Errata to MPC Course

Naman Kumar

March 22, 2024

Abstract

Basic notes on MPC.

Contents

1 Secure N-party Computation

It's probably worth reiterating some of the formalisms of these definitions with a bit more lucidity, just as a simple reference and as some sort of illumination.

1.1 Real World Instantiation

In the real world, N parties have inputs $\mathbf{x} = (x_1, \ldots, x_n)$, an agreed-upon function $f : (x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n)$ and they follow a protocol Π which gives them the set of *party outputs* $\text{OUT}_\Pi(\lambda, \mathbf{x}) := (y_1, \ldots, y_n)$. Assume that t of these parties are controlled by the adversary; define the *view* $\text{VIEW}_{\Pi,\mathcal{A}}(\lambda, \mathbf{x})$ to be the set of all inputs of corrupted parties, along with messages exchanged that the adversary has sent, received, or eavesdropped-upon.

Definition 1.1 (Real World Distribution)**.** *The real-world output* $\text{REAL}_{\Pi,\mathcal{A}}(\lambda, \mathbf{x})$ *is defined as the tuple*

$$\text{REAL}_{\Pi,\mathcal{A}}(\lambda, \mathbf{x}) := (\text{OUT}_\Pi(\lambda, \mathbf{x}), \text{VIEW}_{\Pi,\mathcal{A}}(\lambda, \mathbf{x})).$$

1.2 Ideal World Instantiation

In the ideal world, there is no protocol, only a functionality $\mathcal{F}$ which takes in inputs from each party, computes the output f, and returns $(f(x_1), \ldots, f(x_n))$ to the parties. The output, $\text{OUT}_F(\lambda, \mathbf{x})$ is the same – the output of the parties after the protocol execution – while instead of the *view* (which is, of course, not identical to that of the actual ideal-world 'protocol' execution; a protocol could be multi-round), there is the *view of a simulator* (a 'fake' adversary which works in the ideal world, but has access to the inputs of the real adversary), which is the output of the simulator on any given execution.

Definition 1.2 (Ideal World Distribution)**.** *The ideal-world output* $\text{IDEAL}_{\mathcal{F},\mathcal{S}}(\lambda, \mathbf{x})$ *is defined as the tuple*

$$\text{IDEAL}_{\mathcal{F},\mathcal{S}}(\lambda, \mathbf{x}) := (\text{OUT}_\mathcal{F}(\lambda, \mathbf{x}), \text{VIEW}_{\mathcal{F},\mathcal{S}}(\lambda, \mathbf{x})).$$

Note that these distributions are *joint* distributions.

1.3 Secure N-Party Protocol

Definition 1.3 (t-Privacy of a Protocol)**.** *An N-Party protocol Π is considered t-private if for any PPT adversary that corrupts t of the parties, there exists a PPT simulator such that*

$$\{\text{IDEAL}_{\mathcal{F},\mathcal{S}}(\lambda, \mathbf{x})\} \equiv \{\text{REAL}_{\Pi,\mathcal{A}}(\lambda, \mathbf{x})\}.$$

Remark. There's two examples here of the non-privacy of a protocol. One of them is a function which takes no inputs and outputs a random $b \leftarrow \{0, 1\}$ to one of the parties,

say P_0. The protocol in which P_1 samples a bit and sends it to P_0 is not secure since if P_1 is corrupted, the view of the adversary is different; it has the bit b in it. Similarly, if a functionality takes no input and outputs pq, $(p+q)$ to the parties, then, again, the protocol in which P_0 chooses p, q and sends $p+q$ is insecure since it learns the numbers p and q.

2 Oblivious Transfer

Oblivious Transfer is a simple MPC functionality parametrized by a selection of sender inputs and a receiver index.

Parameters: The sender S has a selection of N input strings $(m_0, \ldots, m_{N-1})$, while the receiver R has an index $i \in [N]$.
Outputs: S receives nothing while R receives m_i.

Figure 1: 1-out-of-N Oblivious Transfer.

2.1 Protocol for 1-out-of-2 OT

We now demonstrate a simple protocol for 1-out-of-2 OT [EGL85]. We begin with a CPA-secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, and define the notion of oblivious sampling.

Definition 2.1 (PKE with Obliviously Sampleable Encryption Key)**.** *An encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has obliviously sampleable public keys if there exist algorithms* Samp *and* pkSamp *such that*

- $\{\mathsf{Samp}(1^\lambda)\}$ *is computationally indistinguishable from* $\{\mathsf{pk} : (\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)\}$.
- $\{(\mathsf{pk}, r) : r \xleftarrow{\$} \{0,1\}^\lambda, \mathsf{pk} \leftarrow \mathsf{Samp}(1^\lambda; r)\}$ *is computationally indistinguishable from* $\{(\mathsf{pk}, r) : (\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}, r \leftarrow \mathsf{pkSim}(\mathsf{pk})\}$.

The protocol proceeds as follows.

- Receiver runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and $\mathsf{pk}' \leftarrow \mathsf{Samp}(1^\lambda)$ and sets $\mathsf{pk}_b = \mathsf{pk}$, and $\mathsf{pk}_{1-b} = \mathsf{pk}'$.
- Receiver sends pk_0 and pk_1.
- Sender encrypts $c_i = \mathsf{Enc}_{\mathsf{pk}_i}(m_i)$.
- Sender sends c_0, c_1.
- Receiver decrypts $m_b = \mathsf{Dec}_{\mathsf{sk}_b}(c_b)$.

Figure 2: 1-out-of-2 Oblivious Transfer from obliviously sampleable PKE.

3 Modifications to OT

We first construct two variants of OT that provide the additional property of *information-theoretic* security against the receiver.

3.1 Information-Theoretically Secure OT

- Sender samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$ and sends pk to receiver.
- Receiver samples $s_0, s_1 \leftarrow \{0,1\}$ and sets $c_b = \mathsf{Enc}_{\mathsf{pk}}(s_b)$ and $c_{1-b} \leftarrow \mathsf{Samp}$, and sends c_0, c_1 to sender.
- Sender sets $s_i = \mathsf{Dec}_{\mathsf{sk}}(c_i)$ and sends $x_i \oplus s_i$.
- Receiver computes
$$x_b = s_b \oplus (s_b \oplus x_b).$$

Figure 3: Variant of OT.

Informally, the security against sender is dependent on the (computational) indistinguishability of determining c_{1-b} from an honestly sampled encryption and on the security of the encryption scheme, while the security against the receiver is information-theoretic since it cannot determine s_1.

We see another variant of OT, secure under the DDH assumption.

- Receiver samples $r \leftarrow \mathbb{Z}_q$ and sets $(h_0, h_1) = (g_0^r, g_1^{r+b})$. It sends (h_0, h_1).
- Sender samples $a_0, b_0, a_1, b_1 \leftarrow \mathbb{Z}_q$ and sets
$$c_i = (g_0^{a_i} g_1^{b_i}, h_0^{a_i} h_1^{b_1} / (i \cdot g_1^{b_1}) x_i)$$
and sends (c_0, c_1).
- Receiver parses c_b as (c^1, c^2) and computes $x_b = c^2/(c^1)^r$.

Figure 4: Variant of OT secure assuming DDH.

3.1.1 Dual-Mode Cryptosystem

A dual-mode cryptosystem serves as a generic 'toggling' mechanism to achieve information-theoretic OT against the sender or the receiver. A strict definition of dual-mode cryptosystem is given in [PVW08], along with a generic technique that realizes the OT functionality.

3.2 Extending the Usefulness of OT

We now look at two techniques which allow subtle (more useful) variants of OT.

3.2.1 Domain Extension

This technique allows us to obtain OT for ℓ-bit strings from OT for λ-bit strings. Note that λ is the security parameter – hence, it has to be a reasonable key length.

- Sender samples $k_0, k_1 \leftarrow \{0,1\}^\lambda$ and sends it to $\mathcal{F}_{\mathsf{OT}}$.
- Receiver sends b to $\mathcal{F}_{\mathsf{OT}}$ and receives k_b.
- Sender sends $c_i = \mathsf{Enc}_{k_i}(m_i)$ for each $i \in \{0,1\}$.
- Receiver decrypts $m_b = \mathsf{Dec}_{k_b}(c_b)$.

Figure 5: OT Domain Extension.

3.2.2 1-out-of-N OT from 1-out-of-2 OT

For simplicity, we can assume $N = 2^k$ for some k. Suppose that the receiver wants m_α for some $|\alpha| = k$.

- Sender samples k_i^b for $b \in \{0,1\}$ and $i \in [k]$, and submits (k_i^0, k_i^1) to $\mathcal{F}_{\mathsf{OT}}$.
- Receiver sends α_i to the ith OT.
- Sender sets
$$c_\beta = m_\beta \oplus \bigoplus_{i=1}^{k} F_{k_{\beta_i}}(\beta)$$
and sends each c_β.
- Receiver decrypts $m_\alpha = c_\alpha \oplus \bigoplus_{i=1}^{k} F_{k_{\alpha_i}}(\alpha)$.

Figure 6: 1-out-of-N OT from 1-out-of-2 OT.

4 OT Extension

As we have seen, it is possible to perform OT of strings of long length using OT for strings of shorter length and a PRG. We will now answer the question of whether it's possible to perform a greater *number* of OTs using a fewer number of OTs.

4.1 IKNP OT Extension

In this section we will describe the IKNP OT Extension protocol from [IKNP03]. The protocol works by extending λ pairs of m-bit OT to m pairs of ℓ-bit OT. A full description is below. We note by $\mathbf{OT}_b^a$ an a-pairs b-bit OT functionality.

Parameters: The Sender holds m pairs of ℓ-bit strings $\{(m_{i,b})\}$. The receiver holds m selection bits $r = (r_1, \ldots, r_m)$. $H : [m] \times \{0,1\}^\lambda \to \{0,1\}^\ell$ is a random oracle.

Protocol:

- Sender selects $s \xleftarrow{\$} \{0,1\}^\lambda$ and sends it to $\mathbf{OT}_m^\lambda$ (as the receiver).
- Receiver selects a random matrix

$$T = \begin{pmatrix} T_1 & T_2 & \ldots & T_\lambda \end{pmatrix}_{m \times \lambda} = \begin{pmatrix} T^1 \\ T^2 \\ \vdots \\ T^m \end{pmatrix}_{m \times \lambda}$$

  and sends the inputs $\{(T_i, r \oplus T_i)\}_{i \in [\lambda]}$ to $\mathbf{OT}_m^\lambda$ (as the sender).
- Denote by Q the matrix received by the sender, which is

$$Q = \begin{pmatrix} Q_1 & Q_2 & \ldots & Q_\lambda \end{pmatrix}_{m \times \lambda} = \begin{pmatrix} Q^1 \\ Q^2 \\ \vdots \\ Q^m \end{pmatrix}_{m \times \lambda}$$

- Sender sends

$$(y_{j,0}, y_{j,1}) = (x_{j,0} \oplus H(j, Q^j), x_{j,1} \oplus H(j, Q^j \oplus s)).$$

- For each $1 \leq j \leq m$, receiver outputs $y_{j,r_j} \oplus H(j, T^j)$.

Figure 7: The IKNP OT Extension Protocol.

Correctness. We will now see why this protocol works. Consider $r_i = 0$ for some i. It follows that $(r_i)_{j \in [\lambda]} \oplus T^i = T^i$, and so $T^i = Q^i$ (regardless of whatever s was). It

immediately follows that $H(j, T^j) = H(j, Q^j)$.

On the other hand, if $r_i = 1$, then $Q^i = (s \cdot r) \oplus T^j = s \oplus T^j$. Correctness follows similarly.

Discussion. The protocol provides perfect security against a semi-honest (even malicious) sender and statistical security against a semi-honest receiver. The protocol is instantiated with a random oracle, but can be instantiated with a weaker primitive called a *correlation-robust hash function.*

Definition 4.1 (Correlation Robustness)**.** *An efficiently computable function $h : \{0,1\}^* \to \{0,1\}$ is said to be correlation robust if for any polynomial q and PPT adversary $\mathcal{A}$ there exists a negligible function $\epsilon(\cdot)$ such that*

$$|\Pr[\mathcal{A}(t_1, \ldots, t_m, h(t_1 \oplus s), \ldots, h(t_m \oplus s)) = 1] - \Pr[\mathcal{A}(U_{(t+1)m}) = 1]| \leq \epsilon(\lambda)$$

where $m \leq q(\lambda)$ and $|C| \leq p(\lambda)$. The probability is taken over random and independent choices of $s, t_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i \in [m]$.

The authors also modify the scheme to achieve full security against a malicious receiver through a cut-and-choose mechanism, in which a the receiver and the sender execute k instances of the (committed) protocol in parallel and the sender randomly checks whether a certain number of instances were correctly performed (and aborts otherwise).

Parameters. The above protocol is secure as long as $m = 2^{o(\lambda)}$. Note that if, say, $m = 2^\lambda$, then the security of the correlation-robust hash function breaks down since this allows a 'repeat', i.e. $H(j, Q^j \oplus s)$ will be called twice for some value of j. We will see the impossibility of $2^{\Omega(\lambda)}$-pairs OT extension in Minicrypt along with other impossibility results below. This concludes that the protocol is asymptotically optimal.

Improvements. [BCG+19] achieves n pairs using $\log n$ OTs and the (computational) LPN (Learning Parity with Noise) assumption. [Roy22] presents a practical improvement on the protocol.

4.2 Feasibility of OT Extension

The protocol of [IKNP03] is not the only OT Extension protocol. Before this, [Bea96] showed that k OTs can be extended to k^c OTs making a non-black box use of a one way function. The protocol of [IKNP03] can gain up to superpolynomial OTs with a black-box use of an OWF, but makes use of a Random Oracle.

Open Problem. Is there a protocol for (subexponential) OT extension that makes black-box use of a One Way Function without a random oracle?

[LZ13] study the feasibility of OT Extension, and prove the following results.

Theorem 4.1 (Information-Theoretic OT Extension)**.** *If there exists an OT extension protocol from n to $n+1$ (with security in the presence of static semi-honest adversaries), then there exist one-way functions.*

This proves that OT cannot be extended information-theoretically. Note that while the protocol of [IKNP03] does not make explicit use of an OWF, it does use a random oracle (which implies OT extension).

Theorem 4.2 (Adaptively Secure OT Extension)**.** *If there exists an OT extension protocol from n to $n+1$ that is secure in the presence of adaptive semi-honest adversaries, then there exists an oblivious transfer protocol that is secure in the presence of static semi-honest adversaries.*

This proves that any adaptive OT extension protocol involves constructing statically secure OT extension from scratch. The problem of constructing adaptively secure OT extension was solved in [BPRS17], though their protocol uses public-key cryptography.

Theorem 4.3 (Extending Logarithmic OTs)**.** *If there exists an OT extension protocol from $f(n) = O(\log n)$ to $f(n)_1$ that is secure in the presence of static malicious adversaries, then there exists an OT protocol that is secure in the presence of static malicious adversaries.*

This proves that any OT extension protocol that have exponential extension must make use of an exponential number of OTs itself.

5 The GMW Protocol

References

[BCG+19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent ot extension and more. Cryptology ePrint Archive, Paper 2019/448, 2019. https://eprint.iacr.org/2019/448.

[Bea96] Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 479–488, 1996.

[BPRS17] Megha Byali, Arpita Patra, Divya Ravi, and Pratik Sarkar. Fast and universally-composable oblivious transfer and commitment scheme with adaptive security. *Cryptology ePrint Archive*, 2017.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, jun 1985.

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 145–161, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[LZ13] Yehuda Lindell and Hila Zarosim. On the feasibility of extending oblivious transfer. In *Theory of Cryptography Conference*, pages 519–538. Springer, 2013.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 554–571, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Roy22] Lawrence Roy. Softspokenot: Communication–computation tradeoffs in ot extension. Cryptology ePrint Archive, Paper 2022/192, 2022. https://eprint.iacr.org/2022/192.