ETH Zurich, Department of Computer Science
SS 2018

Prof. Ueli Maurer
Dr. Martin Hirt
Chen-Da Liu Zhang

Cryptographic Protocols
Exercise 10

10.1 Information-Theoretic Commitment Transfer Protocol

a) Consider the information-theoretically secure (distributed) commitment scheme from the lecture. Describe the state achieved by the COMMIT protocol, i.e., describe the output of each player and the consistency condition among these outputs.

b) Design a commitment transfer protocol CTP for a commitment created via COMMIT. Show that your protocol is secure. How many corrupted players can be tolerated?

10.2 Information-Theoretic Commitment Multiplication Protocol

a) Show that the commitment multiplication protocol (CMP) from the lecture is secure for $t < n/3$, i.e., that it satisfies the properties:

1. CORRECTNESS: At the end of CMP, either the dealer D is committed to c such that $c = ab$, or it is publicly seen that D is corrupted.
2. PRIVACY: Up to t players (not including D) obtain no information on the values a and b.

b) Show that the protocol CMP is insecure if $t \geq n/3$.

HINT: Show that if $n = 3t$, then an adversary corrupting t players (including D) can achieve that at the end of the protocol player D is committed to some $c' \neq ab$.

10.3 Beaver's Multiplication Triples

One of the major bottlenecks in the MPC protocol seen in the lecture is to evaluate the multiplication gates. In this task, we consider an (usually expensive) *off-line* phase in which neither the input nor the function to be evaluated needs to be known, and an (efficient) *on-line* phase which uses the pre-computed values in the first phase to evaluate the actual function. We show that one can compute *multiplication triples* in the off-line phase to significantly reduce the cost of evaluating multiplication gates in the on-line phase. A multiplication triple consists of three secret values a, b and c which are shared among the parties, such that a, b are uniformly random and $c = ab$.

a) Design a protocol that allows the parties to create a multiplication triple.

b) Let (a, b, c) be a multiplication triple. Given a sharing of x and a sharing of y, how can a party compute a sharing of xy efficiently?