CS 294 – Secure Computation April 7, 2015

Lecture 12: Two Round MPC from iO

Instructor: Sanjam Garg *Scribe: Tobias Boelter*

1 Motivation

In the last lecture we saw how fully homomorphic encryption (FHE) can be used to reduce the communication of MPC protocols to be *independent* of the circuit size, which is a celebrated result.

Today we will see how indistinguishability obfuscation (iO short) can be used to achieve a two-round MPC protocol secure against static malicious adversaries [1]. The stronger property of adaptive security can also be achieved [2], precisely two-round MPC secure against active adaptive attackers in the model where honest parties are not trusted to properly erase their internal state.

Constant-round MPC can also be achieved under other assumptions, but in this lecture we will look at how to build it from iO.

2 Indistinguishability Obfuscation

The goal of obfuscation is, given a program P, to produce a program $\mathrm{iO}(P)$ that has the same input-output behavior of P, but reveals nothing about the internals of P.

Definition 1 (Indistinguishability Obfuscation) *A uniform* PPT *machine* iO *is called an indistinguishability obfuscator for the circuit class $\{C_k\}$, if:*

1. *correctness:*
$$\forall k \in \mathbb{N}, \forall C \in \{C_k\}, \forall x$$
$$\Pr[C'(x) = C(x) \mid C' \leftarrow \mathrm{iO}(1^k, C)] = 1$$

2. *poly-slowdown*
$$|\mathrm{iO}(1^k, C)| = \mathrm{poly}(k + |C|)$$

3. *iO-security*
$$\forall C_1, C_2 \in \{C_k\} \textit{ such that } |C_1| = |C_2| \textit{ and } \forall x : C_1(x) = C_2(x)$$
$$\textit{it should hold that } \mathrm{iO}(1^k, C_1) \overset{c}{\approx} \mathrm{iO}(1^k, C_2)$$

3 Two Round MPC from iO

3.1 Prerequisite

Our goal is to construct a two-round multi-party computation protocol secure against static active attackers. We will construct this in the common reference string model and in the model of an open asynchronous multi-party network with broadcast. We further assume $IND\text{-}CCA$ secure public-key encryption and statistically-sound non-interactive zero knowledge (NIZK) proofs exist.

3.2 Intuition

Lemma 1 *One-round MPC is impossible*

Proof. The malicious party could just send garbage but still compute the functionality with its real input in its head. Thereby it can learn the output of the function on different inputs without letting the other parties learn it, which violates security. ■

Because of this attack, we need at least two rounds of communication. One way to avoid this attack is to have all parties commit to their inputs in the first round and have the output only be generated in the second round. The concrete idea is to have every party commit to its input and its randomness in this first round. Then we apply a compiler that takes an (possibly highly interactive) existing MPC protocol and have each party obfuscate their next-message function in each round and broadcast them to the other parties. Given the obfuscations all parties can emulate the protocol in their head and finally obtain the output.

Recall that the next-message circuits have hard-coded the party's input x_i, randomness r_i and take as input all previous received messages to produce the next message. There is a problem with this approach: Given the obfuscated circuit, one party can evaluate it multiple times on different inputs thereby possibly subverting security. Therefore, we add a check into the circuit that checks, if the input is consistent with the inputs and randomness all parties committed to in the first round. Concretely, each circuit also produces a NIZK proof that their output is consistent with their input, randomness, and transcript so far. Also, the circuits take a NIZK from the previous round as input and check it.

3.3 The Construction

We start with the semi-honest case and subsequently extend it to achieve our final goal. Let π be a semi-honest MPC protocol with t rounds and n parties $P_1, \dots, P_n$. Recall that a message depends on the secret state of the party and the messages this party received previously. Let $m_{i,j}$ be the j-th message sent by the i-th party in the protocol:

$$m_{i,j} = \pi(i, M_{j-1}, x_i, s_i)$$

where s_i is the randomness and x_i the input of the i-th party and M_{r-1} is a matrix that describes the transcript up to round $r-1$, inclusive:

$$M_{r-1} :\overset{\text{def}}{=} (m_{i,j})_{i \leq n,\, j \leq r-1}$$

The common reference string (CRS) σ in the construction consists of a NIZK proof system and a public key pk corresponding to a $IND\text{-}CCA$ secure public key encryption scheme.

Round 1: The i-th party P_i commits to its input x_i and a random string $r_{i,j}$ for each $j \in [n]$, using the $IND\text{-}CCA$ secure encryption scheme comm, i. e.

$$c_i = \mathrm{comm}(x_i), d_{i,j} = \mathrm{comm}(r_{i,j})$$

Round 2:

- The party P_i reveals the random values $\{r_{i,j}\}_{j \neq i \in [n]}$ and generates proofs $\{\gamma_{i,j}\}_{j \neq i \in [n]}$ that these are indeed the values that are encrypted in the ciphertexts $\{d_{i,j}\}_{j \neq i \in [n]}$
- Each party P_i generates and reveals t obfuscations of its augmented next-round function $\mathcal{P}_{i,j}^{0,x_i,\rho_{x_i},r_{i,i},\rho_{r_{i,i}},\{Z_i\},0^{\ell_{i,j}}}$: $(\mathrm{iO}_{i,1}, \dots, \mathrm{iO}_{i,t})$

- P_i broadcasts all the values $\{r_{i,j}\}_{j \neq i \in [n]}$, $\{\gamma_{i,j}\}_{j \neq i \in [n]}$, and $\{\mathrm{iO}_{i,k}\}_{k \in [t]}$

Evaluation (MPC in the Head): For each round $r \in [t]$ each party P_i evaluates the obfuscation $\mathrm{iO}_{i,r}$ of the program $\mathcal{P}_{i,r}$ on input $(R, \Gamma, M_{r-1}, \Phi_{r-1})$ where R is the matrix of all $r_{i,j}$ without the $r_{i,i}$. Similarly, Γ is the matrix of all $\gamma_{i,j}$ without the $\gamma_{i,i}$. M_{r-1} is as defined as above and Φ is the matrix of all $(\phi_{i,j})_{i \in [n], j \in [r]}$. The party obtains $m_{i,r}, \ldots, m_{n,r}$ and $\phi_{i,r}, \ldots, \phi_{n,r}$

$\mathcal{P}_{i,j}^{\mathrm{flag}, x_i, \rho_{x_i}, r_{i,i}, \rho_{r_{i,i}}, \{Z_i\}, \mathrm{fixedOutput}}(R, \Gamma, M_{j-1}, \Phi)$ does the following:

- $\forall p, q \in [n]$ such that $p \neq q$ check that $\gamma_{p,q}$ is an accepting proof under σ for the NP-statement:

$$\{\exists \rho_{r_{p,q}} \mid d_{p,q} = \mathsf{Enc}(p \,\|\, r_{p,q}; \rho_{r_{p,q}})\}$$

- $\forall p \in [n], q \in [j-1]$ check that $\phi_{p,q}$ is an accepting proof for the NP-statement:

$$\left\{\begin{matrix} \exists (x_p, r_{p,p}, \rho_{x_p}, \rho_{r_{p,p}}) \mid & & \\ (c_p = \mathsf{Enc}(p, \| \, x_p; \rho_{x_p}) & \wedge \, d_{p,p} = \mathsf{Enc}(p \,\|\, r_{p,p}, \rho_{r_{p,p}}) \, \wedge \, m_{p,p} = \pi_p(x_p, \oplus_{k \in [n]} r_{k,p}, M_{q-1})) \end{matrix}\right\}$$

- If the checks above fail, output $\perp$. Otherwise, if $\mathrm{flag} = 0$ then output $(\pi_i(x_i, \oplus_{j \in [n]} r_{j,i}, M_{j-1}), \phi_{i,j})$ where $\phi_{i,j}$ is the proof that this round was done correctly. Otherwise, output $\mathrm{fixedOutput}$

Figure 1: Obfuscated Programs in the Protocol

3.4 Proof Idea

Correctness is easy to reason from the correctness of the underlying MPC protocol, the correctness of the primitives used and the efficiency of the primitives used.

The idea of the proof is to have four hybrids where H_0 is the honest case, in H_1 the $m_{i,\ell}$ and $d_{i,\ell}$ are hard-coded, in H_2 the NIZK is simulated and H_3 finally uses the simulator of π.

Note that the simulator does not get access to input and randomness of parties. We therefore use a proof of knowledge instead of NIZK.

References

[1] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. *Theory of Cryptography: 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, chapter Two-Round Secure MPC from Indistinguishability Obfuscation, pages 74–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[2] Sanjam Garg and Antigoni Polychroniadou. *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, chapter Two-Round Adaptively Secure MPC from Indistinguishability Obfuscation, pages 614–637. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.