Chapter 7
Oblivious Transfer and Applications

In this chapter we carry out an in-depth study of the problem of constructing efficient protocols for oblivious transfer (as introduced in Section 3.2.2). Oblivious transfer is one of the most important building blocks in cryptography, and is very useful for constructing secure protocols. We demonstrate this by showing how to achieve secure pseudorandom function evaluation using oblivious transfer.

The 1-out-of-2 oblivious transfer (OT) functionality, denoted by \( \mathcal{F}_{\text{OT}} \), is defined by \( ((x_0, x_1), \sigma) \mapsto (\lambda, x_{\sigma}) \) where \( \lambda \) denotes the empty string. That is, a sender has a pair of inputs \( (x_0, x_1) \) and a receiver has a bit \( \sigma \). The aim of the protocol is for the receiver to receive \( x_{\sigma} \) (and \( x_{\sigma} \) only), without revealing anything about \( \sigma \) to the sender. Oblivious transfer is one of the basic building blocks of cryptographic protocols and its efficiency is a bottleneck in many protocols using it; see Chapter 4 for just one example. In this chapter we present three protocols for three different notions of security: privacy only, one-sided simulation and full simulation, as defined in Sections 2.6.1, 2.6.2 and 2.3, respectively.

Our starting point is the private and highly efficient protocol of [62] which relies on the hardness of the decisional Diffie-Hellman (DDH) problem. Thereafter, in Section 7.3 we show how to slightly modify this protocol so that it achieves one-sided simulatability. Then, in Section 7.4 we present another protocol for this task that is fully secure in the presence of malicious adversaries. The security of this protocol follows from the same hardness assumption (e.g., DDH) and the (exact) computation and communication costs are approximately doubled. We conclude with a construction for batch oblivious transfer which is of great importance for protocols such as that of Chapter 4, where many OT executions are run together. This protocol is considerably more efficient than just taking the previous protocol and running it many times in parallel.

In addition to the protocols for OT, we present the protocol of [28] for securely computing the pseudorandom function evaluation functionality, denoted by \( \mathcal{F}_{\text{PRF}} \), and defined by \( (k, x) \mapsto (\lambda, F_{\text{PRF}}(k, x)) \). The aim of the pro-
protocol is for party $P_2$ to learn $F_{PRF}(k, x)$ (and $F_{PRF}(k, x)$ only), without revealing anything about $x$ to $P_1$. We consider the concrete pseudorandom function of [64] and present the secure (slightly modified) construction of [28] for this specific function. We will use $F_{PRF}$ in later constructions. We remark that with one exception, the protocols do not utilize any specific properties of this function and can employ any other (possibly more efficient) secure protocol that implements the pseudorandom function evaluation functionality. All the protocols in this chapter achieve security in the presence of malicious adversaries.

7.1 Notational Conventions for Protocols

We now describe some notational conventions that are used in this and the subsequent chapters. The aim of our notation is to enable the reader to quickly identify if a protocol is secure in the semi-honest or covert model, or whether it provides privacy only, one-sided simulatability or full security in the presence of malicious adversaries. In addition, since our more complex protocols use a number of subprotocols, we include notation to identify the function being computed. The general template is

$$\pi^S_f$$

where $f$ is the function being computed and $S$ is the level of security obtained. For example, protocols for computing the oblivious transfer functionality in the presence of semi-honest and covert models are denoted by $\pi^{SH}$ and $\pi^{CO}$, respectively. Regarding malicious adversaries, protocols achieving privacy only and one-sided simulatability are denoted by $\pi^{P}$ and $\pi^{OS}$, respectively. Protocols achieving full simulation are not given any superscript, and so $\pi^{OT}$ denotes an OT protocol achieving full security in the presence of malicious adversaries.

7.2 Oblivious Transfer – Privacy Only

7.2.1 A Protocol Based on the DDH Assumption

We now present the protocol of [62] that computes $F_{OT}$ with privacy in the presence of malicious adversaries; see Definition 2.6.1 in Section 2.6.1 for the definition of private oblivious transfer. This protocol involves a sender $S$ and a receiver $R$ and is implemented in two rounds. Its high efficiency is due to