Institute of Computer Science, University of Bern
Cryptographic Protocols, FS 2021

Prof. Christian Cachin
Jovana Mićić

Exercise 7

7.1 Oblivious transfer from private set intersection (4pt)

Oblivious transfer (OT) and private set intersection (PSI) are important and basic secure computation protocols for two parties. They are also equivalent, in the sense that one primitive may be used to implement the other one without adding any further cryptographic mechanisms.

Recall that in a $\binom{2}{1}$-OT of bits, the sender S has two inputs, $x_0 \in \{0,1\}$ and $x_1 \in \{0,1\}$, and the receiver R has one input $b \in \{0,1\}$. The goal is that R learns x_b, and neither S nor R learns anything beyond that.

Show how to implement $\binom{2}{1}$-OT of bits using PSI of 2-bit strings. In such a reduction of one primitive to another one, it is usually permitted to call the underlying primitive mulitple times, but here it suffices to call PSI once.

Hint: Let S and R determine a set of two 2-bit strings each.

7.2 Private set intersection from additively homomorphic encryption (4pt)

Recall the additively homomorphic encryption (e.g., based on ElGamal) that has operations $\otimes$ and $\oplus$ such that for every two messages m_1 and m_2 encrypted under the same public key, there is an encryption of a message $m_3 = m_1 \oplus m_2$ such that

$$\mathsf{Enc}(pk, m_1) \otimes \mathsf{Enc}(pk, m_2) \;=\; \mathsf{Enc}(pk, m_3).$$

Imagine that $\oplus$ represents addition in some finite field $GF(q)$. It is not possible to compute an encryption $(m_1)^2$ or any other polynomial in m_1 (computed over $GF(q)$) from an encryption of m_1. However, the party that knows m_1 may initially supply encryptions of $(m_1)^2$, $(m_1)^3$, ..., and the other party may use those pre-computed values to obtain encryptions of arbitrary polynomials in m_1.

Recall the model of PSI, where A starts with a set $\mathcal{X}$ and B starts with a set $\mathcal{Y}$ and their goal is to compute $\mathcal{X} \cap \mathcal{Y}$ privately.

1. Consider the polynomial $P(y) = \prod_{x \in \mathcal{X}}(x - y)$. Use the above ideas to devise a first protocol, in which A learns whether $P(y) = 0$ for some $y \in \mathcal{Y}$ that B chooses. B should not learn whether $y \in \mathcal{X}$.
2. Extend this to a second protocol, which runs the first protocol for each $y \in \mathcal{Y}$. The goal is that A learns $\mathcal{X} \cap \mathcal{Y}$.

7.3 Secure two-party AND using oblivious transfer (2pt)

Develop a protocol for A and B, with private inputs $x \in \{0,1\}$ and $y \in \{0,1\}$, respectively, to compute $x \wedge y$, i.e., $x \cdot y$ in $GF(2)$. It should rely on $\binom{2}{1}$-OT of bits and may use further direct messages.