ETH Zurich, Department of Computer Science
SS 2021

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

Cryptographic Protocols
Exercise 7

7.1 Protocols and Specifications

Parties P_1 and P_2 hold input bits x_1 and x_2, respectively. They want that P_2 learns the AND of their inputs.

Specification 1

P_1 (resp. P_2) holds input bit x_1 (resp. x_2).
1: P_1 (resp. P_2) sends x_1 (resp. x_2) to TTP.
2: TTP sends $y = x_1$ to P_2.
3: P_2 outputs y.

Specification 2

P_1 (resp. P_2) holds input bit x_1 (resp. x_2).
1: P_1 (resp. P_2) sends x_1 (resp. x_2) to TTP.
2: TTP sends $y = x_1 \wedge x_2$ to P_2.
3: P_2 outputs y.

Protocol 3

P_1 holds input bit x_1, P_2 holds input bit x_2.
1: P_1 sends x_1 to P_2.
2: P_2 computes $y = x_1 \wedge x_2$.
3: P_2 outputs y.

a) Does Protocol 3 satisfy Specification 1 in the case where both parties are honest? What about Specification 2?

b) Does Protocol 3 satisfy Specification 2 when the adversary passively corrupts P_2? What if the adversary actively corrupts P_2?

Now consider three parties P_1, P_2 and P_3 with input bits x_1, x_2 and x_3, respectively. They want that P_1 and P_3 learn the AND of the three inputs.

Specification 4

P_1 (resp. P_2, P_3) has input bit x_1 (resp. x_2,x_3)
1: Each party P_i sends x_i to TTP.
2: TTP sends $y = x_1 \wedge x_2 \wedge x_3$ to P_1 and P_3.
3: P_1 and P_3 output y.

Protocol 5

P_1 (resp. P_2, P_3) has input bit x_1 (resp. x_2,x_3)
1: P_1 sends x_1 to P_2.
2: P_2 sends $y_2 = x_1 \wedge x_2$ to P_3.
3: P_3 sends $y_3 = y_2 \wedge x_3$ to P_1.
4: P_1 and P_3 output y_3.

c) Does Protocol 5 satisfy Specification 4 when the adversary passively corrupts P_1 and P_2? What about P_1 and P_3? Is there a subset of players the adversary can passively corrupt so that the protocol is secure? For the same sets of corrupted players, analyze the protocol when the adversary is active.

7.2 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.

7.3 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.