Secure Two-Party Computation with Fairness – A Necessary Design Principle

Yehuda Lindell

ABSTRACT: Protocols for secure two-party computation enable a pair of mutually distrustful parties to carry out a joint computation of their private inputs without revealing anything but the output. One important security property that has been considered is that of fairness which guarantees that if one party learns the output then so does the other. In the case of two-party computation, fairness is not always possible, and in particular two parties cannot fairly toss a coin (Cleve, 1986). Despite this, it is actually possible to securely compute many two-party functions with fairness. However, all two-party protocols known that achieve fairness have the unique property that the effective input of the corrupted party is not determined at any fixed point in the protocol, in contrast to almost all other known protocols.

In this talk, we address the question as to whether or not the property of not having an input committal round is inherent for achieving fairness for two parties. In order to do so, we revisit the definition of security of Micali and Rogaway, that explicitly requires the existence of such a committal round, and adapt the definition of Canetti in the two-party setting to incorporate the spirit of a committal round. We show that under such a definition, it is impossible to achieve fairness for any non-constant two-party function. This result deepens our understanding as to the type of protocol construction that is needed for achieving fairness. In addition, our result shows that there is a fundamental difference between the definition of security of Micali and Rogaway and that of Canetti which has become the standard today. Specifically, many functions can be securely computed with fairness under the definition of Canetti but no non-constant function can be securely computed with fairness under the definition of Micali and Rogaway.

Joint work with Tal Rabin