Fair Multi-Party Computation - Take Two

Juan A. Garay (Bell Labs - Lucent Technologies)

Oct 5 2005

This is joint work with Phil MacKenzie, Manoj Prabhakaran and Ke Yang.

Secure multi-party computation (MPC) has been one of the most fundamental problems in cryptography. At a high level, the problem is concerned with n parties, each holding a private input, that want to compute a function of those inputs so that each party learns its own output, but no other information is revealed, even in the presence of malicious parties that may deviate arbitrarily from the protocol. Instances of MPC include password authentication, distributed auctions, contract signing, on-line bidding, etc.

MPC is called "fair" if either all the parties learn the outuput of the function being computed, or nobody does. A well-known result due to Cleve shows that fair MPC is impossible when a majority of the parties are corrupted (which in particular applies to the case of two parties and one corruption). In this talk, we show how to circumvent this impossibility result, and present fair MPC constructions that tolerate up to (n-1) corruptions. The constructions make use of some recent results in timed-release cryptography.

In more detail, we introduce the notion of "resource-fair" protocols, a property which (informally) states that if one party learns the output of the protocol, then so can all other parties, as long as they expend roughly the same amount of resources. As opposed to similar previously proposed definitions, our definition follows the standard simulation paradigm and enjoys strong composability properties. In particular, our definition is similar to the security definition in the universal composability framework, but works in a model that allows any party to request additional resources from the environment to deal with dishonest parties that may prematurely abort. Doing this improves on a previous approach that required the protocol to be aware of the adversary's running time, which we will also briefly review.