ETH Zurich, Department of Computer Science
SS 2018

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

Cryptographic Protocols
Exercise 7

7.1 Types of Oblivious Transfer

Oblivious transfer (OT) comes in several variants:

- *Rabin OT:* Alice transmits a bit b to Bob, who receives b with probability 1/2 while Alice does not know which is the case. That is, the output of Bob is either b or $\perp$ (indicating that the bit was not received).
- 1*-out-of-*2 *OT:* Alice holds two bits b_0 and b_1. For a bit $c \in \{0, 1\}$ of Bob's choice, he can learn b_c but not b_{1-c}, and Alice does not learn c.
- 1*-out-of-k OT for $k > 2$:* Alice holds k bits $b_1, \ldots, b_k$. For $c \in \{1, \ldots, k\}$ of Bob's choice, he can learn b_c but none of the others, and Alice does not learn c.

Prove the equivalence of these three variants, by providing the following reductions:

a) 1-out-of-k OT $\Longrightarrow$ 1-out-of-2 OT

b) 1-out-of-2 OT $\Longrightarrow$ 1-out-of-k OT
Hint: In your protocol, the sender should choose k random bits and invoke the 1-out-of-2 OT protocol k times.

c) 1-out-of-2 $\Longrightarrow$ Rabin OT

d) Rabin OT $\Longrightarrow$ 1-out-of-2 OT
Hint: Use Rabin OT to send sufficiently many random bits. In your protocol, the receiver might learn both bits, but with negligible probability only.

7.2 Multi-Party Computation with Oblivious Transfer

In the lecture, it was shown that 1-out-of-k oblivious string transfer (OST) can be used by two parties A and B to securely evaluate an arbitrary function $g : \mathbb{Z}_m^2 \to \mathbb{Z}_m$.

a) Generalize the above protocol to the case of *three* parties A, B, and C, with inputs $x, y, z \in \mathbb{Z}_m$, respectively, who wish to compute a function $f : \mathbb{Z}_m^3 \to \mathbb{Z}_m$.
Hint: Which strings should A send to B via OT? Which entry should B choose, and which strings should he send to C via OT?

b) Is your protocol from **a)** secure against a passive adversary? If not, give an example of a function f where some party receives too much information by executing the protocol.

c) Modify your protocol to make it secure against a passive adversary.

7.3 Trusted Party Operations

In the lecture we consider a trusted party who can receive inputs, give outputs, and perform addition and multiplication over a field $\mathbb{F}$ (see Slides: Part 06). In this exercise, we investigate how the trusted party can perform further operations. Consider a field $\mathbb{F}$ with $|\mathbb{F}| = p$ for a prime p.

a) An instruction we would like the trusted party to be able to do is to generate a secret random value. How can this be achieved?

b) Given a value $x \in \mathbb{F}$, how can the trusted party compute x^{-1}? What happens when $x = 0$? How many multiplications are evaluated?

HINT: Use Fermat's Little theorem.

c) Consider a trusted party who can also generate secret random values. Design a more efficient way to compute the inverse operation. What happens when $x = 0$?

HINT: Generate a random value r, compute and reveal $y = x \cdot r$.

d) Let $x, y, c \in \mathbb{F}$. Consider the following instruction:

$$z = \begin{cases} x & \text{if } c = 0 \\ y & \text{otherwise} \end{cases}$$

How can the trusted party compute this instruction?

HINT: First, find a solution that works for $c \in \{0, 1\}$. Then, solve the general case.