Invoking Semi-Honest 2PC Simulators

Christopher Smith

Last Updated: November 20, 2025

Abstract

Theoretical constructions of MPC protocols often include executions of MPC subprotocols. While composition theorems sometimes yield easier security proofs of the overall protocol, it may be easier/instructive/necessary to prove security by "manually" invoking the simulators for the subprotocols. Restricting ourselves to the semi-honest two-party case, we give a couple generic examples of how a 2PC subprotocol might show up in a larger protocol, and how we can transition from a real view to a hybrid view where the execution of the subprotocol has been replaced by invocation of the simulator. Our first example is an easy case with a uniform reduction where the protocol begins with the 2PC subprotocol. Our second example is only a slightly harder case with a non-uniform reduction where the protocol exchanges some arbitrary messages in a preamble, and then ends with a 2PC subprotocol that is dependent on this preamble.

1 Preliminaries

Non-Uniform Computation. A non-uniform Turing machine is a Turing machine equipped with an extra "advice" tape. At the start of the machine's execution, the tape may be loaded with arbitrary information that can depend on the length of the input on the machine's input tape (but not the content of the input). Typically, one considers polynomial time machines where the length of the advice is polynomial in the input length. In cryptography, the input length is typically the security parameter λ, which is written in unary on every machine's input tape as 1^λ. In what follows, it may often look *notationally* like the non-uniform reductions we give advice to receive more input than just 1^λ, and so one might complain that advice needs to depend on the length of the "entire input", not just a λ-length prefix of the input. However, this is not the case (at least in this document), as any "extra" input beyond 1^λ for reductions receiving advice semantically comes from other machines as part of an interaction or oracle query. More precisely, this means that the extra input is written on additional communication/oracle tapes distinct from the machine's "one true input tape"[1], which is just initialized with 1^λ. These formal technicalities are not particularly important for the purposes of this document, at least beyond convincing the reader that non-uniform advice in our reductions only depends on λ. For a more formal reference on interactive and oracle access Turing machines, and, more generally, computational models in cryptography, see [Gol01].

Definition 1 (Negligible Function)**.** *We say a function μ is negligible if for every positive polynomial p there exists $N \in \mathbb{N}$ such that for all $\lambda > N$ it holds that $\mu(\lambda) < 1/p(\lambda)$.*

[1]To really get in the weeds, interactive Turing machines share a common input tape, so all parties in an interactive computation automatically share a common security parameter. The situation is different with an oracle access machine, which may query its oracle on any input of its choosing, and this input may include an arbitrary security parameter. This subtle issue is sometimes important in security proofs (*e.g.*, [GW11])

Definition 2 (Computational Indistinguishability [Lin17]). *Two probability ensembles (*i.e.*, infinite sequences of random variables)* $X = \{X(a,\lambda)\}_{a\in\{0,1\}^*,\lambda\in\mathbb{N}}$ *and* $Y = \{Y(a,n)\}_{a\in\{0,1\}^*,\lambda\in\mathbb{N}}$ *are computationally indistinguishable, denoted* $X \overset{c}{\approx} Y$*, if for every non-uniform PPT algorithm* D *there exists a negligible function* μ *such that for every* $a \in \{0,1\}^*$ *and every* $\lambda \in \mathbb{N}$*,*

$$\left|\Pr\left[D(1^\lambda, X(a,\lambda), a) = 1\right] - \Pr\left[D(1^\lambda, Y(a,\lambda), a) = 1)\right]\right| \leq \mu(\lambda)$$

Definition 3 (Semi-Honest 2PC [Lin17]). *We say that the two-party protocol* $\Pi = (A, B)$ *securely evaluates a PPT functionality* $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ *in the presence of semi-honest adversaries, if there exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim}_A, \mathsf{Sim}_B)$ *such that for all* x_A, x_B*, and for all* $\lambda \in \mathbb{N}$*, it holds that:*

$$\left\{(\mathsf{Sim}_A(1^\lambda, x_A, f_A(x_A,x_B)), f(x_A,x_B))\right\}_{x_A,x_B,\lambda} \overset{c}{\approx} \left\{(\mathsf{view}_A^\Pi(x_A,x_B,\lambda), \mathsf{out}^\Pi(x_A,x_B,\lambda))\right\}_{x_A,x_B,\lambda}$$

$$\left\{(\mathsf{Sim}_B(1^\lambda, x_B, f_B(x_A,x_B)), f(x_A,x_B))\right\}_{x_A,x_B,\lambda} \overset{c}{\approx} \left\{(\mathsf{view}_B^\Pi(x_A,x_B,\lambda), \mathsf{out}^\Pi(x_A,x_B,\lambda))\right\}_{x_A,x_B,\lambda}$$

where:

- f_A *is* f *projected onto its first coordinate, and* f_B *is* f *projected onto its second coordinate.*
- $\mathsf{view}_A^\Pi(x_A,x_B,\lambda) = (x_A, r_A, \tau_A)$ *with* r_A *as the random coins used by a semi-honest* A *and* τ_A *as the messages received by* A*.* $\mathsf{view}_B^\Pi(x_A,x_B,\lambda)$ *is defined analogously, with* τ_B *as the messages received by* B*.*
- $\mathsf{out}^\Pi(x_A,x_B,\lambda) = (\mathsf{out}_A^\Pi(x_A,x_B,\lambda), \mathsf{out}_B^\Pi(x_A,x_B,\lambda))$ *is the joint output of both parties in an execution of* Π *on inputs* x_A, x_B *with security parameter* λ*.*

2 Easy: Simulated Preamble, Uniform Reduction

Let f_{pre} be any two-party PPT functionality securely computed in the semi-honest setting by some 2PC protocol Π_{pre}. Define a new two-party protocol protocol $\Pi = (A, B)$ where A and B first execute Π_{pre} and then exchange some polynomial number of (basically) arbitrary additional messages. In more detail, define the view of party B in a real execution of Π as follows:

$\mathsf{view}_B^\Pi(x_A, x_B, \lambda)$:

1. Sample sufficient poly-sized randomness $r_A = r_A^{pre} \| r_A^{post}$ and $r_B = r_B^{pre} \| r_B^{post}$ for both parties.
2. Emulate an execution of Π_{pre} where A enters the protocol with input $(1^\lambda, x_A)$ and random coins r_A^{pre}, and B does the same with $(1^\lambda, x_B), r_B^{pre}$. This generates a preamble transcript $(\tau_A^{pre}, \tau_B^{pre})$, where τ_A^{pre} are the messages *received* by A, and τ_B^{pre} is defined analogously. It also generates preamble outputs (y_A^{pre}, y_B^{pre}).
3. Emulate some arbitrary interactive protocol between A and B. If this interaction is randomized, then A uses r_A^{post} as random coins and B uses r_B^{post} as random coins. Let $(\tau_A^{post}, \tau_B^{post})$ denote the transcript of this interaction.
4. Output party B's view consisting of: $(x_B, r_B, \tau_B^{pre}, \tau_B^{post})$.

Obviously, it is impossible to say whether Π is secure. But all we wish to show in this example is that $\mathsf{view}_B^\Pi(x_A, x_B, \lambda)$ is indistinguishable from the following hybrid view where we invoke the simulator Sim_B^{pre} guaranteed by the 2PC security of Π_{pre} for party B:

$H_1(x_A, x_B, \lambda)$:

1. Letting $(y_A^{pre}, y_B^{pre}) = f_{pre}(x_A, x_B)$, compute $(x_B, r_B^{pre}, \widetilde{\tau}_B^{pre}) \leftarrow \mathsf{Sim}_B^{pre}(1^\lambda, x_B, y_B^{pre})$.
2. Generate $\tau_A^{post}, \tau_B^{post}$ with emulation of the same interactive protocol used in the real execution, sampling random coins r_A^{post}, r_B^{post} on the fly as necessary.
3. Output party B's hybrid view consisting of: $(x_B, (r_B^{pre} \| r_B^{post}), \widetilde{\tau}_B^{pre}, \tau_B^{post})$.

Claim 4. $\mathsf{view}_B^\Pi(x_A, x_B, \lambda) \overset{c}{\approx} H_1(x_A, x_B, \lambda)$.

Proof of Claim 4. Let D_H be any poly-time distinguisher between the hybrids. We construct an adversary D_{2pc} with oracle access to D_H against the semi-honest security of Π_{setup} as follows. D_{2pc} parses its input as $(1^\lambda, (x_B, r_B^{pre}, \hat{\tau}_B^{pre}), (y_A^{pre}, y_B^{pre})), x_A, x_B)$, and uses this information to compute $\tau_A^{post}, \tau_B^{post}$ as in the real execution, sampling r_A^{post}, r_B^{post} on the fly as necessary, and setting $r_B = r_B^{pre} \| r_B^{post}$. It then queries $b \leftarrow D_H(1^\lambda, (x_B, r_B, \hat{\tau}_B^{pre}, \tau_B^{post}), x_A, x_B)$, and outputs b.

Suppressing ensemble indices (x_A, x_B, λ) for clarity and without confusion, see that if D_{2pc}'s input is distributed according to $(\mathsf{view}_B^{\Pi_{pre}}, \mathsf{out}^{\Pi_{pre}})$, then D_H's input is distributed according to view_B^{Π}. Conversely, if D_{2pc}'s input is distributed according to $(\mathsf{Sim}_B^{pre}, f_{pre})$, then D_H's input is distributed according to H_1. Thus, if any D_H distinguishes between view_B^{Π} and H_1 with non-negligible advantage, then D_{2pc} breaks the semi-honest security of Π_{pre}. The claim follows. □

3 Harder: Simulated Postamble, Non-Uniform Reduction

Now let f_{post} be any two-party PPT functionality securely computed in the semi-honest setting by some 2PC protocol Π_{post}. Define a new protocol $\Pi = (A, B)$ where A and B first exchange some arbitrary messages in a preamble, and then run Π_{post} using information derived from the preamble. More formally, define the view of party B in a real execution of Π as follows:

$\mathsf{view}_B^{\Pi}(x_A, x_B, \lambda)$:

1. Sample sufficient poly-sized randomness $r_A = r_A^{pre} \| r_A^{post}$ and $r_B = r_B^{pre} \| r_B^{post}$ for both parties.
2. Let $z(\cdot, \cdot)$ be some arbitrary PPT computable function. Emulate an interactive protocol between A and B for computing $z_A, z_B = z(x_A, x_B)$. If the interaction is randomized, A uses r_A^{pre} as random coins and B uses r_B^{pre} as random coins. Let $\tau_{pre} = (\tau_A^{pre}, \tau_B^{pre})$ denote the transcript of this preamble.
3. Emulate an execution of Π_{post}, where A enters the protocol with input $(1^\lambda, z_A)$ and random coins r_A^{post}, and B enters the protocol with input $(1^\lambda, z_B)$ and random coins r_B^{post}. Let $y = (y_A, y_B)$ and $\tau_{post} = (\tau_A^{post}, \tau_B^{post})$ denote the output and transcript of this execution, respectively.
4. Output party B's view consisting of: $(x_B, r_B, \tau_B^{pre}, \tau_B^{post})$.

We wish to show that this view is indistinguishable from the following hybrid view where we invoke the simulator Sim_B^{post} guaranteed by the 2PC security of Π_{post} for party B:

$H_1(x_A, x_B, \lambda)$:

1. Emulate the same interactive protocol between A and B for computing $z_A, z_B = z(x_A, x_B)$ as in the real execution, sampling random coins r_A^{pre}, r_B^{pre} as necessary. Let $\tau_{pre} = (\tau_A^{pre}, \tau_B^{pre})$ denote the transcript of this preamble.
2. Letting $(y_A, y_B) = f_{post}(z_A, z_B)$, compute $(z_B, r_B^{post}, \widetilde{\tau}_B^{post}) \leftarrow \mathsf{Sim}_B^{post}(1^\lambda, z_B, y_B)$.
3. Output party B's hybrid view consisting of $x_B, (r_B^{pre} \| r_B^{post}), \tau_B^{pre}, \widetilde{\tau}_B^{post}$.

Claim 5. $\mathsf{view}_B^{\Pi}(x_A, x_B, \lambda) \stackrel{c}{\approx} H_1(x_A, x_B, \lambda)$

We present two proofs, where both construct essentially the same non-uniform reduction. The first proof carefully unpacks the formal definitions of non-uniformity and computational indistinguishability (right down to quantifiers), but results in an annoyingly verbose argument. The second proof is less formal but significantly shorter and easier to read; much of the awkwardness of the first proof is avoided by re-interpreting the "rules of the game" for adversaries in a reduction proof of the kind we are dealing with here. Specifically, in the second proof we allow adversaries to submit ensemble indices to their outside challengers.

Verbose Proof of Claim 5. Suppose the two distributions are not computationally indistinguishable. Then (unpacking definitions and quantifiers, including that of negligible function), there

exists a non-uniform poly-time distinguisher D_H, and a polynomial $p(\cdot)$, such that for every $N \in \mathbb{N}$ there exists a "bad" $\lambda^* > N$ and some "bad" x_A^*, x_B^* such that

$$\left|\Pr\left[D_H(1^{\lambda^*}, \mathsf{view}_B^{\Pi}(x_A^*, x_B^*, \lambda^*), x_A^*, x_B^*)\right] - \Pr\left[D_H(1^{\lambda^*}, H_1(x_A^*, x_B^*, \lambda^*), x_A^*, x_B^*)\right]\right| \geq 1/p(\lambda^*)$$

We use this D_H to construct the following non-uniform adversary D_{2pc} against the semi-honest security of Π_{post} as follows.

D_{2pc} parses its input as $(1^\lambda, ((z_B, r_B^{post}, \hat{\tau}_B^{post}), (y_A, y_B)), z_A, z_B)$. If λ is one of those bad λ^* for which there exist some bad (x_A^*, x_B^*) on which D_H distinguishes with non-negligible advantage, then D_{2pc} receives this bad pair as non-uniform advice, along with some $(\tau_B^{pre*}, r_B^{pre}, z_A^*, z_B^*)$ distributed according to $z(x_A^*, x_B^*; r_A^{pre} \| r_B^{pre})$ (over random choice of r_A^{pre})[2]. Otherwise, it receives an empty string as advice. If D_{2pc} has nothing written on its advice tape, or if the (z_A, z_B) parsed from its input does not equal the (z_A^*, z_B^*) written on its advice tape, then D_{2pc} outputs a random bit[3]. Else, it must be that $(z_A, z_B) = (z_A^*, z_B^*)$. In this case, D_{2pc} queries and outputs $b \leftarrow D_H(1^{\lambda^*}, (x_B^*, (r_B^{pre} \| r_B^{post}), \tau_B^{pre*}, \hat{\tau}_B^{post}), x_A^*, x_B^*)$.

Let $(x_A^*, x_B^*, \tau_B^{pre*} z_A^*, z_B^*)$ be an advice string for D_{2pc} on input a bad λ^*. See that if D_{2pc}'s input is distributed according to $(\mathsf{view}_B^{\Pi_{post}}, \mathsf{out}^{\Pi_{post}})(z_A^*, z_B^*, \lambda^*)$, then D_H's input is distributed according to $\mathsf{view}_B^{\Pi}(x_A^*, x_B^*, \lambda^*)$. Else, if D_{2pc}'s input is distributed according to $(\mathsf{Sim}_B^{post}(1^{\lambda^*}, z_B^*, y_B), (y_A, y_B))$ for $(y_A, y_B) \leftarrow f_{post}(z_A^*, z_B^*)$, then D_H's input is distributed according to $H_1(x_A^*, x_B^*, \lambda^*)$. The claim follows, as we have just shown that for every $N \in \mathbb{N}$ there exists a $\lambda^* > N$ and some z_A^*, z_B^* such that the distinguishing advantage of D_{2pc} between $(\mathsf{view}_B^{\Pi_{post}}, \mathsf{out}^{\Pi_{post}})(z_A^*, z_B^*, \lambda^*)$ and $(\mathsf{Sim}_B^{post}(1^{\lambda^*}, z_B^*, y_B), (y_A, y_B))$ for $(y_A, y_B) \leftarrow f_{post}(z_A^*, z_B^*)$ is non-negligible. □

Observe that much of the verbosity of the above proof concerns itself with boilerplate for getting the reduction to only play against challenges of its choosing: it basically "gives up" on any challenge that does not agree with its non-uniform advice. Thus, in this specific situation where adversaries only need to show indistinguishability for an infinite subsequence of "bad" ensemble indices that they receive non-uniform advice for, we can take the liberty of re-interpreting the rules of the reduction proof such that adversaries can submit desired ensemble indices to their outside challengers. This significantly cleans up the proof while preserving the main idea of the reduction, and not sacrificing too much rigor.

Shorter Proof. Let D_H be a (non-uniform) poly-time distinguisher between the hybrids. We use this D_H to construct a (non-uniform) adversary D_{2pc} against the semi-honest security of Π_{post} as follows. $D_{2pc}(1^\lambda)$ queries $(x_A, x_B) \leftarrow D_H(1^\lambda)$ and computes $\tau_B^{pre}, r_B^{pre}, z_A, z_B$ distributed according to $z(x_A, x_B; r_A^{pre} \| r_B^{pre})$ (over random choice of r_A^{pre}). It then submits z_A, z_B to the challenger, and parses the received challenge as $((z_B, r_B^{post}, \hat{\tau}_B^{post}), (y_A, y_B))$. Letting $r_B = r_B^{pre} \| r_B^{post}$, D_{2pc} queries and outputs $b \leftarrow D_H(x_B, r_B, \tau^{pre}, \hat{\tau}^{post})$.

See that if $\hat{\tau}^{post}$ is belongs to a real view of Π_{post}, then D_H's input is distributed according to view_B^{Π}. If $\hat{\tau}^{post}$ belongs to a simulated view, then D_H's input is distributed according to H_1. Thus, if D_H distinguishes with non-negligible probability, so does D_{2pc}. □

References

[Gol01] Oded Goldreich. *Foundations of cryptography: volume 1, basic tools*, volume 1. Cambridge university press, 2001.

[2]Giving the reduction $(\tau_B^{pre*}, r_B^{pre}, z_A^*, z_B^*)$ as advice is only a convenience, as it could sample these on its own if necessary from (x_A^*, x_B^*).

[3]Or it outputs a garbage bit, or it aborts. Intuitively, this is because our proof only needs D_{2pc} to "win" against its challenger for some infinite subsequence of bad ensemble indices $(z_A^*, z_B^*, \lambda^*)_{N \in \mathbb{N}}$ that it receives advice for. That is, we only care about the behavior of D_{2pc} on the bad challenges it is rigged to win against.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 99–108, 2011.

[Lin17] Yehuda Lindell. How to simulate it - A tutorial on the simulation proof technique. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 277–346. Springer International Publishing, 2017.