal{F}(x_1, \ldots, x_n)$ since that is the only message it receives; the outputs given to the honest parties are all consistent and legal; the adversary's choice of inputs is independent of the honest parties'.

Although the ideal world is easy to understand, the presence of a fully-trusted third party makes it imaginary. We use the ideal world as a benchmark against which to judge the security of an actual protocol.

Real World. In the real world, there is no trusted party. Instead, all parties communicate with each other using a protocol. The protocol π specifies for each party P_i a "next-message" function π_i. This function takes as input a security parameter, the party's private input x_i, a random tape, and the list of messages P_i has received so far. Then, π_i outputs either a next message to send along with its destination, or else instructs the party to terminate with some specific output.

In the real world, an adversary can corrupt parties—corruption at the beginning of the protocol is equivalent to the original party being an adversary. Depending on the threat model (discussed next), corrupt parties may either follow the protocol as specified, or deviate arbitrarily in their behavior.

Intuitively speaking, the real world protocol π is considered secure if any effect that an adversary can achieve in the real world can also be achieved by a corresponding adversary in the ideal world. Put differently, the goal of a protocol is to provide security in the real world (given a set of assumptions) that is equivalent to that in the ideal world.

2.3.2 Semi-Honest Security

A *semi-honest* adversary is one who corrupts parties but follows the protocol as specified. In other words, the corrupt parties run the protocol honestly but they may try to learn as much as possible from the messages they receive from other parties. Note that this may involve several colluding corrupt parties pooling their views together in order to learn information. Semi-honest adversaries are also considered *passive* in that they cannot take any actions other than attempting to learn private information by observing a view of a protocol execution. Semi-honest adversaries are also commonly called *honest-but-curious*.

The *view* of a party consists of its private input, its random tape, and the list of all messages received during the protocol. The view of an adversary consists of the combined views of all corrupt parties. Anything an adversary learns from running the protocol must be an efficiently computable function of its view. That is, without loss of generality we need only consider an "attack" in which the adversary simply outputs its entire view.

Following the real-ideal paradigm, security means that such an "attack" can also be carried out in the ideal world. That is, for a protocol to be secure, it must be possible in the ideal world to generate something indistinguishable from the real world adversary's view. Note that the adversary's view in the ideal world consists of nothing but inputs sent to $\mathcal{T}$ and outputs received from $\mathcal{T}$. So, an ideal-world adversary must be able to use this information to generate what looks like a real-world view. We refer to such an ideal-world adversary as a *simulator*, since it generates a "simulated" real-world view while in the ideal-world itself. Showing that such a simulator exists proves that there is nothing an adversary can accomplish in the real world that could not also be done in the ideal world.

More formally, let π be a protocol and $\mathcal{F}$ be a functionality. Let C be the set of parties that are corrupted, and let Sim denote a simulator algorithm. We define the following distributions of random variables:

- $\mathsf{Real}_\pi(\kappa, C; x_1, \ldots, x_n)$: run the protocol with security parameter κ, where each party P_i runs the protocol honestly using private input x_i. Let V_i denote the final view of party P_i, and let y_i denote the final output of party P_i.
  Output $\{V_i \mid i \in C\}, (y_1, \ldots, y_n)$.
- $\mathsf{Ideal}_{\mathcal{F},\mathsf{Sim}}(\kappa, C; x_1, \ldots, x_n)$: Compute $(y_1, \ldots, y_n) \leftarrow \mathcal{F}(x_1, \ldots, x_n)$.
  Output $\mathsf{Sim}(C, \{(x_i, y_i) \mid i \in C\}), (y_1, \ldots, y_n)$.

A protocol is secure against semi-honest adversaries if the corrupted parties in the real world have views that are indistinguishable from their views in the ideal world:

Definition 2.2. A protocol π *securely realizes $\mathcal{F}$ in the presence of semi-honest adversaries* if there exists a simulator Sim such that, for every subset of corrupt parties C and all inputs $x_1, \ldots, x_n$, the distributions

$$\mathsf{Real}_\pi(\kappa, C; x_1, \ldots, x_n)$$