Round-Efficient Black-Box Construction of Composable Multi-Party Computation

Susumu Kiyoshima

NTT Secure Platform Laboratories, Japan.
kiyoshima.susumu@lab.ntt.co.jp

Abstract

We present a round-efficient black-box construction of a general MPC protocol that satisfies composability in the plain model. The security of our protocol is proven in angel-based UC framework under the minimal assumption of the existence of semi-honest oblivious transfer protocols. When the round complexity of the underlying oblivious transfer protocol is $r_{\mathrm{OT}}(n)$, the round complexity of our protocol is $\max(\widetilde{O}(\log^2 n), O(r_{\mathrm{OT}}(n)))$. Since constant-round semi-honest oblivious transfer protocols can be constructed under standard assumptions (such as the existence of enhanced trapdoor permutations), our result gives $\widetilde{O}(\log^2 n)$-round protocol under these assumptions. Previously, only an $O(\max(n^{\epsilon}, r_{\mathrm{OT}}(n)))$-round protocol was shown, where $\epsilon > 0$ is an arbitrary constant.

We obtain our MPC protocol by constructing a $\widetilde{O}(\log^2 n)$-round CCA-secure commitment scheme in a black-box way under the assumption of the existence of one-way functions.

1 Introduction

Protocols for *secure multi-party computation* (MPC) enable mutually distrustful parties to compute a functionality without compromising the correctness of the outputs and the privacy of their inputs. In the seminal work of Goldreich et al. [GMW87], a general MPC protocol was constructed in a model with malicious adversaries and a dishonest majority.[1] (By "a general MPC protocol," we mean a protocol that can be used to securely compute any functionality.)

In this paper, we consider a *black-box construction* of a general MPC protocol that guarantees *composable security*. Before stating our result, we explain black-box constructions and composable security.

Black-Box Constructions.

A construction of a protocol is *black-box* if it uses the underlying cryptographic primitives only in a black-box way (that is, only through their input/output interfaces). In contrast, if a construction uses the codes of the underlying primitives, it is *non-black-box*.

As argued in [IKLP06], constructing black-box constructions is important for both theoretical and practical reasons. Theoretically, it is important because understanding whether

This is the full version of a paper that appears in CRYPTO 2014 [Kiy14].

[1] In the following, we consider only such a model.

non-black-box use of cryptographic primitives is necessary for a cryptographic task is of great interest. Practically, it is important because black-box constructions are typically more efficient than non-black-box ones in terms of both communication complexity and computational complexity. In fact, since known non-black-box constructions of general MPC protocols compute general NP reductions to execute zero-knowledge proofs (this is where the codes of the primitives are used), they are highly inefficient and hard to implement. Thus, constructing black-box constructions of general MPC protocols is an important step toward practical general MPC protocols.

Recently, a series of works studied black-box constructions of general MPC protocols. Ishai et al. [IKLP06] showed the first construction of a general MPC protocol that uses the underlying low-level primitives (such as enhanced trapdoor permutations and homomorphic public-key encryption schemes) in a black-box way. Combined with the subsequent work of Haitner [Hai08], which showed a black-box construction of a (malicious) oblivious transfer protocol based on a semi-honest oblivious transfer protocol, their work gives a black-box construction of a general MPC protocol based on a semi-honest oblivious transfer protocol [HIK$^+$11]. Subsequently, Wee [Wee10] reduced the round complexity of [IKLP06] to $O(\log^* n)$, and Goyal [Goy11] further reduced the round complexity to $O(1)$.

These black-box protocols are proven to be secure in the *stand-alone setting*. That is, the protocols of [IKLP06, Wee10, Goy11] are secure in the setting where a single instance of the protocol is executed at a time.

Composable Security.

Compared with the stand-alone setting, the *concurrent setting* is more general and realistic security notion. In the concurrent setting, many instances of many different protocols are concurrently executed in an arbitrary schedule. Thus, in the concurrent setting, adversaries can perform a coordinated attack in which adversaries choose messages in each instance based on the executions of the other instances.

As a strong and realistic security notion in the concurrent setting, Canetti [Can01] proposed *universally composable (UC) security*. The main advantage of UC security is *composability*, which guarantees that when we compose many UC-secure protocols, we can prove the security of the resultant protocol from the security of its components. Thus, UC security enables us to construct secure protocols in a modular way. Composability also guarantees that a protocol remains secure even when it is concurrently executed with any other protocols in any schedule. Thus, UC-secure protocols are secure in the concurrent setting. Canetti et al. [CLOS02] constructed a UC-secure general MPC protocol in the *common reference string (CRS) model* (i.e., in a model in which all parties are given a common public string that is chosen by a trusted third party). Black-box constructions of UC-secure general MPC protocols were shown in the $\mathcal{F}_{\mathrm{OT}}$-hybrid model [IPS08] and in the $\mathcal{F}_{\mathrm{COM}}$-hybrid model [CDSMW09] (i.e., in a model with the ideal oblivious transfer functionality and in a model with the ideal commitment functionality).

UC security, however, turned out to be too strong to achieve in the *plain model*. That is, even with non-black-box use of cryptographic primitives, we cannot construct UC-secure general MPC protocols in a model with no trusted setup [CF01, CKL03].

To achieve composable security in the plain model, Prabhakaran and Sahai [PS04] proposed a variant of UC security called *angel-based UC security*. Roughly speaking, angel-based UC security is the same as UC security except that the adversary and the simulator have access to an additional entity—the *angel*—that allows some judicious use of

super-polynomial-time resources. Although angel-based UC security is weaker than UC security, angel-based UC security guarantees meaningful security in many cases. (For example, angel-based UC security implies *super-polynomial-time simulation (SPS) security* [Pas03, BS05, GGJS12, PLV12]. In SPS security, we allow the simulator to run in super-polynomial time; thus SPS security guarantees that whatever an adversary can do in the real world can also be done in the ideal world *in super-polynomial time*.) Furthermore, it was proven that, like UC security, angel-based UC security guarantees composability. Prabhakaran and Sahai [PS04] presented a general MPC protocol that satisfies angel-based UC security in the plain model based on new assumptions. Subsequently, Malkin et al. [MMY06] constructed another general MPC protocol that satisfies angel-based UC security in the plain model based on a new number-theoretic assumption.

Recently, several works constructed general MPC protocols with angel-based UC security under standard assumptions. Canetti et al. [CLP10] constructed a polynomial-round general MPC protocol in angel-based UC security assuming the existence of enhanced trapdoor permutations. Subsequently, Lin [Lin11] and Goyal et al. [GLP$^+$12] reduced the round complexity to $\widetilde{O}(\log n)$ under the same assumption. They also showed that with enhanced trapdoor permutations that are secure against quasi-polynomial-time adversaries, the round complexity of their protocols can be reduced to $O(1)$.

The construction of these MPC protocols are non-black-box. That is, in the protocols of [CLP10, Lin11, GLP$^+$12], the underlying primitives are used in a non-black-box way.

Black-Box Constructions of Composable Protocols.

Lin and Pass [LP12] showed the first black-box construction of a general MPC protocol that guarantees composable security in the plain model. The security of their protocol is proven under angel-based UC security and based on the minimal assumption of the existence of semi-honest oblivious transfer (OT) protocols. The round complexity of their protocol is $O(\max(n^\epsilon, r_{\text{OT}}(n)))$, where $\epsilon > 0$ is an arbitrary constant and $r_{\text{OT}}(n)$ is the round complexity of the underlying semi-honest OT protocols. Thus, with enhanced trapdoor permutations (from which we can construct constant-round semi-honest OT protocols), their result gives an $O(n^\epsilon)$-round protocol. Subsequently, an $O(1)$-round protocol was constructed in [KMO14] from $O(1)$-round semi-honest OT protocols that are secure against quasi-polynomial-time adversaries and one-way functions that are secure against subexponential-time adversaries.

Summarizing the state-of-the-art, for composable protocols in the plain model, we have

- $\tilde{O}(\log n)$-round non-black-box constructions under a standard polynomial-time hardness assumption [Lin11, GLP$^+$12],
- a $O(n^\epsilon)$-round black-box construction under a standard polynomial-time hardness assumption [LP12], and
- $O(1)$-round black-box or non-black-box constructions under standard super-polynomial-time hardness assumptions [Lin11, GLP$^+$12, KMO14].

Thus, for composable protocols based on standard polynomial-time hardness assumptions, there exists a gap between the round complexity of the non-black-box protocols ($\tilde{O}(\log n)$ rounds [Lin11, GLP$^+$12]) and that of the black-box protocols ($O(n^\epsilon)$ rounds [LP12]). The following is therefore an important open question.

Does there exist a ***round-efficient*** *black-box construction of a general MPC protocol that guarantees composability in the plain model under polynomial-time hardness assumptions?*

1.1 Our Result

In this paper, we narrow the gap between the round complexity of black-box composable general MPC protocols and the round complexity of non-black-box ones.

Main Theorem (Informal). *Assume the existence of $r_{\mathrm{OT}}(n)$-round semi-honest oblivious transfer protocols. Then, there exists a $\max(\widetilde{O}(\log^2 n), O(r_{\mathrm{OT}}(n)))$-round black-box construction of a general MPC protocol satisfying angel-based UC security in the plain model.*

Recall that, assuming the existence of enhanced trapdoor permutations, we have a constant-round semi-honest OT protocol. Thus, under this assumption, our main theorem gives a $\widetilde{O}(\log^2 n)$-round protocol.

We prove our main theorem by constructing a $\widetilde{O}(\log^2 n)$-round black-box construction of a *CCA-secure commitment scheme* [CLP10, Lin11, LP12, GLP$^+$12, KMO14] from one-way functions.

Theorem (Informal). *Assume the existence of one-way functions. Then, there exists a $\widetilde{O}(\log^2 n)$-round black-box construction of a CCA-secure commitment scheme.*

Roughly speaking, a CCA-secure commitment scheme is a tag-based commitment scheme (i.e., a commitment scheme that takes an n-bit string—a *tag*—as an additional input) such that the hiding property holds even against adversaries that interact with the *committed-value oracle* during the interaction with the challenger. The committed-value oracle interacts with the adversary as an honest receiver in many concurrent sessions of the commit phase. At the end of each session, if the commitment of this session is invalid or has multiple committed values, the oracle returns $\perp$ to the adversary. Otherwise, the oracle returns the unique committed value to the adversary.

Lin and Pass [LP12] showed that in angel-based UC security, an $O(\max(r_{\mathrm{CCA}}(n), r_{\mathrm{OT}}(n)))$-round general MPC protocol can be obtained in a black-box way from a $r_{\mathrm{CCA}}(n)$-round CCA-secure commitment scheme and a $r_{\mathrm{OT}}(n)$-round semi-honest OT protocol. Thus, we can prove our main theorem by combining the above theorem with the result of [LP12].

1.2 Outline

In Section 2, we give an overview of our CCA secure commitment scheme. In Section 3, we give definitions that are used throughout the paper. In Section 4, we show the building blocks that are used in our CCA-secure commitment scheme. In Section 5, we show our CCA-secure commitment scheme and prove its security. In Section 6, we show our main theorem. In Appendix A, we explain a technical detail for the proof given in Section 5.1.1. In Appendix B, we prove a lemma that is used in Section 4.2.

2 Overview of Our CCA-Secure Commitment Scheme

Key elements for obtaining CCA-secure commitment schemes are *concurrent extractability* and *non-malleability*. With these elements, it can be shown that the committed-value oracle

is useless for breaking the hiding property. Non-malleability is used to show that the sessions between the adversary and the oracle are independent of the session between the adversary and the challenger. Concurrent extractability is used to show that the committed-value oracle can be emulated in polynomial time by extracting the committed values from the adversary.

Before constructing our CCA-secure commitment scheme, we first construct two building blocks: (i) a commitment scheme CECom' that is *concurrently extractable without over-extraction* and (ii) a *one-one CCA-secure* commitment scheme $\mathsf{CCACom}^{1:1}$. The former guarantees concurrent extractability and the latter guarantees (slightly strong) non-malleability.

2.1 Building Block 1: Concurrently Extractable Commitment Scheme without Over-Extraction

A commitment scheme is *concurrently extractable* if a rewinding extractor can extract the committed values from any committer even in the concurrent setting, and a concurrently extractable commitment scheme is *concurrently extractable without over-extraction* if the extractor outputs $\perp$ whenever the commitment is invalid.[2] (Basic extractability, in contrast, allows the extractor to output an arbitrary value when the commitment is invalid.) There exists a commitment scheme CECom that is concurrently extractable *with* over-extraction based on the existence of one-way functions [MOSV06].

To construct a commitment scheme that is concurrently extractable without over-extraction, we start from the following scheme (in which the cut-and-choose technique is used in the same way as in the previous works of black-box protocols [CDSMW08, CDSMW09, Wee10, LP12, KMO14]).

1. Let v be the value to be committed. Then, the committer computes an $(n+1)$-out-of-$10n$ Shamir's secret sharing $\boldsymbol{s} = (s_1, \ldots, s_{10n})$ of value v and commits to each s_j in parallel by using CECom.

2. The receiver sends a random subset $\Gamma \subset [10n]$ of size n.

3. The committer reveals s_j for every $j \in \Gamma$ and decommits the corresponding commitments.

4. The receiver accepts the commitment if and only if the decommitments of CECom are valid for every $j \in \Gamma$.

For $j \in [10n]$, let the j-th *column* be the j-th CECom commitment. The use of the cut-and-choose technique guarantees that when the receiver accepts a commitment, the CECom commitments are valid in "most" columns. Then, since we can extract the committed value of CECom whenever the CECom commitment is valid, we can extract s_j in most columns on an accepted commitment. We can therefore recover v from the extracted values of the CECom commitments by using the error-correcting property of Shamir's secret sharing scheme.[3]

Unfortunately, although the above scheme is concurrently extractable without over-extraction, we cannot prove its hiding property. This is because the receiver requests the committer to open adaptively-chosen CECom commitments (in other words, the receiver performs a selective opening attack).

[2] A commitment is *valid* if there exists a valid decommitment of this commitment; otherwise, it is *invalid*. A commitment is *accepted* if the receiver does not abort in the commit phase; otherwise, it is *rejected*.

[3] Recall that Shamir's secret sharing is also a codeword of Reed-Solomon code.

We therefore modify the scheme in the following way. At the beginning of the scheme, we let the receiver commit to Γ by using a statistically binding commitment scheme Com. Now, since the receiver no longer choose the subset adaptively, we can prove the hiding property by a standard technique. Furthermore, at first sight, the hiding property of Com seems to guarantee that the scheme remains to be concurrently extractable without over-extraction.

In the modified scheme, however, we cannot prove that the scheme is concurrently extractable without over-extraction. This is because we can no longer show that most of the CECom commitments are valid in an accepted commitment. Consider, for example, that there exists a cheating committer C^* such that receiving a Com commitment to Γ at the beginning, C^* somehow generates an invalid CECom commitment in the j-th column for every $j \notin \Gamma$ and commits to 0 in the j-th column for every $j \in \Gamma$. Then, although C^* seems to break the hiding property of Com, we do not know how to use C^* to break the hiding property of Com. To see this, observe the following. Recall that since CECom is an extractable commitment scheme *with* over-extraction, the extractor of CECom may output an arbitrary value when the CECom commitment is invalid. Thus, when we extract the committed values of CECom from $\mathcal{C}^*$, the extracted value may be 0 in every column. Hence, although $\mathcal{C}^*$ behaves differently in CECom based on the value of Γ, we cannot detect it.

To overcome this problem, we use the commitment scheme $\mathsf{wExtCom}$ that was introduced by Goyal et al. [GLOV12]. The commit phase of $\mathsf{wExtCom}$ consists of three stages: `commit`, `challenge`, and `reply`. In the `commit` stage, the committer commits to random $a_0, a_1 \in \{0,1\}^n$ such that $a_0 \oplus a_1 = v$; in the `challenge` stage, the receiver sends a random bit $ch \in \{0,1\}$; in the `reply` stage, the committer reveals a_{ch} and decommits the corresponding commitment. We note that $\mathsf{wExtCom}$ is extractable only in a weak sense—extractions may fail with probability at most $1/2$—but $\mathsf{wExtCom}$ is extractable without over-extraction. That is, the extractor may output $\perp$ with probability at most $1/2$, but when the extractor outputs $v \neq \perp$, the commitment is valid and its committed value is v. We also note that $\mathsf{wExtCom}$ satisfies the following property: For a fixed transcript of the `commit` stage, if a cheating committer returns a valid reply with probability $1/\mathsf{poly}(n)$ for both $ch = 0$ and $ch = 1$, then the committed value can be extracted with probability 1 in expected polynomial time.

With $\mathsf{wExtCom}$, we modify our scheme as follows: After committing to $\boldsymbol{s}$ with CECom, the committer commits to (s_j, d_j) for each $j \in [10n]$ in parallel with $\mathsf{wExtCom}$, where (s_j, d_j) is a decommitment of the j-th CECom commitment. Then, we show that in most columns on an accepted commitment, the $\mathsf{wExtCom}$ commitment is valid and its committed value is a valid decommitment of the corresponding CECom commitment. Toward this end, we observe the following.

- If a cheating committer generates an accepting commitment with non-negligible probability, then in $\mathsf{wExtCom}$ of more than $9n$ columns, the cheating committer returns a valid reply with non-negligible probability for both $ch = 0$ and $ch = 1$. (If the cheating committer returns a valid reply with non-negligible probability for both $ch = 0$ and $ch = 1$ in $\mathsf{wExtCom}$ of at most $9n$ columns, then there are n columns in which the $\mathsf{wExtCom}$ commitment is accepted with probability at most $1/2 + \mathsf{negl}(n)$. Thus, the probability that all $\mathsf{wExtCom}$ commitments are accepted is negligible, and therefore the commitment is accepted with at most negligible probability.[4])

- Thus, from the property of $\mathsf{wExtCom}$, we can extract the committed values of $\mathsf{wExtCom}$

[4]The formal proof is more complicated because the $\mathsf{wExtCom}$ commitments are executed in parallel and thus the columns are not independent of each other.

without over-extraction in most columns.

- Then, from the property of the cut-and-choose technique, we can show that in most columns of an accepted commitment, the $\mathsf{wExtCom}$ commitment is valid and its committed value is a valid decommitment of the corresponding CECom commitment. Note that since the committed values of $\mathsf{wExtCom}$ commitments can be extracted without over-extraction, we can show that the cheating committer cannot give invalid $\mathsf{wExtCom}$ commitments in many columns.

Then, since this implies that most of the CECom commitments are valid whenever the commitment is accepted, we can extract the committed value of the scheme without over-extraction as before, i.e., by extracting the committed values of CECom commitments and using the error-collecting property of Shamir's secret sharing scheme.

2.2 Building Block 2: One-One CCA-Secure Commitment Scheme

A *one-one CCA-secure commitment scheme*, which is closely related to a *non-malleable commitment scheme*, is one that is CCA secure w.r.t. a restricted class of adversaries that execute only a single session with the committed-value oracle and immediately receive the answer from the oracle at the end of the session.[5]

We construct a black-box $O(\log n)$-round one-one CCA-secure commitment scheme by simplifying the CCA-secure commitment scheme of [LP12] and using the *DDN* $\log n$ *trick* [DDN00, LPV08], which transforms a concurrent non-malleable commitment scheme for tags of length $O(\log n)$ to a non-malleable commitment scheme for tags of length $O(n)$ without increasing the round complexity. In the following, we assume the familiarity to the scheme of [LP12]. Roughly speaking, the scheme of [LP12] consists of polynomially-many *rows*—each row is a parallel execution of (a part of) the trapdoor commitment scheme of [PW09]—and a cut-and-choose phase, which forces the committer to give valid and consistent trapdoor commitments in every row. If we reduce the number of rows from $\mathsf{poly}(n)$ to $\ell(n)$ in the scheme of [LP12], where $\ell(n)$ is the length of the tags, the resultant scheme is no longer CCA secure. It is easy to verify, however, that the scheme is *parallel CCA secure*, i.e., it is CCA secure w.r.t. a restricted class of adversaries that give a single parallel query to the oracle and receive the answers immediately. (This is because when the adversaries give only a single parallel query, the recursive rewinding does not occur in the extraction and thus we require only a single rewinding opportunity.) Then, we set $\ell(n) := O(\log n)$ and apply the DDN $\log n$ trick to the above parallel CCA-secure commitment scheme. It is not hard to see that the resultant scheme is one-one CCA secure.

2.3 CCA-Secure Commitment Scheme from the Building Blocks

Given CECom' and $\mathsf{CCACom}^{1:1}$, we construct a CCA-secure commitment scheme CCACom roughly as follows, where the committer commits to a value v with tag tag.

1. The receiver commits to a random subset $\Gamma \subset [10n]$ of size n by using $\mathsf{CCACom}^{1:1}$ with tag tag.

[5] In contrast, a non-malleable commitment scheme is one that is CCA secure w.r.t. a restricted class of adversaries that execute a single session with the oracle and receive the answer *after completing the interactions with the challenger and the oracle.*

2. The committer computes an $(n+1)$-out-of-$10n$ Shamir's secret sharing $\boldsymbol{s} = (s_1, \ldots, s_{10n})$ of value v and commits to each s_j in parallel by using a statistically binding commitment scheme Com.

3. For $\eta(n) := r_{\mathrm{CEC}}(n) + 1$ times in sequence (where $r_{\mathrm{CEC}}(n)$ is the round complexity of CECom'), the committer does the following: the committer commits to s_j for every $j \in [10n]$ by using CECom' in parallel. Each parallel commitment is called a *row*.

4. The receiver decommits the commitment of the first step and reveals Γ.

5. For every $j \in \Gamma$, the committer decommits all of the $\eta(n)$ commitments whose committed values are s_j.

Our scheme differs from the previous CCA-secure commitment schemes [CLP10, LP12, Lin11, GLP$^+$12] in that it uses a one-one CCA-secure commitment scheme instead of a non-malleable commitment scheme; furthermore, our scheme uses a one-one CCA-secure commitment scheme in the reverse order. That is, whereas the previous schemes (implicitly or explicitly) use non-malleable commitment schemes from the committer to the receiver, our scheme uses a one-one CCA secure commitment scheme from the receiver to the committer. (Very recently, the same strategy is used in [KMO14].)

Using a one-one CCA-secure commitment scheme in the reverse order is crucial in showing the *simulation-soundness* of the cut-and-choose phase. We say that the adversary (or the challenger) *cheats* if in an accepted commitment there exists a row whose committed shares disagree with $\boldsymbol{s}$ in more than n indexes. Using the one-one CCA security of $\mathsf{CCACom}^{1:1}$, we can show that the adversary cannot cheat in every session of the *right interaction* (i.e., the interaction between the adversary and the oracle) even when the adversary receives a commitment in which the challenger cheats in the *left interaction* (i.e., the interaction between the adversary and the challenger). Roughly speaking, this is because the adversary can emulate the cheating challenger in polynomial time by making a single query to the committed-value oracle of $\mathsf{CCACom}^{1:1}$ and receiving Γ; therefore, from one-one CCA security of $\mathsf{CCACom}^{1:1}$, the commitment that the adversary receives on the left is useless for breaking the hiding property of $\mathsf{CCACom}^{1:1}$ on the right, and thus the adversary cannot cheat on the right from the property of the cut-and-choose technique. Note that non-malleability is insufficient for this argument since the hiding property of $\mathsf{CCACom}^{1:1}$ need to hold even when the adversary receives the answer from the oracle immediately after completing the query to the oracle. We also note that CECom' must be concurrently extractable *without* over-extraction since otherwise the adversary may give invalid commitments in more than n indexes without being detected in the cut-and-choose phase. (As explained in Section 2.1, the existence of such an adversary does not contradict the one-one CCA security of $\mathsf{CCACom}^{1:1}$ if over-extraction can occur.)

Given the simulation-soundness of the cut-and-choose phase, we can show the CCA security of CCACom by, as in the analysis of previous CCA-secure commitment schemes [CLP10, Lin11, LP12], rewinding the adversary and emulating the committed-value oracle in polynomial time. Toward this end, we consider a series of hybrid experiments in which the commitment that the adversary receives on the left is gradually changed as follows: In the i-th hybrid experiment ($i \in [\eta(n)]$), we switch the committed value from s_j to 0 for every $j \notin \Gamma$ in the i-th row, where Γ is extracted by brute force. Note that the $(i-1)$-st hybrid and the i-th hybrid differ only in the i-th row. The problem is that the adversary accesses

the committed-value oracle, which runs in super-polynomial time. Then, to show the indistinguishability between the $(i-1)$-st hybrid and the i-th hybrid, we observe the following. Since there are $r_{\mathrm{CEC}}+1$ rows (in particular, the number of rows is bigger than the number of rounds in CECom'), we can extract the committed shares in a row on every right session without disturbing the hiding property of CECom' in the i-th row on the left. (Here, we use a technique used in [LP11]. Roughly speaking, we extract the committed shares from a row that contains no message of the CECom' commitment of the i-th row on the left.) Recall that, since CECom' is concurrently extractable without over-extraction, we can extract the committed shares without over-extraction. Then, since the simulation-soundness guarantees that these shares agree with $\boldsymbol{s}$ in at least $9n$ indexes, we can compute v from these shares by using the error-correcting property of Shamir's secret sharing. Therefore we can emulate the oracle in polynomial time by rewinding the adversary (without disturbing the hiding property of CECom' in the i-th row) and computing v as above. Thus, the indistinguishability of the $(i-1)$-st hybrid and the i-th hybrid follows from the hiding property of CECom'. Then, we consider another hybrid experiment: This experiment is the same as the $\eta(n)$-th hybrid except that the committed value of the j-th Com commitment in Step 2 is switched from s_j to 0 for every $j \notin \Gamma$. From the same argument as above, this hybrid is indistinguishable from the $\eta(n)$-th hybrid. Then, since in this hybrid the adversary does not receive any information about v, the CCA security follows.

We note that the formal proof is more complicated. For example, we need to show the simulation-soundness even for the adversary accessing the committed-value oracle. To solve this problem, we increase the number of rows (i.e., $\eta(n)$) and emulate the oracle in polynomial time without disturbing the one-one CCA security of $\mathsf{CCACom}^{1:1}$. To show that the oracle can be emulated, we require the simulation soundness; thus, there seems to be a circular argument, i.e., we require the simulation soundness to show the simulation soundness. In the formal analysis, we show that this issue can be avoided.

Comparison with the CCA-secure commitment scheme of [KMO14]. The above CCA-secure commitment scheme is based on the CCA-secure commitment scheme of [KMO14], which is constructed from one-way functions that are secure against subexponential-time adversaries. The scheme of [KMO14] is the same as the above scheme except for the following.

- There is only a single row, and CECom is used instead of CECom' (i.e., a concurrently extractable scheme *with* over-extraction is used).
- The underlying commitment schemes Com, CECom, and $\mathsf{CCACom}^{1:1}$ are secure against subexponential-time adversaries. In particular, Com is hiding against T_1-time adversaries but is completely broken in time $o(T_2)$, CECom is hiding against T_2-time adversaries but is completely broken in time $o(T_3)$, and $\mathsf{CCACom}^{1:1}$ is one-one CCA secure against T_3-time adversaries, where (T_1, T_2, T_3) is a hierarchy of running times such that $T_3 \gg T_2 \gg T_1 \gg n^{\omega(1)}$. This is where subexponentially hard one-way functions are required.

The high-level strategy for proving CCA security is the same, i.e., showing the simulation soundness from one-one CCA security of $\mathsf{CCACom}^{1:1}$ and then considering hybrid experiments in which committed values of CECom and Com are gradually switched. The proof of [KMO14] is, however, different from ours in the following.

- In the proof of the simulation soundness, the issue of over-extraction is solved by extracting the committed values of CECom by brute force. (Even when the committed

values of CECom are extracted by brute force, the one-one CCA security of $\mathsf{CCACom}^{1:1}$ still holds since the committed values of CECom are extractable in time $o(T_3)$ and one-one CCA security of $\mathsf{CCACom}^{1:1}$ holds against T_3-time adversaries.)

- When the committed values of CECom are switched, the indistinguishability follows immediately from the fact that CECom is hiding against T_2-time adversaries and the running time of the committed-value oracle is $o(T_2)$. (The committed-value oracle computes its output by extracting the committed values of Com by brute force. Thus, its running-time is $o(T_2)$.)

Thus, the proof of [KMO14] heavily depends on the subexponentially hard security of the underlying commitment schemes. Roughly speaking, we weaken the assumption of [KMO14] by doing the following.

- To show the simulation soundness without subexponentially hard security, we replace CECom with CECom', which is concurrently extractable *without* over-extraction.
- To show the indistinguishability when we switch the committed values of CECom', we increase the number of rows so that the committed-value oracle can be emulated in polynomial time by rewinding the adversary while preserving the hiding property of CECom'.

Overall, despite of the similarity of the high-level structure between the scheme of [KMO14] and ours, the details of the security proofs have a lot of difference.

3 Preliminaries

Throughout the paper, we use n to denote the security parameter. For any $k \in \mathbb{N}$, let $[k] \stackrel{\text{def}}{=} \{1, 2, \ldots, k\}$.

3.1 Shamir's Secret Sharing

We recall Shamir's secret sharing scheme. To compute an $(k+1)$-out-of-m secret sharing $\boldsymbol{s} = (s_1, \ldots, s_m)$ of a value $v \in GF(2^n)$, we choose random $a_1, \ldots, a_k \in GF(2^n)$, let $p(z) \stackrel{\text{def}}{=} v + a_1 z + \cdots + a_k z^k$, and set $s_i := p(i)$ for each $i \in [m]$. For any positive real number $x \leq 1$ and any $\boldsymbol{s} = (s_1, \ldots, s_m)$ and $\boldsymbol{s'} = (s'_1, \ldots, s'_m)$, we say that $\boldsymbol{s}$ and $\boldsymbol{s'}$ are *x-close* if $|\{i \in [m] \mid s_i = s'_i\}| \geq x \cdot m$. If $\boldsymbol{s}$ and $\boldsymbol{s'}$ are not x-close, then we say that they are *$(1-x)$-far*. We note that $(k+1)$-out-of-m Shamir's secret sharing is a codeword of the Reed-Solomon code with minimum relative distance $(m-k)/m$. Thus, for any $\boldsymbol{s}$ that is $(1-(m-k)/2m)$-close to a valid codeword $\boldsymbol{w}$, we can compute $\boldsymbol{w}$ from $\boldsymbol{s}$.

3.2 Commitment Schemes

Recall that commitment schemes are two-party protocols between the committer C and the receiver R. Without stated otherwise, all commitment schemes in this paper are statistically binding and computationally hiding. A transcript of the commit phase is *accepted* if R does not abort in the commit phase. A transcript of the commit phase is *valid* if there exists a valid decommitment of this transcript. In this paper, we define the committed value of an invalid commitment as $\perp$.

There exist a 2-round statistically binding commitment scheme based on one-way functions [Nao91], which uses the underlying one-way function in a black-box way.

Strong Computational Binding Property. We say that a commitment scheme $\langle C, R\rangle$ satisfies the *strong computational binding property* if any PPT committer C^* can generate a commitment that has more than one committed value with at most negligible probability in the interaction with the honest receiver R.[6]

3.3 Extractable Commitment Schemes

We recall the definition of *extractable commitment schemes* from [PW09]. Roughly speaking, a commitment scheme is *extractable* if there exists an expected polynomial-time oracle machine (called an *extractor*) E such that for any committer C^* that generates a commitment, E^{C^*} extracts the committed value when the commitment is valid. We note that when the commitment is invalid, E can output an arbitrary garbage value. (This is called *over-extraction*.)

Formally, extractable commitment schemes are defined as follows. A commitment scheme $\langle C, R\rangle$ is *extractable* if there exists an expected polynomial-time probabilistic extractor E such that for any PPT committer $\mathcal{C}^*$, extractor $E^{\mathcal{C}^*}$ outputs a pair (τ, σ) such that

- τ is identically distributed with the view of C^* interacting with honest receiver R in the commit phase.
- If τ is accepted, then $\sigma \neq \perp$ except with negligible probability.
- If $\sigma \neq \perp$, then it is statistically impossible to decommit τ to any value other than σ.

There exists a 4-round extractable commitment ExtCom based on one-way functions [PW09], which uses the underlying one-way function in a black-box way.

Weakly Extractable Commitment Schemes. A commitment scheme $\langle C, R\rangle$ is *weakly extractable* if there exists an expected polynomial-time probabilistic extractor E such that for any PPT committer $\mathcal{C}^*$, extractor $E^{\mathcal{C}^*}$ outputs a pair (τ, σ) such that

- τ is identically distributed with the view of C^* interacting with honest receiver R in the commit phase.
- The probability that τ is accepted and $\sigma = \perp$ is at most $1/2$.
- If $\sigma \neq \perp$, then it is statistically impossible to decommit τ to any value other than σ.

Goyal et al. [GLOV12] showed that the commitment scheme $\mathsf{wExtCom}$ in Figure 1 is weakly extractable. We note that given two accepted transcripts of $\mathsf{wExtCom}$ such that `commit` stage is identical but `challenge` stage is different, we can extract the committed value.

[6] The standard computational binding property guarantees only that for any PPT committer C^*, the commitment that C^* generates cannot be decommitted to more than one value *in polynomial time*. Thus, this commitment may have more than one committed value.

Commit Phase

The committer C and the receiver R receive common inputs 1^n. To commit to $v \in \{0,1\}^n$, the committer C does the following with the receiver R.

`commit` **stage.** C chooses a pair of random n-bit strings (a_0, a_1) such that $a_0 \oplus a_1 = v$. Then, C commits to a_0 and a_1 by using Com. For each $b \in \{0,1\}$, let c_b be the commitment to a_b.

`challenge` **stage.** The receiver R sends a random bit $e \in \{0,1\}$ to C.

`reply` **stage.** C decommits c_e to a_e.

Decommit Phase

To decommit, C sends v to R and decommits c_0 and c_1. Then, R checks whether $a_0 \oplus a_1 = v$.

Figure 1: Weakly extractable commitment scheme $\mathsf{wExtCom}$ [GLOV12].

3.4 Concurrently Extractable Commitment Schemes

Roughly speaking, a commitment scheme is *concurrently extractable* if in the concurrent setting a rewinding extractor can simulate the view of any malicious committer and extract the committed values in the view.

Micciancio et al. [MOSV06] showed a $\widetilde{O}(\log n)$-round concurrently extractable commitment CECom, which is an abstraction of the preamble stage of the concurrent zero-knowledge protocol of [PRS02]. In [MOSV06], the rewinding extractor uses the rewinding strategy of [PRS02]. In this paper, we instead use the rewinding strategy of [PTV12]. This is because the analysis of [PTV12] is more general than that of [PRS02], and therefore helpful when we use CECom as a building block. (From the same reason, several works (e.g., [LPTV10, LP11]) use the rewinding strategy of [PTV12] instead of that of [PRS02].) In the following, we use "PTV rewinding strategy" to denote the rewinding strategy of [PTV12]. In PTV rewinding strategy (as well as the rewinding strategy of of [PRS02]), the extractor internally runs C^* and the honest receivers, and computes a sequence of "threads of execution." Each thread consists of the views of all the parties, and satisfies the following properties.

- Each thread is a perfect simulation of a prefix of an actual execution.
- The last thread, called the *main thread*, is a perfect simulation of a complete execution. Any other thread is called a *look-ahead thread*. The extractor outputs the view of C^* in the main thread.
- Any two threads share a (possibly empty) prefix, but they are independently simulated after the shared prefix.

Furthermore, the extractor runs in strict polynomial time, and except with negligible probability it extracts the committed value of any session on any thread *immediately after* the session is completed.

3.5 Trapdoor Commitment Schemes

Roughly speaking, *trapdoor commitment schemes* [PW09] are commitment schemes such that there exists a simulator that can generate a simulated commitment and can later decommit it to any value. Pass and Wee [PW09] showed that the black-box scheme TrapCom in Figure 2 is a trapdoor bit commitment. TrapCom is not statistically binding, but it satisfies the strong computational binding property. This follows from the fact that if C^* generates a commitment that can decommit to both 0 and 1, we can compute the committed value e of Com. Pass and Wee also showed that by running TrapCom in parallel, we obtain a black-box trapdoor commitment scheme PTrapCom for multiple bits. PTrapCom also satisfies the strong computational binding property.

Commit Phase

To commit to $\sigma \in \{0,1\}$ on common input 1^n, the committer C does the following with the receiver R:

Step 1. R chooses a random n-bit string $e = (e_1, \ldots, e_n)$ and commits to e by using Com.

Step 2. For each $i \in [n]$, the committer C chooses a random $\eta_i \in \{0,1\}$ and sets

$$v_i := \begin{pmatrix} v_i^{00} & v_i^{01} \\ v_i^{10} & v_i^{11} \end{pmatrix} = \begin{pmatrix} \eta_i & \eta_i \\ \sigma \oplus \eta_i & \sigma \oplus \eta_i \end{pmatrix} .$$

Then, for each $i \in [n]$, $\alpha \in \{0,1\}$, and $\beta \in \{0,1\}$ in parallel, C commits to $v_i^{\alpha\beta}$ by using ExtCom; let $(v_i^{\alpha\beta}, d_i^{\alpha\beta})$ be the corresponding decommitment.

Step 3. R decommits the Step 1 commitment to e.

Step 4. For each $i \in [n]$, C sends $(v_i^{e_i 0}, d_i^{e_i 0})$ and $(v_i^{e_i 1}, d_i^{e_i 1})$ to R. Then, R checks whether these are valid decommitments and whether $v_i^{e_i 0} = v_i^{e_i 1}$.

Decommit Phase

To decommit, C sends σ and random $\gamma \in \{0,1\}$ to R. In addition, for every $i \in [n]$, C sends $(v_i^{0\gamma}, d_i^{0\gamma})$ and $(v_i^{1\gamma}, d_i^{1\gamma})$ to R. Then, R checks whether $(v_i^{0\gamma}, d_i^{0\gamma})$ and $(v_i^{1\gamma}, d_i^{1\gamma})$ are valid decommitments and whether $v_0^{0\gamma} \oplus v_0^{1\gamma} = \cdots = v_n^{0\gamma} \oplus v_n^{1\gamma} = \sigma$.

Figure 2: Black-box trapdoor bit commitment Scheme TrapCom.

3.6 CCA-Secure Commitment Schemes

We recall the definition of CCA security and κ-robustness [CLP10, LP12].

CCA Security (w.r.t. the Committed-Value Oracle)

Roughly speaking, a tag-based commitment scheme $\langle C, R \rangle$ is *CCA-secure* if the hiding property of $\langle C, R \rangle$ holds even against adversary $\mathcal{A}$ that interacts with the *committed-value oracle*

during the interaction with the committer. The committed-value oracle $\mathcal{O}$ interacts with $\mathcal{A}$ as an honest receiver in many concurrent sessions of the commit phase of $\langle C, R\rangle$ using tags chosen adaptively by $\mathcal{A}$. At the end of each session, if the commitment of this session is invalid or has multiple committed values, $\mathcal{O}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{O}$ returns the unique committed value to $\mathcal{A}$.

More precisely, let us consider the following probabilistic experiment $\mathrm{IND}_b(\langle C, R\rangle, \mathcal{A}, n, z)$ for each $b \in \{0,1\}$. On input 1^n and auxiliary input z, adversary $\mathcal{A}^{\mathcal{O}}$ adaptively chooses a pair of challenge values $v_0, v_1 \in \{0,1\}^n$ and an n-bit tag $\mathsf{tag} \in \{0,1\}^n$. Then, $\mathcal{A}^{\mathcal{O}}$ receives a commitment to v_b with tag tag from the challenger. Let y be the output of $\mathcal{A}$. The output of the experiment is $\perp$ if during the experiment, $\mathcal{A}$ sends $\mathcal{O}$ any commitment using tag tag. Otherwise, the output of the experiment is y. Let $\mathsf{IND}_b(\langle C, R\rangle, \mathcal{A}, n, z)$ denote the output of experiment $\mathrm{IND}_b(\langle C, R\rangle, \mathcal{A}, n, z)$.

Definition 1. Let $\langle C, R\rangle$ be a tag-based commitment scheme and $\mathcal{O}$ be the committed-value oracle of $\langle C, R\rangle$. Then, $\langle C, R\rangle$ is *CCA-secure (w.r.t the committed-value oracle)* if for any PPT adversary $\mathcal{A}$, the following are computationally indistinguishable:

- $\{\mathsf{IND}_0(\langle C, R\rangle, \mathcal{A}, n, z)\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$
- $\{\mathsf{IND}_1(\langle C, R\rangle, \mathcal{A}, n, z)\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$

The *left session* is the session of the commit phase between the challenger and $\mathcal{A}$, and *right sessions* are the sessions between $\mathcal{A}$ and $\mathcal{O}$. $\Diamond$

We say a commitment scheme is *one-one CCA-secure* if it is CCA secure w.r.t. a restricted class of adversaries that start only a single right session.

κ-Robustness (w.r.t. the Committed-Value Oracle)

Roughly speaking, a tag-based commitment scheme is *κ-robust* if for any adversary $\mathcal{A}$ and any ITM B, the joint output of a κ-round interaction between $\mathcal{A}^{\mathcal{O}}$ and B can be simulated without $\mathcal{O}$ by a PPT simulator. Thus, the κ-robustness guarantees that the committed-value oracle is useless for attacking any κ-round protocol.

Definition 2. Let $\langle C, R\rangle$ be a tag-based commitment scheme and $\mathcal{O}$ be the committed-value oracle of $\langle C, R\rangle$. For any constant $\kappa \in \mathbb{N}$, we say that $\langle C, R\rangle$ is *κ-robust (w.r.t. the committed value oracle)* if there exists a PPT oracle machine (called *simulator*) $\mathcal{S}$ such that for any PPT adversary $\mathcal{A}$ and any κ-round PPT ITM B, the following are computationally indistinguishable:

- $\left\{\mathsf{out}_{B,\mathcal{A}^{\mathcal{O}}}[\langle B(y), \mathcal{A}^{\mathcal{O}}(z)\rangle(1^n, x)]\right\}_{n\in\mathbb{N}, x,y,z\in\{0,1\}^n}$
- $\left\{\mathsf{out}_{B,\mathcal{S}^{\mathcal{A}}}[\langle B(y), \mathcal{S}^{\mathcal{A}}(z)\rangle(1^n, x)]\right\}_{n\in\mathbb{N}, x,y,z\in\{0,1\}^n}$

Here, for any ITM A and B, we use $\mathsf{out}_{A,B}[\langle A(y), B(z)\rangle(x)]$ to denote the joint output of A and B in an interaction between them on inputs x, y to A and x, z to B respectively. If $\langle C, R\rangle$ is κ-robust for any constant κ, we say that $\langle C, R\rangle$ is *robust*. $\Diamond$

4 Building Blocks

In this section, we construct (i) a commitment scheme that is concurrently extractable without over-extraction and (ii) a one-one CCA-secure commitment scheme. Both schemes are used in our $\widetilde{O}(\log^2 n)$-round CCA-secure commitment scheme in Section 5.

Commit Phase

To commit to $\sigma \in \{0,1\}^n$, the committer C does the following with the receiver R.

Step 1. R commits to a random sublet $\Gamma \subset [40n]$ of size n by using Com.

Step 2. C computes an $(n+1)$-out-of-$40n$ Shamir's secret sharing $\boldsymbol{s} = (s_1, \ldots, s_{40n})$ of value σ. Then, for each $j \in [40n]$ in parallel, C commits to s_j by using CECom. Let (s_j, d_j) be the decommitment of the j-th commitment.

Step 3. For each $j \in [40n]$ in parallel, C commits to (s_j, d_j) by using $\mathsf{wExtCom}$.

Step 4. R decommits the Step 1 commitment to Γ.

Step 5. For each $j \in \Gamma$, C decommits the j-th Step 3 commitment to (s_j, d_j). Then, for each $j \in \Gamma$, R checks whether the decommitment is valid and whether the decommitted value (s_j, d_j) is a valid decommitment of the j-th Step 2 commitment.

Decommit Phase

C sends σ to R, and decommits all the Step 2 commitments. Then, R defines $\boldsymbol{s} = (s_1, \ldots, s_{40n})$ as follows: If the j-th decommitment is invalid, $s_j := \bot$; otherwise, s_j is the j-th decommitted value. Then, R checks the following:

- $\boldsymbol{s}$ is 0.9-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{40n})$.
- For each $j \in \Gamma$, w_j equals the value revealed in Step 5.

If both of these hold and $\boldsymbol{w}$ is a codeword corresponding to σ, then R accepts the decommitment. Otherwise, R rejects it.

Figure 3: A concurrently commitment scheme CECom'.

4.1 Concurrently Extractable Commitment Scheme without Over-Extraction

Using one-way functions in a black-box way, we construct a $\widetilde{O}(\log n)$-round commitment scheme CECom' that is concurrently extractable without over-extraction. Recall that a commitment scheme is concurrently extractable without over-extraction if the rewinding extractor outputs $\bot$ except with negligible probability when the commitment is invalid.

Lemma 1. *Assume the existence of one-way functions. Then, there exists a $\widetilde{O}(\log n)$-round commitment scheme CECom' that is concurrently extractable without over-extraction. Furthermore, CECom' uses the underlying one-way function only in a black-box way.*

Proof. The commitment scheme CECom' is shown in Figure 3.

First, we show that CECom' is statistically binding and computationally hiding. The binding property follows directly from that of CECom. To show the hiding property, for any PPT receiver R^* and any $\sigma_0, \sigma_1 \in \{0,1\}^n$ we consider the following hybrid experiments for $b \in \{0,1\}$.

- In experiment $H_0^b(n)$, R^* receives a honest commitment to σ_b. The output of the experiment is that of R^*.

- Experiment $H_1^b(n)$ is the same as H_0^b except that R^* receives the following modified commitment.
  - In Step 1, the committed value Γ is extracted by brute force.
  - In Step 2, the committed value is switched from s_j to 0 for every $j \notin \Gamma$.
  - In Step 3, the committed value is switched from (s_j, d_j) to $(0,0)$ for every $j \notin \Gamma$.

  Note that, since the distribution of $(s_j)_{j\in\Gamma}$ is independent of σ_b, the internal R^* receives no information on σ_b. Thus, the output of $H_1^0(n)$ and that of $H_1^1(n)$ are identically distributed.

Let $\mathsf{H}_i^b(n)$ be the output of the experiment $H_i^b(n)$ for $i \in \{0,1\}$ and $b \in \{0,1\}$. Then, to show the hiding property, it suffices to show that $\mathsf{H}_0^b(n)$ and $\mathsf{H}_1^b(n)$ are indistinguishable for every $b \in \{0,1\}$. Assume for contradiction that there exists $b \in \{0,1\}$ such that $\mathsf{H}_0^b(n)$ and $\mathsf{H}_1^b(n)$ are distinguishable with non-negligible probability. Then, from an average argument, there exists a transcript ρ of Step 1 such that, under the condition that the transcript of Step 1 is ρ, $\mathsf{H}_0^b(n)$ and $\mathsf{H}_1^b(n)$ are distinguishable with non-negligible probability. Let Γ be the committed value of Com in ρ. Then, since $H_1^b(n)$ runs in polynomial time after Step 1 and since $H_0^b(n)$ and $H_1^b(n)$ differ only in the committed values of CECom and $\mathsf{wExtCom}$, from the assumption we can break the hiding property of either CECom or $\mathsf{wExtCom}$ by using ρ and Γ as non-uniform advice. Thus, we reach a contradiction.

Next, we show that CECom' is concurrently extractable without over-extraction. We extract the committed values of CECom' by extracting the committed values of CECom in Step 2. To avoid the over-extraction, we combine a technique used in [CDSMW08, CDSMW09, Wee10] (which relies on the property of cut-and-choose technique and the error-correcting property of Shamir's secret sharing scheme) with a technique used in [GLOV12] (which uses $\mathsf{wExtCom}$). Formally, for any PPT concurrent committer C^*, we consider the following extractor E.

- E internally runs C^* and emulates a concurrent interaction between C^* and honest receivers, but in Step 2 of each session, the committed values $\widehat{\boldsymbol{s}} = (\widehat{s}_1, \ldots, \widehat{s}_{40n})$ are extracted by using the concurrent extractability of CECom (i.e., by rewinding C^* in PTV rewinding strategy). Recall that the concurrent extraction of CECom involves the computation of the main thread and many look-ahead threads. Then, if the extraction of CECom fails in any accepted session on any thread, E outputs fail and halts. Otherwise, at the end of every session on every thread, E checks the following:
  - $\widehat{\boldsymbol{s}}$ is *0.8-close* to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{40n})$.
  - For every $j \in \Gamma$, w_j equals the value revealed in Step 5.

  If both of these hold, E sets the committed value σ of this session to be the value decoded from $\boldsymbol{w}$. Otherwise, E sets $\sigma := \bot$.
- When the main thread completes, E outputs τ, where τ is the view of C^* on the main thread.

From the concurrent extractability of CECom, E outputs fail with negligible probability. If E does not output fail, from the property of the rewinding strategy of [PTV12], τ is a perfect simulation of the view of $\mathcal{C}^*$. Thus, it remains to show that in every session on every thread, σ is the committed value of the session except with negligible probability. First, we note that

in a real interaction between C^* and honest receivers, we have the following facts in every session except with negligible probability.

Fact 1. Each commitment in Step 2 has at most one committed value. That is, the committed values $\boldsymbol{s} = (s_1, \ldots, s_{40n})$ of the Step 2 commitments are uniquely determined. (Recall that we define the committed value of an invalid commitment as $\perp$.) This follows from the statistical binding property of CECom.

Fact 2. The following probability is negligible: the probability that (i) every $\mathsf{wExtCom}$ commitment is accepted, (ii) for any $j \in \Gamma$, the committed value of the j-th $\mathsf{wExtCom}$ commitment is a valid decommitment of the corresponding CECom commitment, and (iii) $|\{j \in [40n] \mid s_j = \perp\}| \geq 2n$. That is, except with negligible probability, the session is rejected or there are less than $2n$ invalid CECom commitments. We prove this fact at the end of the proof.

Since in PTV rewinding strategy every thread is a perfect simulation of a real execution, Facts 1 and 2 also hold in every session on every thread in E except with negligible probability. In what follows, we assume that both of the facts indeed hold. Fact 1 guarantees that the committed value of CECom' is uniquely determined in every session on every thread. Then, for any accepted session on any thread, we consider the following two cases.

Case 1. First, we consider the case that $\boldsymbol{s}$ is 0.9-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{40n})$. (Recall that from Fact 1, $\boldsymbol{s}$ is uniquely determined.) In this case, the session is valid if and only if w_j equals the value revealed in Step 5 for every $j \in \Gamma$. Since we have $|\{j \in [40n] \mid s_j = \perp\}| < 2n$ in every accepted session and since the extractor of CECom extracts s_j when $s_j \neq \perp$, the extracted values $\widehat{\boldsymbol{s}}$ is 0.95-close to $\boldsymbol{s}$; hence, $\widehat{\boldsymbol{s}}$ is 0.85-close to $\boldsymbol{w}$. E therefore sets $\sigma := \perp$ if and only if w_j does not equal the value revealed in Step 5 for an index $j \in \Gamma$, i.e., if and only if the session is invalid. In addition, if $\sigma \neq \perp$, then σ is the (unique) committed value.

Case 2. Next, we consider the case that $\boldsymbol{s}$ is 0.1-far from any valid codeword. In this case, the session is invalid (i.e., the committed value is $\perp$). Below, we show that the probability that $\boldsymbol{s}$ is 0.1-far from any valid codeword but E sets $\sigma \neq \perp$ is negligible. If the probability that $\boldsymbol{s}$ is 0.1-far from any valid codeword is negligible, we are done. Thus, in the following, we assume that this probability is non-negligible. Recall that E sets $\sigma \neq \perp$ if and only if (i) the extracted shares $\widehat{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w}$ and (ii) $w_j = s_j$ for every $j \in \Gamma$.[7]

We show that the probability that (i) $\boldsymbol{s}$ is 0.1-far from any valid codeword, (ii) $\widehat{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w}$, and (iii) $w_j = s_j$ for every $j \in \Gamma$ is negligible.

Assume for contradiction that with non-negligible probability, there exists a session on a thread such that in the session (i) $\boldsymbol{s}$ is 0.1-far from any valid codeword, (ii) $\widehat{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w}$, and (iii) $w_j = s_j$ for every $j \in \Gamma$. Since both the number of sessions and the number of threads are at most $\mathsf{poly}(n)$, all of these holds with non-negligible probability in a randomly chosen session on a randomly chosen thread. We note that, since Fact 2 and the extractability of CECom guarantee that $\boldsymbol{s}$ and $\widehat{\boldsymbol{s}}$ are 0.95-close, $\widehat{\boldsymbol{s}}$ is 0.05-far from any valid codeword when $\boldsymbol{s}$ is 0.1-far from any valid codeword. Thus, with non-negligible probability, in

[7]Note that since s_j is the unique committed value of the j-th CECom commitment, s_j equals the value revealed in Stage 5 for any $j \in \Gamma$ in an accepted session.

a randomly chosen session on a randomly chosen thread (i) $\widehat{\boldsymbol{s}}$ is 0.05-far from any codeword, (ii) $\widehat{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w}$, and (iii) $w_j = s_j$ for every $j \in \Gamma$.

Then, consider the following adversary $\mathcal{A}$ against the hiding property of Com. For random subsets $\Gamma_0, \Gamma_1 \subset [40n]$ of size n, $\mathcal{A}$ tries to distinguish a commitment to Γ_0 from a commitment to Γ_1 as follows. $\mathcal{A}$ is the same as E except that in a randomly chosen session on a randomly chosen thread, $\mathcal{A}$ does the following.

- In Step 1, $\mathcal{A}$ receives a Com commitment from the external committer (the committed value is either Γ_0 or Γ_1) and forwards the commitment to the internal C^* as the Step 1 commitment. We note that, since $\mathcal{A}$ rewinds C^*, the Com commitment may be rewound. However, since Com is a *2-round* commitment scheme, this causes no problem. ($\mathcal{A}$ simply resends the same commitment, or receives a new commitment from the external committer and forwards the commitment to C^*.)
- After extracting the committed values of CECom commitments, $\mathcal{A}$ terminates and outputs 1 if and only if (i) the extracted values $\widehat{\boldsymbol{s}} = (\widehat{s}_1, \ldots, \widehat{s}_{40n})$ is 0.05-far from any valid codeword, (ii) $\widehat{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{40n})$, and (iii) $w_j = \widehat{s}_j$ for every $j \in \Gamma_1$; otherwise, $\mathcal{A}$ outputs 0.

We show that $\mathcal{A}$ breaks the hiding property of Com. When $\mathcal{A}$ receives a commitment to Γ_0, the probability that $\mathcal{A}$ outputs 1 is exponentially small. (Since the internal C^* receives no information of Γ_1, we have $\widehat{s}_j = w_j$ for every $j \in \Gamma_1$ with at most exponentially small probability when $\widehat{\boldsymbol{s}}$ and $\boldsymbol{w}$ are 0.05-far.) On the other hand, when $\mathcal{A}$ receives a commitment to Γ_1, from the assumption the probability that $\mathcal{A}$ outputs 1 is non-negligible. (Recall that, from the assumption, with non-negligible probability (i) $\widehat{\boldsymbol{s}}$ is 0.05-far from any codeword, (ii) $\widehat{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w}$, and (iii) $w_j = s_j$ for every $j \in \Gamma_1$. Then, since for every $j \in \Gamma_1$ it hold that $s_j = w_j \neq \perp$, we have $\widehat{s}_j = s_j$. Thus, when this happens, for every $j \in \Gamma_1$ we have $w_j = \widehat{s}_j$.)

Since this contradicts to the hiding property of Com, we conclude that the probability that (i) $\boldsymbol{s}$ is 0.1-far from any valid codeword, (ii) $\widehat{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w}$, and (iii) $w_j = s_j$ for every $j \in \Gamma$ is negligible. Thus, the probability that $\boldsymbol{s}$ is 0.1-far from any valid codeword but E sets $\sigma \neq \perp$ is negligible.

From the analysis of these two cases, we conclude that in every session on every thread, σ is the committed value except with negligible probability. From the union bound, we conclude that except with negligible probability, σ is the committed value in every session on every thread.

Finally, we prove Fact 2.

Proof of Fact 2. First, we give some definitions. In each session, for $j \in [40n]$, the *j-th column* is the pair of the j-th CECom commitment in Step 2 and the j-th $\mathsf{wExtCom}$ commitment in Step 3. We say that a column is *consistent* if in the column the committed value of the $\mathsf{wExtCom}$ commitment is a valid decommitment of the corresponding CECom commitment; otherwise, the column is *inconsistent*. We say that C^* *cheats* in a session if (i) every $\mathsf{wExtCom}$ commitment is accepted, (ii) the j-th column is consistent for every $j \in \Gamma$, and (iii) there exist at least $2n$ inconsistent columns.

To prove Fact 2, we show that in every session the probability that C^* cheats is negligible. From the definition, Fact 2 follows.

Assume for contradiction that for infinitely many n, there is a session in which C^* cheats with probability at least $1/\mathsf{poly}(n)$. In the following, we fix any such n. Then, since the number of sessions is at most $\mathsf{poly}(n)$, there is an $i^* \in [\mathsf{poly}(n)]$ such that in the i^*-th session, C^* cheats with probability at least $1/n^c$ for a constant c.

Then, let us consider an adversary $\mathcal{B}$ against the hiding property of Com. For random subsets $\Gamma_0, \Gamma_1 \subset [40n]$ of size n, $\mathcal{B}$ tries to distinguish a Com commitment to Γ_0 from a Com commitment to Γ_1 as follows. $\mathcal{B}$ internally invokes C^* and honestly emulates the interaction between C^* and honest receivers except that in the i^*-th session, $\mathcal{B}$ does the following.

- In Step 1, $\mathcal{B}$ receives a Com commitment from the external committer (the committed value is either Γ_0 or Γ_1) and forwards the commitment to C^* as the Step 1 commitment.
- When Step 3 is accepted (i.e., all the $\mathsf{wExtCom}$ commitments are accepted), $\mathcal{B}$ does the following repeatedly: $\mathcal{B}$ rewinds C^* to the point that the next-message is the challenge bits of $\mathsf{wExtCom}$ in the i^*-th session; then $\mathcal{B}$ sends new random challenge bits and honestly interacts with C^* until the end of Step 3 (i.e., until receiving the replies in $\mathsf{wExtCom}$). After collecting other n^{c+3} accepted transcripts of Step 3, $\mathcal{B}$ outputs 1 if the following hold:
  - (i) from these $n^{c+3}+1$ accepted transcript (the first one and the subsequent n^{c+3} ones), $\mathcal{B}$ can extract the committed values of $\mathsf{wExtCom}$ in at least $39n$ columns,
  - (ii) in at least n columns of these columns, the extracted values are not valid decommitments of the corresponding CECom commitments, and
  - (iii) for every $j \in \Gamma_1$, either the extraction of the j-th column fails or the extracted value of the j-th column is a valid decommitment of the corresponding CECom commitment.

  Otherwise, $\mathcal{B}$ outputs 0. In the following, the first transcript that $\mathcal{B}$ generates in Step 3 is called the *main thread* and other n^{c+3} accepted transcripts are called the *look-ahead threads*.

If $\mathcal{B}$ rewinds C^* more than n^{3c+4} times, $\mathcal{B}$ terminates and outputs fail.

First, we show that an expected polynomial-time adversary $\mathcal{B}'$ successfully distinguishes Com commitments, where $\mathcal{B}'$ is the same as $\mathcal{B}$ except that $\mathcal{B}'$ does not terminate after $\mathcal{B}'$ rewinds C^* more than n^{3c+4} times. When $\mathcal{B}'$ receives a commitment to Γ_0, since the internal C^* receives no information of Γ_1, the probability that $\mathcal{B}'$ outputs 1 is exponentially small. (This is because when Condition (i) and Condition (ii) hold, the probability that Condition (iii) holds is exponentially small.) Thus, it remains to show that when $\mathcal{B}'$ receives a commitment to Γ_1, the probability that $\mathcal{B}'$ outputs 1 is at least $1/\mathsf{poly}(n)$. Let $\mathsf{extract}$ be the event that $\mathcal{B}'$ extracts the committed values of $\mathsf{wExtCom}$ commitments from at least $39n$ columns, and let cheat be the event that C^* cheats in the i^*-th session on the main thread. Then, to show that $\mathcal{B}'$ outputs 1 with probability at least $1/\mathsf{poly}(n)$, it suffices to show that

$$\Pr[\mathsf{cheat} \wedge \mathsf{extract}] \geq \frac{1}{\mathsf{poly}(n)} \quad . \tag{1}$$

(Recall the we can extract the committed values of $\mathsf{wExtCom}$ without over-extraction.) Let ρ be a prefix of a transcript between C^* and honest receivers such that after ρ, a honest receiver sends challenge bits of $\mathsf{wExtCom}$ in the i^*-th session. Let prefix_ρ be the event that a prefix

of the main thread is ρ. Then, since the probability that C^* cheats in the i^*-th session is at least $1/n^c$, from an average argument, we have $\Pr\left[\mathsf{cheat} \mid \mathsf{prefix}_\rho\right] \geq 1/2n^c$ with probability at least $1/2n^c$ over the choice of ρ (i.e., when we obtain ρ by emulating the interaction between C^* and honest receivers). Let Δ be the set of prefixes such that $\Pr\left[\mathsf{cheat} \mid \mathsf{prefix}_\rho\right] \geq 1/2n^c$ holds. Then, since we have $\sum_{\rho\in\Delta} \Pr\left[\mathsf{prefix}_\rho\right] \geq 1/2n^c$, we have

$$\begin{aligned}
\Pr\left[\mathsf{cheat} \wedge \mathsf{extract}\right] &\geq \sum_{\rho\in\Delta} \Pr\left[\mathsf{cheat} \wedge \mathsf{extract} \mid \mathsf{prefix}_\rho\right] \cdot \Pr\left[\mathsf{prefix}_\rho\right] \\
&\geq \min_{\rho\in\Delta}\left(\Pr\left[\mathsf{cheat} \wedge \mathsf{extract} \mid \mathsf{prefix}_\rho\right]\right) \cdot \sum_{\rho\in\Delta} \Pr\left[\mathsf{prefix}_\rho\right] \\
&\geq \frac{1}{2n^c} \min_{\rho\in\Delta}\left(\Pr\left[\mathsf{cheat} \wedge \mathsf{extract} \mid \mathsf{prefix}_\rho\right]\right) \quad . \qquad (2)
\end{aligned}$$

Thus, to show Equation (1), it suffices to show that for any $\rho \in \Delta$, we have

$$\Pr\left[\mathsf{cheat} \wedge \mathsf{extract} \mid \mathsf{prefix}_\rho\right] \geq \frac{1}{\mathsf{poly}(n)} \quad . \qquad (3)$$

In the following, we fix any $\rho^* \in \Delta$. Then, we have

$$\Pr\left[\mathsf{cheat} \mid \mathsf{prefix}_{\rho^*}\right] \geq \frac{1}{2n^c} \quad . \qquad (4)$$

Thus, from Equation (4), we have

$$\begin{aligned}
\Pr\left[\mathsf{cheat} \wedge \mathsf{extract} \mid \mathsf{prefix}_{\rho^*}\right] &= \Pr\left[\mathsf{cheat} \mid \mathsf{prefix}_{\rho^*}\right] \cdot \Pr\left[\mathsf{extract} \mid \mathsf{prefix}_{\rho^*} \wedge \mathsf{cheat}\right] \\
&\geq \frac{1}{2n^c} \Pr\left[\mathsf{extract} \mid \mathsf{prefix}_{\rho^*} \wedge \mathsf{cheat}\right] \qquad (5)
\end{aligned}$$

Thus, to show Equation (3), it suffices to show that

$$\Pr\left[\mathsf{extract} \mid \mathsf{prefix}_{\rho^*} \wedge \mathsf{cheat}\right] \geq \frac{1}{\mathsf{poly}(n)} \quad . \qquad (6)$$

Recall that when cheat occurs, Step 3 of the i^*-th session is accepted on the main thread. Thus, for any $j \in [40n]$, when cheat occurs and the challenge bit of $\mathsf{wExtCom}$ in the j-th column is $b \in \{0,1\}$ on the main thread, we can extract the committed value of the the j-th column if in the n^{c+3} look-ahead threads there is an accepted transcript of $\mathsf{wExtCom}$ such that the challenge bit of the j-th column is $1-b$. Then, to show Equation (6), we show that when Step 3 of the i^*-th session is accepted on the main thread with prefix ρ^*, the probability that the challenge bit of $\mathsf{wExtCom}$ is b is "high" for any $b \in \{0,1\}$ in "most" columns. Let ch_j be a random variable for the challenge bit of $\mathsf{wExtCom}$ in the j-th column of the i^*-th session on the main thread, and let accept be the event that every $\mathsf{wExtCom}$ commitment is accepted in the i^*-th session on the main thread. (We have $\Pr\left[\mathsf{accept}\right] \geq \Pr\left[\mathsf{cheat}\right]$ from the definitions.) Then, for any $j \in [40n]$ and $b \in \{0,1\}$,

$$\begin{aligned}
\Pr\left[ch_j = b \mid \mathsf{accept} \wedge \mathsf{prefix}_{\rho^*}\right] &= \frac{\Pr\left[ch_j = b \wedge \mathsf{accept} \wedge \mathsf{prefix}_{\rho^*}\right]}{\Pr\left[\mathsf{accept} \wedge \mathsf{prefix}_{\rho^*}\right]} \\
&\geq \frac{\Pr\left[ch_j = b \wedge \mathsf{cheat} \wedge \mathsf{prefix}_{\rho^*}\right]}{\Pr\left[\mathsf{prefix}_{\rho^*}\right]} \\
&= \frac{\Pr\left[\mathsf{cheat} \;\middle|\; ch_j = b \wedge \mathsf{prefix}_{\rho^*}\right] \Pr\left[ch_j = b \wedge \mathsf{prefix}_{\rho^*}\right]}{\Pr\left[\mathsf{prefix}_{\rho^*}\right]} \\
&= \Pr\left[\mathsf{cheat} \;\middle|\; ch_j = b \wedge \mathsf{prefix}_{\rho^*}\right] \Pr\left[ch_j = b\right] \quad . \qquad (7)
\end{aligned}$$

(Here, we use $\Pr\left[ch_j = b \wedge \mathsf{prefix}_{\rho^*}\right] = \Pr\left[ch_j = b\right] \cdot \Pr\left[\mathsf{prefix}_{\rho^*}\right]$.) Below, we show that in at least $39n$ columns of the i^*-th session, for any $b \in \{0,1\}$ we have

$$\Pr\left[\mathsf{cheat} \;\middle|\; ch_j = b \wedge \mathsf{prefix}_{\rho^*}\right] \geq \frac{1}{160n^{c+1}} \ . \tag{8}$$

Let

$$A := \left\{ j \in [40n] \;\middle|\; \exists b_j \in \{0,1\} \text{ s.t. } \Pr\left[\mathsf{cheat} \mid ch_j = b_j \wedge \mathsf{prefix}_{\rho^*}\right] < \frac{1}{160n^{c+1}} \right\} \ .$$

Then we have

$$\begin{aligned}
\Pr\left[\mathsf{cheat} \;\middle|\; \mathsf{prefix}_{\rho^*}\right] &\leq \Pr\left[\bigwedge_{j\in A} ch_j = 1 - b_j\right] + \Pr\left[\mathsf{cheat} \bigwedge \left(\bigvee_{j\in A} ch_j = b_j\right) \;\middle|\; \mathsf{prefix}_{\rho^*}\right] \\
&\leq 2^{-|A|} + \sum_{j\in A} \Pr\left[\mathsf{cheat} \wedge ch_j = b_j \mid \mathsf{prefix}_{\rho^*}\right] \\
&= 2^{-|A|} + \sum_{j\in A} \Pr\left[\mathsf{cheat} \mid ch_j = b_j \wedge \mathsf{prefix}_{\rho^*}\right] \Pr\left[ch_j = b_j\right] \\
&\leq 2^{-|A|} + \sum_{j\in A} \Pr\left[\mathsf{cheat} \mid ch_j = b_j \wedge \mathsf{prefix}_{\rho^*}\right] \\
&< 2^{-|A|} + 40n \cdot \frac{1}{160n^{c+1}} \\
&\leq 2^{-|A|} + \frac{1}{4n^c} \ .
\end{aligned} \tag{9}$$

Then, from Equations (4) and (9), we have $|A| = O(\log n)$ and therefore $|A| \leq n$. Thus, in at least $39n$ columns, for any $b \in \{0,1\}$ we have Equation (8). Then, from Equations (7) and (8) and from $\Pr\left[ch_j = b\right] = 1/2$, for any $j \in [40n] \setminus A$ and any $b \in \{0,1\}$, we have

$$\Pr\left[ch_j = b \mid \mathsf{accept} \wedge \mathsf{prefix}_{\rho^*}\right] \geq \frac{1}{320n^{c+1}} \ .$$

Then, since the distributions of the look-ahead threads are the same as that of the main thread, we have that under the condition that prefix_{ρ^*} and cheat occur, for any $j \in [40n] \setminus A$, the adversary $\mathcal{B}'$ requires another $320n^{c+1}$ accepted transcripts on average to extract the committed value of $\mathsf{wExtCom}$ in the j-th columns. Since $\mathcal{B}'$ collects n^{c+3} accepted transcripts, for any $j \in [40n] \setminus A$ the adversary $\mathcal{B}'$ extracts the committed value of $\mathsf{wExtCom}$ in the j-th column except with probability $320n^{c+1}/n^{c+3} = 320/n^2$ under the condition that prefix_{ρ^*} and cheat occur. (Here, we use Markov's inequality.) Then, from the union bound, except with probability $39n \cdot 320/n^2 = 12480/n$, for every $j \in [40n] \setminus A$ the adversary $\mathcal{B}'$ extracts the committed value of $\mathsf{wExtCom}$ in the j-th column. Thus, we have

$$\Pr\left[\mathsf{extract} \mid \mathsf{prefix}_{\rho^*} \wedge \mathsf{cheat}\right] \geq 1 - \frac{12480}{n} \ . \tag{10}$$

Then, from Equations (5) and (10), we have

$$\Pr\left[\mathsf{cheat} \wedge \mathsf{extract} \mid \mathsf{prefix}_{\rho^*}\right] \geq \frac{1}{2n^c} \cdot \left(1 - \frac{12480}{n}\right) \geq \frac{1}{4n^c} \ . \tag{11}$$

Then, since ρ^* is any prefix in Δ, from Equations (2) and (11) we have

$$\Pr[\mathsf{cheat} \wedge \mathsf{extract}] \geq \frac{1}{2n^c} \cdot \frac{1}{4n^c} = \frac{1}{8n^{2c}} \ .$$

Thus, we have Equation (1). We therefore conclude that $\mathcal{B}'$ outputs 1 with probability at least $1/8n^{2c}$ when $\mathcal{B}'$ receives a commitment to Γ_1. Thus, $\mathcal{B}'$ successfully distinguishes a commitment to Γ_1 from a commitment to Γ_0.

Now, we are ready to show that $\mathcal{B}$ breaks the hiding property of Com. Clearly, the running time of $\mathcal{B}$ is at most $\mathsf{poly}(n)$. Note that, to show that $\mathcal{B}$ can distinguish Com commitments, it suffices to show that the output of $\mathcal{B}$ is the same as that of $\mathcal{B}'$ except with probability $1/n^{2c+1}$. (This is because $\mathcal{B}'$ outputs 1 with negligible probability when $\mathcal{B}'$ receives a commitment to Γ_0 whereas $\mathcal{B}'$ outputs 1 with with probability $1/8n^{2c}$ when $\mathcal{B}'$ receives a commitment to Γ_1.) Recall that the output of $\mathcal{B}$ differs from that of $\mathcal{B}'$ if and only if $\mathcal{B}'$ rewinds C^* more than n^{3c+4} times. Let ρ be any prefix of a transcript between C^* and honest receivers such that after ρ, the next message is the challenge bits of $\mathsf{wExtCom}$ in the i^*-th session. Let $T(n)$ be a random variable for the number of rewinding in $\mathcal{B}'$. Then, we have

$$\mathrm{E}\left[T(n) \mid \mathsf{prefix}_\rho\right] \leq \Pr\left[\mathsf{accept} \mid \mathsf{prefix}_\rho\right] \cdot \frac{n^{c+3}}{\Pr\left[\mathsf{accept} \mid \mathsf{prefix}_\rho\right]} = n^{c+3} \ .$$

Thus, we have

$$\begin{aligned}\mathrm{E}[T(n)] &= \sum_\rho \Pr\left[\mathsf{prefix}_\rho\right] \mathrm{E}\left[T(n) \mid \mathsf{prefix}_\rho\right] \\ &\leq n^{c+3} \sum_\rho \Pr\left[\mathsf{prefix}_\rho\right] \leq n^{c+3} \ .\end{aligned}$$

Then, from Markov's inequality, $\mathcal{B}'$ rewinds C^* more than n^{3c+4} times with probability at most $n^{c+3}/n^{3c+4} = 1/n^{2c+1}$. Thus, the output of $\mathcal{B}$ is the same as that of $\mathcal{B}'$ except with probability $1/n^{2c+1}$, and therefore $\mathcal{B}$ distinguishes a commitment to Γ_1 from a commitment to Γ_0. □

This complete the proof of Lemma 1. □

Remark 1. Note that since the extractor E of CECom' essentially uses only the extractor of CECom, E rewinds C^* in PTV rewinding strategy. Thus, when E extracts the committed values, E runs in strict polynomial-time and all the properties stated in Section 3.4 hold. We use these properties in Section 5.

4.2 One-One CCA-Secure Commitment Scheme

Using one-way functions in a black-box way, we construct a one-one CCA-secure commitment scheme $\mathsf{CCACom}^{1:1}$. Recall that a commitment scheme is one-one CCA secure if it is CCA secure w.r.t. a restricted class of adversaries that start only a single right session. Our scheme does not satisfy the statistically binding property, but satisfy the strong computational binding property.

Lemma 2. *Assume the existence of one-way functions. Then, there exists a $O(\log n)$-round one-one CCA-secure commitment scheme $\mathsf{CCACom}^{1:1}$, which satisfies the strong computational binding property and the computational hiding property. Furthermore, $\mathsf{CCACom}^{1:1}$ uses the underlying one-way function only in a black-box way.*

Proof. We construct $\mathsf{CCACom}^{1:1}$ by slightly modifying the black-box $O(n^\epsilon)$-round CCA-secure commitment scheme of [LP12] and using the DDN $\log n$ trick [DDN00, LPV08], which transforms concurrent non-malleable commitment schemes for tags of length $O(\log n)$ to a non-malleable commitment schemes for tags of length $O(n)$ with no increase on round complexity.

First, we recall the CCA-secure commitment scheme of [LP12] (see Figure 4). Roughly speaking, the commitment scheme of [LP12] consists of $4\ell(n)\eta(n)$ *rows*—each row is a parallel execution of a part of the trapdoor commitment scheme of [PW09]—followed by a cut-and-choose phase, where $\ell(n)$ is the length of the tag and $\eta(n) = n^\epsilon$ for $\epsilon > 0$. In the analysis of [LP12], which is based on that of [CLP10], it is shown that in any transcript of one left session and many right sessions of the scheme, each right session has $\Omega(\eta(n))$ $\mathsf{safe\text{-}points}$, from which we can rewind the right session and extract the committed value of this session without breaking the hiding property of the left session. Then, since each right session has $\Omega(\eta(n))$ $\mathsf{safe\text{-}points}$, we can extract the committed value of each right session even in the concurrent setting by using the rewinding strategy of [RK99] to deal with the issue of recursive rewinding. Thus, by extracting the committed-value of a row in each right session, we can emulate the committed-value oracle in polynomial time without breaking the hiding property of the left session. Thus, the CCA security follows from the hiding property of the left session.

Commit Phase

Let ℓ, η be two polynomials such that $\ell(n) = n^\nu$ and $\eta(n) = n^\epsilon$ for $\nu, \epsilon > 0$, and L be a polynomial such that $L(n) = 4\ell(n)\eta(n)$. To commit to a value v, the committer C and the receiver R, on common input 1^n and $\mathsf{tag} \in \{0,1\}^{\ell(n)}$, do the following.

Stage 1: R sends the Step 1 message of a commitment of $\mathsf{PTrapCom}$. That is, a commitment of Com to a randomly chosen string challenge $\boldsymbol{e} = (e_1, \ldots, e_n)$.

Stage 2: C computes an $(n+1)$-out-of-$10n$ Shamir's secret sharing $\boldsymbol{s} = (s_1, \ldots, s_{10n})$ of value v, and commits to these shares using Step 2 of $\mathsf{PTrapCom}$ in parallel, for $L(n)$ times; we call the i-th parallel commitment the i-th *row*, and all the commitments to s_j the j-th *column*. Messages in the $4\ell(n)\eta(n)$ rows are scheduled based on tag and relies on scheduling pairs of rows according to schedules design_0 and design_1 depicted in Figure 5. More precisely, Stage 2 consist of $\ell(n)$ phases. In phase i, C provides $\eta(n)$ sequential $\mathsf{design}_{\mathsf{tag}_i}$ pairs of rows, followed by $\eta(n)$ sequential $\mathsf{design}_{1-\mathsf{tag}_i}$ pairs of rows.

Stage 3: R decommits the Stage 1 commitment to $\boldsymbol{e}$. C completes the $10nL(n)$ executions of $\mathsf{PTrapCom}$ w.r.t. challenge $\boldsymbol{e}$ in parallel.

Stage 4: R sends a randomly chosen subset $\Gamma \subset [10n]$ of size n. For every $j \in \Gamma$, C decommits all the commitments in the j-th column of Stage 3. R checks that all the decommitments are valid, and reveal the same committed values s_j.

Figure 4: Black-box CCA-secure commitment scheme of [LP12]

Then, we observe that by setting $\eta(n) := 1$ in the scheme of [LP12], we obtain a black-box $O(\ell(n))$-round *parallel CCA-secure* commitment scheme for tags of length $\ell(n)$, where a commitment scheme is *parallel CCA secure* if it is CCA secure w.r.t. a restricted class of adversaries that start only a single parallel right session. This is because, when an adversary

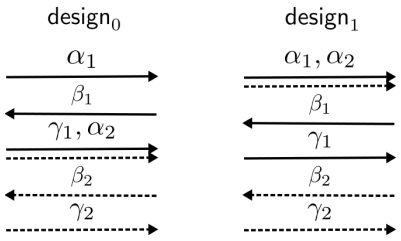

Figure 5: Description of the schedules used in Stage 2 of the protocol of [LP12]. $(\alpha_1, \beta_1, \gamma_1)$ and $(\alpha_2, \beta_2, \gamma_2)$ are respectively the transcripts of a pair of rows in Stage 2.

starts only a single parallel right session, we do not need to worry about recursive rewinding and therefore each right session need to have only a single safe-point as in the concurrent non-malleable commitment scheme of [LPV08] (on which the CCA-secure commitment schemes of [CLP10, LP12] are based). Therefore, by setting $\eta(n) := 1$ and $\ell(n) := O(\log n)$, we obtain a black-box $O(\log n)$-round commitment scheme that is parallel CCA secure for tags of length $O(\log n)$.

Next, we observe that the DDN $\log n$ trick [DDN00, LPV08] transforms any black-box parallel CCA-secure commitment scheme for tags of length $O(\log n)$ to a black-box one-one CCA-secure commitment scheme for tags of length $O(n)$. This can be proven in essentially the same way as the proof of the fact that the DDN $\log n$ trick transforms a concurrent non-malleable commitment scheme for tags of length $O(\log n)$ to a non-malleable commitment scheme for tags of length $O(n)$. For details, see Appendix B.

Combining above, we obtain a black-box $O(\log n)$-round one-one CCA-secure commitment scheme $\mathsf{CCACom}^{1:1}$. $\mathsf{CCACom}^{1:1}$ satisfies the strong computational binding property and the computational hiding property because the CCA-secure commitment scheme of [LP12] satisfies both properties and the DDN $\log n$ trick preserves both properties. (The strong computational binding property of [LP12] follows from that of the trapdoor commitment scheme of [PW09].) □

5 CCA-Secure Commitment Scheme

Using one-way functions in a black-box way, we construct a $\widetilde{O}(\log^2 n)$-round robust CCA-secure commitment scheme.

Theorem 1. *Assume the existence of one-way functions. Then, there exists a $\widetilde{O}(\log^2 n)$-round robust CCA-secure commitment scheme* CCACom*. Furthermore,* CCACom *uses the underlying one-way function only in a black-box way.*

Proof. Figure 6 shows CCACom, which uses CECom' shown in Lemma 1 and $\mathsf{CCACom}^{1:1}$ shown in Lemma 2. For simplicity, we use a non-interactive perfectly binding commitment scheme Com, which is based on one-way permutations. Using a standard technique, we can replace Com with 2-round statistically binding commitment scheme based on one-way functions.[8]

First, we note that the statistical binding property of CCACom follows directly from that of Com.

[8]By sending the first-round message at the beginning of the protocol, we can use a 2-round commitment scheme as a non-interactive commitment scheme.

Commit Phase

The committer C and the receiver R receive common inputs 1^n and $\mathsf{tag} \in \{0,1\}^n$. To commit to $v \in \{0,1\}^n$, the committer C does the following with the receiver R.

Stage 1. R commits to a random subset $\Gamma \subset [10n]$ of size n by using $\mathsf{CCACom}^{1:1}$ with tag tag.

Stage 2. C computes an $(n+1)$-out-of-$10n$ Shamir's secret sharing $\boldsymbol{s} = (s_1, \ldots, s_{10n})$ of value v. Next, for each $j \in [10n]$, C chooses random $d_j \in \{0,1\}^{\mathsf{poly}(n)}$ and computes $c_j := \mathsf{Com}(s_j; d_j)$ (i.e., commits to s_j with randomness d_j). Then, C sends $(c_1, \ldots, c_{10n})$ to R.

Stage 3. Let $r_{\mathrm{CCA}} := r_{\mathrm{CCA}}(n)$ be the number of rounds in $\mathsf{CCACom}^{1:1}$, and $r_{\mathrm{CEC}} := r_{\mathrm{CEC}}(n)$ be the number of rounds in CECom'. Let $\eta := 2r_{\mathrm{CCA}} + r_{\mathrm{CEC}} + 1$. Then, for each $i \in [\eta]$ in sequence, C does the following.

- For each $j \in [10n]$ in parallel, C commits to (s_j, d_j) by using CECom'.

We call the i-th parallel commitments the *i-th row*, and call all the commitments to (s_j, d_j) the *j-th column*.

Stage 4. R decommits the Stage 1 commitment to Γ.

Stage 5. For each $j \in \Gamma$, C decommits all the commitments in the j-th column to (s_j, d_j). R checks whether $c_j = \mathsf{Com}(s_j; d_j)$ for every $j \in \Gamma$.

Decommit Phase

C sends v and decommits all the Stage 2 commitments. Then, R defines $\boldsymbol{s} = (s_1, \ldots, s_{10n})$ as follows: If the j-th decommitment is invalid, $s_j := \bot$; otherwise, s_j is the j-th decommitted value. R accepts the decommitments if and only if $V(\boldsymbol{s}) = v$, where for any $\boldsymbol{t} = (t_1, \ldots, t_{10n})$, $V(\boldsymbol{t})$ is defined as follows: If $\boldsymbol{t}$ is 0.9-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10n})$ and for each $j \in \Gamma$, w_j equals the value revealed in Stage 5, then $V(\boldsymbol{t})$ is the value decoded from $\boldsymbol{w}$; otherwise, $V(\boldsymbol{t}) = \bot$.

Figure 6: CCA commitment scheme CCACom.

It remains to show that CCACom is robust CCA secure. (The hiding property of CCACom follows from CCA security.) Formally, we consider the following lemmas.

Lemma 3. CCACom *is CCA secure.*

Lemma 4. *For any constant κ,* CCACom *is κ-robust.*

The theorem follows from these lemmas. □

5.1 Proof of CCA Security

Proof of Lemma 3. First, we note that at the end of each right session, $\mathcal{O}$ computes the committed value by extracting the committed values $\boldsymbol{s} = (s_1, \ldots, s_{10n})$ of the Stage 2 com-

mitments in super-polynomial time and computing $V(\boldsymbol{s})$. Then, for any PPT adversary $\mathcal{A}$, we consider the following hybrid experiments for each $b \in \{0,1\}$.

Hybrid $H_0^b(n,z)$: Hybrid $H_0^b(n,z)$ is the same as $\mathrm{IND}_b(\mathsf{CCACom}, \mathcal{A}, n, z)$.

Hybrid $H_1^b(n,z)$ to Hybrid $H_\eta^b(n,z)$: For $k \in [\eta]$, hybrid $H_k^b(n,z)$ is the same as $H_0^b(n,z)$ except for the following.

- In Stage 1 on the left, the committed value Γ is extracted by brute force. If the commitment is invalid, Γ is set to be a random subset. If the commitment has more than one committed value, $H_k^b(n,z)$ outputs fail and terminates.
- In Stage 3 on the left, the left committer commits to (0,0) instead of (s_j, d_j) for every $i \in [k]$ and $j \notin \Gamma$.

Hybrid $H_{\eta+1}^b(n,z)$: Hybrid $H_{\eta+1}^b(n,z)$ is the same as $H_\eta^b(n,z)$ except that in Stage 2 on the left, the left committer commits to 0 instead of s_j for every $j \notin \Gamma$.

Let $\mathsf{H}_k^b(n,z)$ be a random variable for the output of $H_k^b(n,z)$. Below, we prove the following claims.

Claim 1. *For every $b \in \{0,1\}$ and $k \in [\eta]$, $\left\{\mathsf{H}_{k-1}^b(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ and $\left\{\mathsf{H}_k^b(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ are computationally indistinguishable.*

Claim 2. *For every $b \in \{0,1\}$, $\left\{\mathsf{H}_\eta^b(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ and $\left\{\mathsf{H}_{\eta+1}^b(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ are computationally indistinguishable.*

Since $\mathcal{A}$ receives no information about b in $H_{\eta+1}^b(n,z)$, distribution $\left\{\mathsf{H}_{\eta+1}^0(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ and $\left\{\mathsf{H}_{\eta+1}^1(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ are identically distributed. Thus, Lemma 3 follows from these claims. □

5.1.1 Proof of Claim 1

Proof of Claim 1. In this proof, we use the following claim.

Claim 3. *For every $b \in \{0,1\}$, $k \in [\eta+1]$ and $z \in \{0,1\}^*$, $H_k^b(n,z)$ outputs fail with at most negligible probability.*

The proof of Claim 3 is given in Section 5.1.3.

Below, we show the indistinguishability of this claim under the condition that $H_{k-1}^b(n,z)$ and $H_k^b(n,z)$ do not output fail. From Claim 3, this suffices to prove the claim.

Since $H_{k-1}^b(n,z)$ and $H_k^b(n,z)$ differ only in the commitments in the k-th row on the left, we prove the indistinguishability by using the hiding property of CECom'. The problem is that in $H_{k-1}^b(n,z)$ and $H_k^b(n,z)$, $\mathcal{A}$ makes queries to $\mathcal{O}$, which runs in super-polynomial time. Thus, we cannot directly use the computational hiding property of CECom' to prove the indistinguishability.

To use the computational hiding property of CECom', we consider experiments $\widetilde{H}_{k-1}^b(n,z)$ and $\widetilde{H}_k^b(n,z)$, in which $\mathcal{O}$ is emulated in polynomial time by rewinding $\mathcal{A}$. Formally, for every $h \in \{k-1,k\}$, $\widetilde{H}_h^b(n,z)$ is the same as $H_h^b(n,z)$ except for the following.

- In Stage 3 of each right session, let us consider the following condition.

Condition 1. A row satisfies Condition 1 if and only if the row contains no message of Stage 1 on the left and the k-th row on the left (i.e., after the row starts, $\mathcal{A}$ sends no such message until the row ends).

Note that, since every session has $2r_{\mathrm{CCA}} + r_{\mathrm{CEC}} + 1$ rows, at least $r_{\mathrm{CCA}} + 1$ rows satisfy Condition 1 in every right session. Then, the committed values of every row satisfying Condition 1 are extracted by using the concurrent extractability of CECom' and the technique of [LP11] (see Appendix A). We note that *neither Stage 1 on the left nor the k-th row on the left is rewound* in the extraction. (Later, we use this fact when we use the hiding property of the k-th row commitment and the one-one CCA security of the Stage 1 commitment.) If the extraction fails, $\widetilde{H}^b_h(n,z)$ outputs $\mathsf{fail}_{\mathrm{ext}}$ and halts. (From the concurrent extractability of CECom', this happens with negligible probability.)

- For any $\boldsymbol{t} = (t_1, \ldots, t_{10n})$, we define $V'(\boldsymbol{t})$ as follows: If $\boldsymbol{t}$ is *0.8-close* to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10n})$ and for every $j \in \Gamma$, w_j equals the value revealed in Stage 5, then $V'(\boldsymbol{t})$ is the value decoded from $\boldsymbol{w}$; otherwise, $V'(\boldsymbol{t}) = \bot$. Then, at the end of each right session, $\mathcal{O}$ returns $V'(\boldsymbol{s'})$ to $\mathcal{A}$ instead of $V(\boldsymbol{s})$, where $\boldsymbol{s'} = (s'_1, \ldots, s'_{10n})$ is the extracted values of the first row satisfying Condition 1 in this session. We note that $\mathcal{O}$ does not break the hiding property of the Stage 2 commitments, and therefore $\mathcal{O}$ runs in polynomial time.

Then, we consider the following claim.

Claim 4. *For every $h \in \{k-1, k\}$, $\left\{\mathsf{H}^b_h(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ and $\left\{\widetilde{\mathsf{H}}^b_h(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ are statistically indistinguishable.*

Before proving Claim 4, we finish the proof of Claim 1 by using Claim 4. Given Claim 4, we can prove Claim 1 by showing that $\left\{\widetilde{\mathsf{H}}^b_{k-1}(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ and $\left\{\widetilde{\mathsf{H}}^b_k(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ are computationally indistinguishable. $\widetilde{H}^b_{k-1}(n,z)$ and $\widetilde{H}^b_k(n,z)$ differ only in the committed values of the k-th row on the left. In addition, both in $\widetilde{H}^b_{k-1}(n,z)$ and in $\widetilde{H}^b_k(n,z)$, oracle $\mathcal{O}$ is emulated in polynomial time and the k-th row on the left is not rewound. Thus, we can directly use the hiding property of CECom' to show the indistinguishability. Formally, assume for contradiction that for infinitely many n, there exists $z \in \{0,1\}^*$ such that $\widetilde{\mathsf{H}}^b_{k-1}(n,z)$ and $\widetilde{\mathsf{H}}^b_k(n,z)$ can be distinguished with probability $1/\mathsf{poly}(n)$. Then, since $\widetilde{H}^b_{k-1}(n,z)$ and $\widetilde{H}^b_k(n,z)$ proceed identically until the k-th row starts on the left, there exists a prefix ρ of $\widetilde{H}^b_{k-1}(n,z)$ such that (i) immediately after ρ, the k-th row starts on the left and (ii) under the condition that a prefix is ρ, $\widetilde{\mathsf{H}}^b_{k-1}(n,z)$ and $\widetilde{\mathsf{H}}^b_k(n,z)$ can be distinguished with probability $1/\mathsf{poly}(n)$. Since ρ contains the entire transcript of Stage 1 on the left, ρ uniquely determines the committed value Γ of the Stage 1 commitment on the left. Then, we consider the following PPT adversary $\mathcal{B}$ against the hiding property of CECom'.

- Receiving ρ and Γ as auxiliary inputs, $\mathcal{B}$ internally invokes $\mathcal{A}$ and honestly emulates $\widetilde{H}^b_{k-1}(n,z)$ after ρ except that in the k-th row on the left, $\mathcal{B}$ receives either commitments to $(s_j, d_j)_{j\notin\Gamma}$ or commitments to $(0, \ldots, 0)$ from the external committer and forwards them to $\mathcal{A}$. Then, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

Since $\mathcal{B}$ perfectly emulates either $\widetilde{H}^b_{k-1}(n,z)$ or $\widetilde{H}^b_k(n,z)$, our assumption implies that $\mathcal{B}$ distinguishes the commitments to $(s_j, d_j)_{j\notin\Gamma}$ and the commitments to $(0, \ldots, 0)$ with probability $1/\mathsf{poly}(n)$. Thus, we reach a contradiction. □

Next, we prove Claim 4. Let us say that $\mathcal{A}$ *cheats in a right session* if (i) the session is accepted and (ii) there exists a row that satisfies Condition 1 in the session and whose committed values $(s'_1, d'_1), \ldots, (s'_{10n}, d'_{10n})$ satisfy $V'(\boldsymbol{s'}) \neq V(\boldsymbol{s})$, where $\boldsymbol{s}$ is the committed values of the Stage 2 commitment and $\boldsymbol{s'} = (s'_1, \ldots, s'_{10n})$. Then, to prove Claim 4, we show that in $H^b_h(n,z)$, $\mathcal{A}$ does not cheat in any right session. Recall that $V(\boldsymbol{s})$ is the answer that $\mathcal{O}$ returns to $\mathcal{A}$ in $H^b_h(n,z)$. Hence, if $\mathcal{A}$ does not cheat in a right session, the answer of $\mathcal{O}$ in the session can be computed by extracting the committed values from any row satisfying Condition 1. (Recall that we can extract the committed values of CECom' without over-extraction.)

Claim 5. *In any right session in $H^b_h(n,z)$, $\mathcal{A}$ cheats with at most negligible probability.*

The proof of Claim 5 is given after the proof of Claim 4.

Proof of Claim 4. To show the indistinguishability between the output of $H^b_h(n,z)$ and that of $\widetilde{H}^b_h(n,z)$, we consider an intermediate experiment $\widehat{H}^b_h(n,z)$, which is the same as $\widetilde{H}^b_h(n,z)$ except that at the end of each right session on each thread, $\mathcal{O}$ returns $V(\boldsymbol{s})$ to $\mathcal{A}$ as in $H^b_h(n,z)$ instead of $V'(\boldsymbol{s'})$. From the concurrent extractability of CECom', $\widehat{H}^b_h(n,z)$ outputs $\mathsf{fail}_{\mathrm{ext}}$ with negligible probability.[9] Then, since the property of PTV rewinding strategy guarantees that the main thread of $\widehat{H}^b_h(n,z)$ is identically distributed with the execution of $H^b_h(n,z)$, the output of $H^b_h(n,z)$ and that of $\widehat{H}^b_h(n,z)$ are statistically indistinguishable. In addition, since each thread in $\widehat{H}^b_h(n,z)$ is identically distributed with the execution of $H^b_h(n,z)$, Claim 5 guarantees that in every session on every thread in $\widehat{H}^b_h(n,z)$, $\mathcal{A}$ cheats with at most negligible probability. Since $\widehat{H}^b_h(n,z)$ and $\widetilde{H}^b_h(n,z)$ differ only in the answers of $\mathcal{O}$, from the definition of cheating, we conclude that the output of $\widehat{H}^b_h(n,z)$ and that of $\widetilde{H}^b_h(n,z)$ are statistically indistinguishable. (Recall that we can extract the committed value of CECom' without over-extraction.) Thus, we conclude that the output of $H^b_h(n,z)$ and that of $\widetilde{H}^b_h(n,z)$ are statistically indistinguishable. □

Now, we prove Claim 5. First, we introduce notations. For any $q \in \mathbb{N}$, we say that a right session has the *end-index* q if this session is the q-th right session that $\mathcal{A}$ completes. Similarly, we say that a right session has the *start-index* q if this session is the q-th right session that $\mathcal{A}$ starts. Note that the end-index of a session is undefined until the session completes, whereas the start-index is defined when the session starts. Jumping ahead, in the proof, we assume for contradiction that there exists an end-index q_1 such that $\mathcal{A}$ cheats in the session having the end-index q_1. Then, since we do not know which session has the end-index q_1 until the session completes, we guess a start-index q_2 such that the session having the start-index q_2 has the end-index q_1.

Proof of Claim 5. Intuitively, $\mathcal{A}$ cannot cheat in any right session because of the hiding property of $\mathsf{CCACom}^{1:1}$ in Stage 1. That is, the hiding property of $\mathsf{CCACom}^{1:1}$ and the cut-and-choose technique guarantees that whenever $\mathcal{A}$ tries to cheat in a right session, the session is rejected. However, there exist two problems.

[9] Actually, this is not trivial, since $\widehat{H}^b_h(n,z)$ involves the computation of $V(s)$, which requires super-polynomial time. Nevertheless, we can show this since (i) the extraction of CECom' fails if and only if the extraction of CECom fail, and (ii) the extraction of CECom fails with negligible probability even when the cheating committer runs in super-polynomial time, as long as the cheating committer starts only polynomially-many sessions.

- Since $\mathcal{A}$ accesses super-polynomial-time oracle $\mathcal{O}$, we cannot directly use the computational hiding property of $\mathsf{CCACom}^{1:1}$. We overcome this problem by emulating $\mathcal{O}$ in polynomial time.

- $\mathcal{A}$ may cheat in a right session by using the messages received in the left session, in which the left committer "cheat." We overcome this problem by using the one-one CCA-security of $\mathsf{CCACom}^{1:1}$ instead of the hiding property of $\mathsf{CCACom}^{1:1}$. (Note that, given the committed value Γ of the Stage 1 commitment on the left, we can emulate the left session in polynomial time.)

To simulate $\mathcal{O}$ in polynomial time, we rewind $\mathcal{A}$. Since we want to use the one-one CCA security of $\mathsf{CCACom}^{1:1}$, we make sure that we do not rewind the $\mathsf{CCACom}^{1:1}$ commitment in the left session and the $\mathsf{CCACom}^{1:1}$ commitment in a right session.

Formally, assume for contradiction that there exists a right session in which $\mathcal{A}$ cheats with non-negligible probability. Then, there exists an end-index q_1 such that (i) $\mathcal{A}$ cheats with at most negligible probability in any right session having the end-index less than q_1, but (ii) $\mathcal{A}$ cheats with non-negligible probability in the session having the end-index q_1.

To reach a contradiction, we first consider the following hybrid experiments $G_h^b(n,z)$ and $\widehat{G}_h^b(n,z)$.

- Experiment $G_h^b(n,z)$ is the same as $H_h^b(n,z)$ except that $G_h^b(n,z)$ halts immediately after $\mathcal{A}$ completes the session having the end-index q_1 (i.e., immediately before $\mathcal{O}$ returns the answer of this session to $\mathcal{A}$). Note that in $G_h^b(n,z)$, $\mathcal{O}$ returns answers to $\mathcal{A}$ only in the right sessions having the end-index less than q_1.

- Experiment $\widehat{G}_h^b(n,z)$ is the same as $G_h^b(n,z)$ except that a start-index q_2 is chosen at random, and in Stage 3 of each right session, the committed values are extracted from all the row satisfying the following Condition 2.

  Condition 2. A row satisfies Condition 2 if and only if the row contains no message of Stage 1 on the left, the k-th row of Stage 3 on the left, and Stage 1 of the right session having the start-index q_2.

  As in $\widetilde{H}_h^b(n,z)$, this extraction is done by using the technique of [LP11]. Thus, none of Stage 1 on the left, the k-th row on the left, and Stage 1 of the right session having the start-index q_2 is rewound. (Later, we use this fact to use the one-one CCA security of $\mathsf{CCACom}^{1:1}$.) If the extraction fails, $\widehat{G}_h^b(n,z)$ outputs $\mathsf{fail}_{\mathrm{ext}}$ and halts. (From the concurrent extractability of CECom', this happens with negligible probability.) Note that, since every session has $2r_{\mathrm{CCA}} + r_{\mathrm{CEC}} + 1$ rows, at least one row satisfies Condition 2 in every session. We also note that in the session having start-index q_2, Condition 2 is equivalent to Condition 1, since every row in the session having the start-index q_2 does not contain the messages of Stage 1 of the right session having the start-index q_2.

As in $H_h^b(n,z)$, on every thread in $\widehat{G}_h^b(n,z)$ $\mathcal{A}$ cheats with at most negligible probability in any session having the end-index less than q_1 and $\mathcal{A}$ cheats with non-negligible probability in the session having the end-index q_1. This is because in PTV rewinding strategy, the transcript of every thread in $\widehat{G}_h^b(n,z)$ is identically distributed with the transcript of $G_h^b(n,z)$, which is identically distributed with a prefix of the transcript of $H_h^b(n,z)$.

Then, we observe that except with negligible probability, on every thread of $\widehat{G}_h^b(n,z)$ we have $V(\boldsymbol{s}) = V'(\boldsymbol{s'})$ in every session having the end-index less than q_1, where $\boldsymbol{s'} =$

$(s'_1, \ldots, s'_{10n})$ is the committed values of the first row satisfying Condition 2 in the session. This is because (i) if a row satisfies Condition 2, it also satisfies Condition 1 and (ii) in any session with the end-index less than q_1, $\mathcal{A}$ cheats with at most negligible probability, and therefore we have $V(\boldsymbol{s}) = V'(\boldsymbol{s''})$, where $\boldsymbol{s''}$ is the committed values of any row satisfying Condition 1.

Motivated by this observation, we define another experiment $\widetilde{G}^b_h(n, z)$ as follows.

- $\widetilde{G}^b_h(n, z)$ is the same as $\widehat{G}^b_h(n, z)$ except that at the end of each right session on each thread, $\mathcal{O}$ returns $V'(\boldsymbol{s'})$ to $\mathcal{A}$ instead of $V(\boldsymbol{s})$, where $\boldsymbol{s'} = (s'_1, \ldots, s'_{10n})$ is the extracted values of the first row satisfying Condition 2 in the session.

As in $\widehat{G}^b_h(n, z)$, on every thread of $\widetilde{G}^b_h(n, z)$, $\mathcal{A}$ cheats with non-negligible probability in the session having the end-index q_1, since the transcript of every thread in $\widehat{G}^b_h(n, z)$ and that of $\widetilde{G}^b_h(n, z)$ are identically distributed except with negligible probability. (Recall that in $\widehat{G}^b_h(n, z)$, oracle $\mathcal{O}$ returns the answer to $\mathcal{A}$ only at the end of the sessions having the end-indexes less than q_1, and we can extract the committed values $\boldsymbol{s'} = (s'_1, \ldots, s'_{10n})$ without over-extraction.)

Since the number of right sessions is at most polynomial, on any thread in $\widetilde{G}^b_h(n, z)$, $\mathcal{A}$ cheats with non-negligible probability in the session having the start-index q_2.

Then, we will reach a contradiction by showing that in the session having the start-index q_2 on the main thread in $\widetilde{G}^b_h(n, z)$, $\mathcal{A}$ cheats with at most negligible probability. Toward this end, we first show the following subclaim, which says that if the session having the start-index q_2 is accepted, then for every row satisfying Condition 1 in the session, the committed values of more than $9.5n$ columns of this row are the valid decommitments of the corresponding Stage 2 commitments.

Subclaim 1. *On the main thread in $\widetilde{G}^b_h(n, z)$, except with negligible probability, if the session having the start-index q_2 is accepted, then for every row satisfying Condition 1 in the session, the committed values $((s'_1, d'_1), \ldots, (s'_{10n}, d'_{10n}))$ of this row satisfy $\left|\left\{j \in [10n] \mid c_j = \mathsf{Com}(s'_j; d'_j)\right\}\right| \geq 9.5n$.*

Proof of Subclaim 1. Assume for contradiction that with non-negligible probability, the session with the start-index q_2 is accepted but there exists a row that satisfies Condition 1 and whose committed values $(s'_1, d'_1), \ldots, (s'_{10n}, d'_{10n})$ satisfy $\left|\left\{j \in [10n] \mid c_j = \mathsf{Com}(s'_j; d'_j)\right\}\right| < 9.5n$. Then, we consider the following PPT adversary $\mathcal{M}$ against the one-one CCA security of $\mathsf{CCACom}^{1:1}$.

- $\mathcal{M}$ internally invokes $\mathcal{A}$ and emulates $\widetilde{G}^b_h(n, z)$ as follows. In the left session, $\mathcal{M}$ forwards the Stage 1 commitment from $\mathcal{A}$ to the committed-value oracle $\mathcal{O}$, and receives Γ from $\mathcal{O}$. Then, $\mathcal{M}$ honestly emulates the left session by using Γ. In right sessions, $\mathcal{M}$ honestly emulates every session except that in the session having the start-index q_2 $\mathcal{M}$ receives either a commitment to Γ_0 or a commitment to Γ_1 from the external committer (where $\Gamma_0, \Gamma_1 \subset [10n]$ are random subsets of size n) and forwards this commitment to $\mathcal{A}$ as the Stage 1 commitment. (Note that the commitments that $\mathcal{M}$ externally forwards are not rewound in $\widetilde{G}^b_h(n, z)$.) $\mathcal{M}$ continues the emulation until $\widetilde{G}^b_h(n, z)$ ends or Stage 3 of the right session having the start-index q_2 ends. $\mathcal{M}$ outputs 1 if in the right session having the start-index q_2, there exists a row that satisfies Condition 1 and whose extracted values $(s''_1, d''_1), \ldots, (s''_{10n}, d''_{10n})$ satisfies that (i) for every $j \in \Gamma_1$, $c_j = \mathsf{Com}(s''_j; d''_j)$ and (ii) $\left|\left\{j \in [10n] \mid c_j = \mathsf{Com}(s''_j; d''_j)\right\}\right| < 9.5n$. Otherwise, $\mathcal{M}$ outputs 0.

When $\mathcal{M}$ receives a commitment to Γ_0, $\mathcal{M}$ outputs 1 with at most exponentially small probability. That is, when $\left|\left\{j \in [10n] \mid c_j = \mathsf{Com}(s'_j; d'_j)\right\}\right| < 9.5n$, since the internal $\mathcal{A}$ receives no information about Γ_1 and since CECom' is extractable without over-extraction, there exists an index $j \in \Gamma_1$ such that $c_j \neq \mathsf{Com}(s'_j; d'_j)$ except with exponentially small probability. On the other hand, when $\mathcal{M}$ receives a commitment to Γ_1, our assumption guarantees that $\mathcal{M}$ outputs 1 with non-negligible probability. Thus, we reach a contradiction. □

We go back to the proof of Claim 5. As noted above, we reach a contradiction by showing that on the main thread in $\widetilde{G}^b_h(n,z)$, $\mathcal{A}$ cheats with at most negligible probability in the right session having the start-index q_2. That is, we show that when the right session having the start-index q_2 is accepted, we have $V(\boldsymbol{s}) = V'(\boldsymbol{s'})$ except with negligible probability, where $\boldsymbol{s'} = (s'_1, \ldots, s'_{10n})$ is the committed values of any row satisfying Condition 1 in the session. From the binding property of Com, the transcript of the main thread in $\widetilde{G}^b_h(n,z)$ uniquely determines the committed values $\boldsymbol{s} = (s_1, \ldots, s_{10n})$ of the Stage 2 commitments in the right session having the start-index q_2. Then, for any row satisfying Condition 1 in the session, we consider the following two cases.

- First, we consider the case that $\boldsymbol{s}$ is 0.9-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10n})$. In this case, since Subclaim 1 guarantees that the committed shares $\boldsymbol{s'} = (s'_1, \ldots, s'_{10n})$ of the row is 0.95-close to $\boldsymbol{s}$ except with negligible probability, $\boldsymbol{s''}$ is 0.85-close to $\boldsymbol{w}$ except with negligible probability. Thus, from the definitions of $V(\cdot)$ and $V'(\cdot)$, we have $V(\boldsymbol{s}) = V'(\boldsymbol{s'})$ except with negligible probability.

- Next, we consider the case that $\boldsymbol{s}$ is 0.1-far from any valid codeword. In this case, we have $V(\boldsymbol{s}) = \perp$. If $\boldsymbol{s'}$ is 0.2-far from any valid codeword, we have $V'(\boldsymbol{s'}) = \perp$ as well. Thus, it suffices to consider the case that $\boldsymbol{s'}$ is 0.8-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10n})$. Then, since $\boldsymbol{s}$ and $\boldsymbol{w}$ are 0.1-far and $\boldsymbol{s}$ and $\boldsymbol{s'}$ are 0.95-close, $\boldsymbol{s'}$ and $\boldsymbol{w}$ is 0.05-far. Thus, if we have $s'_j = w_j$ for every $j \in \Gamma$ with non-negligible probability, we can break the one-one CCA security of $\mathsf{CCACom}^{1:1}$ in the same way as in the proof of Subclaim 1. (Note that we can efficiently compute $\boldsymbol{w}$ from $\boldsymbol{s'}$.) Thus, there exists an index $j \in \Gamma$ such that $s'_j \neq w_j$ except with negligible probability. Then, since for every $j \in \Gamma$, s'_j equals the values revealed in Stage 5 whenever the session is accepted, from the definition of $V'(\cdot)$, we have $V'(\boldsymbol{s'}) = \perp$ except with negligible probability.

We therefore conclude that $\mathcal{A}$ cheats in the session having start-index q_2 with at most negligible probability. Thus we reach a contradiction. □

5.1.2 Proof of Claim 2

Claim 2 can be proven in essentially the same way as Claim 1. In particular, $H^b_\eta(n,z)$ and $H^b_{\eta+1}(n,z)$ differ only in the committed values of the Stage 2 commitments, whereas $H^b_{k-1}(n,z)$ and $H^b_k(n,z)$ differ only in the committed values of the k-th row commitments. We can therefore prove Claim 2 by modifying the proof of Claim 1 accordingly. We omit the formal proof.

5.1.3 Proof of Claim 3

Proof of Claim 3. Since $H^b_k(n,z)$ outputs fail only if the Stage 1 commitment has more than one committed value in the left session, we prove this claim by using the binding property

of $\mathsf{CCACom}^{1:1}$. The problem is that since $\mathcal{A}$ accesses the committed-value oracle $\mathcal{O}$, which runs in super-polynomial time, we cannot directly use the strong computational binding property of $\mathsf{CCACom}^{1:1}$. We overcome this problem, again, by rewinding $\mathcal{A}$ and emulating $\mathcal{O}$ in polynomial time. The proof is similar to the proof of Claim 1.

Formally, assume for contradiction that H_k^b outputs fail with non-negligible probability. Then, with non-negligible probability, the Stage 1 commitment on the left has more than one committed value.

First, we consider hybrid experiment $\widetilde{H'}_k^b(n,z)$, which is the same as $H_k^b(n,z)$ except for the following.

- In Stage 3 of each right session, let us consider the following condition.

  Condition $1'$. A row satisfies Condition $1'$ if and only if the row contains no message of Stage 1 on the left.

  Then, the committed values of every row satisfying Condition $1'$ are extracted by using the concurrent extractability of CECom' without rewinding Stage 1 on the left.

- For any $\boldsymbol{t} = (t_1, \ldots, t_{10n})$, we define $V'(\boldsymbol{t})$ in the same way as in $\widetilde{H}_h^b(n,z)$ in the proof of Claim 1. Then, at the end of each right session, $\mathcal{O}$ returns $V'(\boldsymbol{s'})$ to $\mathcal{A}$ instead of $V(\boldsymbol{s})$, where $\boldsymbol{s'} = (s'_1, \ldots, s'_{10n})$ is the extracted values of the first row satisfying Condition $1'$ in this session.

- $H_k^b(n,z)$ terminates immediately after Stage 1 ends in the left session.

From essentially the same proof as that of Claim 4, we have the following claim.

Claim 6. $\left\{\mathsf{H}_k^b(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ *and* $\left\{\widetilde{\mathsf{H}'}_k^b(n,z)\right\}_{n\in\mathbb{N},z\in\{0,1\}^*}$ *are statistically indistinguishable.*

From Claim 6 and our assumption, the Stage 1 commitment has more than one committed value in the left session in $\widetilde{H'}_k^b(n,z)$ with non-negligible probability. Since the Stage 1 commitment on the left is not rewound and since $\widetilde{H'}_k^b(n,z)$ can be executed in polynomial time, this contradicts to the strong computational binding property of $\mathsf{CCACom}^{1:1}$. □

5.2 Proof of Robustness

Like the robustness of previous CCA-secure commitments [CLP10, LP12], the robustness of CCACom can be shown by using the techniques in the proof of its CCA security.

Proof of Lemma 4. We show that there exists a PPT simulator $\mathcal{S}$ such that for any PPT adversary $\mathcal{A}$ and any κ-round PPT ITM B, the following are computationally indistinguishable.

- $\left\{\mathsf{out}_{B,\mathcal{A}^{\mathcal{O}}}[\langle B(y), \mathcal{A}^{\mathcal{O}}(z)\rangle(1^n, x)]\right\}_{n\in\mathbb{N},x,y,z\in\{0,1\}^n}$
- $\left\{\mathsf{out}_{B,\mathcal{S}^{\mathcal{A}}}[\langle B(y), \mathcal{S}^{\mathcal{A}}(z)\rangle(1^n, x)]\right\}_{n\in\mathbb{N},x,y,z\in\{0,1\}^n}$

Given oracle access to $\mathcal{A}$, simulator $\mathcal{S}$ simulates the interaction between B and $\mathcal{A}^{\mathcal{O}}$ as follows. On the left, $\mathcal{S}$ forwards messages from B to $\mathcal{A}$ and forwards those from $\mathcal{A}$ to B. On the right, $\mathcal{S}$ honestly simulates each session between $\mathcal{A}$ and $\mathcal{O}$ except for the following.

- In Stage 3, $\mathcal{S}$ extracts $\widehat{\boldsymbol{s}} = (\widehat{s}_1, \ldots, \widehat{s}_{10n})$ from every row containing no message of the left session by using the technique of [LP11]. There must exist such rows, since CCACom has $\widetilde{O}(\log n)$ rows in Stage 3. Note that in the extraction, the left session is not rewound.
- At the end of the session, $\mathcal{S}$ returns $V'(\boldsymbol{s'})$ to $\mathcal{A}$, where $\boldsymbol{s'}$ is the extracted values of the first row containing no message of the left session.

We show that $\mathcal{S}$ correctly simulates the interaction between $\mathcal{B}$ and $\mathcal{A}^{\mathcal{O}}$. First, as in the proof of Claim 5, we can show that when $\mathcal{A}^{\mathcal{O}}$ interacts with B, in any right session between $\mathcal{A}$ and $\mathcal{O}$, we have $V(\boldsymbol{s}) \neq V'(\boldsymbol{s'})$ with at most negligible probability, where $\boldsymbol{s'}$ is the committed values of any row containing no message of the left session. (we use the hiding property of $\mathsf{CCACom}^{1:1}$ instead of one-one CCA security). Then, as in the proof of Claim 4, we can show that the view of the internal $\mathcal{A}$ (in $\mathcal{S}$) is statistically close to the view of $\mathcal{A}^{\mathcal{O}}$ that interacts with B. □

6 Black-Box Composable MPC Protocol

In this section, we show our black-box construction of a general MPC protocol. Our protocol is secure in angel-based UC framework. Roughly speaking, this framework (called $\mathcal{H}$-EUC framework) is the same as the UC framework except that both the adversary and the environment in the real and ideal worlds have access to a super-polynomial-time functionality $\mathcal{H}$ called a *angel* (or a *helper*). For details, see [PS04, CLP10].

We use the results of [CLP10] and [LP12]. Let $\langle C, R\rangle$ be any $r_{\mathrm{CCA}}(n)$-round robust CCA-secure commitment scheme, $\langle S, R\rangle$ be any $r_{\mathrm{OT}}(n)$-round semi-honest oblivious transfer protocol, and $\mathcal{H}$ be a helper that breaks $\langle C, R\rangle$ in essentially the same way as the committed-value oracle of $\langle C, R\rangle$ does. Then, Lin and Pass [LP12] showed that there exists a black-box $O(\max(r_{\mathrm{OT}}(n), r_{\mathrm{CCA}}(n)))$-round protocol that securely realizes the ideal oblivious transfer functionality $\mathcal{F}_{OT}$ in the $\mathcal{H}$-EUC framework.

Theorem 2 ([LP12])**.** *Assume the existence of an $r_{\mathrm{CCA}}(n)$-round robust CCA-secure commitment scheme $\langle C, R\rangle$ and the existence of an $r_{\mathrm{OT}}(n)$-round semi-honest oblivious transfer protocol $\langle S, R\rangle$. Then, there exists an $O(\max(r_{\mathrm{CCA}}(n), r_{\mathrm{OT}}(n)))$-round protocol that $\mathcal{H}$-EUC-realizes $\mathcal{F}_{OT}$. Furthermore, this protocol uses $\langle C, R\rangle$ and $\langle S, R\rangle$ only in a black-box way.*

In [CLP10], Canetti et al. showed the following.

Theorem 3 ([CLP10])**.** *For every well-formed functionality $\mathcal{F}$, there exists a constant-round $\mathcal{F}_{OT}$-hybrid protocol that $\mathcal{H}$-EUC-realizes $\mathcal{F}$.*

Then, we obtain the following theorem by combining Theorems 1, 2, and 3.

Theorem 4. *Assume the existence of r_{OT}-round semi-honest oblivious transfer protocols. Then, there exists a super-polynomial-time helper $\mathcal{H}$ such that for every well-formed functionality $\mathcal{F}$, there exists a $\max(\widetilde{O}(\log^2 n), O(r_{\mathrm{OT}}(n)))$-round protocol that $\mathcal{H}$-EUC-realizes $\mathcal{F}$. Furthermore, this protocol uses the underlying oblivious transfer protocol only in a black-box way.*

References

[BS05] Boaz Barak and Amit Sahai. How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In *FOCS*, pages 543–552, 2005.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.

[CDSMW08] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In *TCC*, pages 427–444, 2008.

[CDSMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In *TCC*, pages 387–402, 2009.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *CRYPTO*, pages 19–40, 2001.

[CKL03] Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In *EUROCRYPT*, pages 68–86, 2003.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503, 2002.

[CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. Adaptive hardness and composable security in the plain model from standard assumptions. In *FOCS*, pages 541–550, 2010.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[GGJS12] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Concurrently secure computation in constant rounds. In *EUROCRYPT*, pages 99–116, 2012.

[GLOV12] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *FOCS*, pages 51–60, 2012.

[GLP+12] Vipul Goyal, Huijia Lin, Omkant Pandey, Rafael Pass, and Amit Sahai. Round-efficient concurrently composable secure computation via a robust extraction lemma. Cryptology ePrint Archive, Report 2012/652, 2012. `http://eprint.iacr.org/`.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

[Goy11] Vipul Goyal. Constant round non-malleable protocols using one way functions. In *STOC*, pages 695–704, 2011.

[Hai08] Iftach Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In *TCC*, pages 412–426, 2008.

[HIK+11] Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions of protocols for secure computation. *SIAM J. Comput.*, 40(2):225–266, 2011.

[IKLP06] Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions for secure computation. In *STOC*, pages 99–108, 2006.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In *CRYPTO*, pages 572–591, 2008.

[Kiy14] Susumu Kiyoshima. Round-efficient black-box construction of composable multi-party computation. In *CRYPTO*, 2014. To appear.

[KMO14] Susumu Kiyoshima, Yoshifumi Manabe, and Tatsuaki Okamoto. Constant-round black-box construction of composable multi-party computation protocol. In *TCC*, pages 343–367, 2014.

[Lin11] Huijia Lin. *Concurrent Security.* PhD thesis, Cornell University, 2011.

[LP11] Huijia Lin and Rafael Pass. Concurrent non-malleable zero knowledge with adaptive inputs. In *TCC*, pages 274–292, 2011.

[LP12] Huijia Lin and Rafael Pass. Black-box constructions of composable protocols without set-up. In *CRYPTO*, pages 461–478, 2012.

[LPTV10] Huijia Lin, Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable zero knowledge proofs. In *CRYPTO*, pages 429–446, 2010.

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In *TCC*, pages 571–588, 2008.

[MMY06] Tal Malkin, Ryan Moriarty, and Nikolai Yakovenko. Generalized environmental security from number theoretic assumptions. In *TCC*, pages 343–359, 2006.

[MOSV06] Daniele Micciancio, Shien Jin Ong, Amit Sahai, and Salil P. Vadhan. Concurrent zero knowledge without complexity assumptions. In *TCC*, pages 1–20, 2006.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *EUROCRYPT*, pages 160–176, 2003.

[PLV12] Rafael Pass, Huijia Lin, and Muthuramakrishnan Venkitasubramaniam. A unified framework for UC from only OT. In *ASIACRYPT*, pages 699–717, 2012.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *FOCS*, pages 366–375, 2002.

[PS04] Manoj Prabhakaran and Amit Sahai. New notions of security: achieving universal composability without trusted setup. In *STOC*, pages 242–251, 2004.

[PTV12] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent zero knowledge, revisited. *J. Cryptology*, pages 1–22, 2012.

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *TCC*, pages 403–418, 2009.

[RK99] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In *EUROCRYPT*, pages 415–431, 1999.

[Wee10] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *FOCS*, pages 531–540, 2010.

A Technical Detail for the Proof of Claim 1.

Let $(m_1, \ldots, m_{r_{\text{CCA}}+r_{\text{CEC}}})$ be the messages that $\mathcal{A}$ sends in Stage 1 on the left and the k-th row on the left, and let a_i be the reply to m_i from the left committer. Then, the execution of $\mathcal{A}$ is equivalent to the sequential execution of the following adversaries $\mathcal{A}_1, \ldots, \mathcal{A}_{r_{\text{CCA}}+r_{\text{CEC}}+1}$.

- On input a_{i-1} and a partial view $\mathcal{V}_{i-1}$ of $\mathcal{A}$ up until $\mathcal{A}$ sends m_{i-1} ($a_0 = \epsilon$ and $\mathcal{V}_0 = \bot$), $\mathcal{A}_i$ emulates the execution of $\mathcal{A}$ from $\mathcal{V}_{i-1}$ by feeding $\mathcal{V}_{i-1}$ and a_{i-1} to $\mathcal{A}$ and forwarding every message from $\mathcal{A}$ externally. When $\mathcal{A}$ halts or outputs m_i, $\mathcal{A}_i$ outputs the current view $\mathcal{V}_i$ of $\mathcal{A}$ and halts.

Then, we extract the committed values of every row satisfying Condition 1 by sequentially executing $\mathcal{A}_1, \ldots, \mathcal{A}_{r_{\text{CCA}}+r_{\text{CEC}}+1}$ and extracting the committed values from every row that is entirely contained in the execution of each $\mathcal{A}_i$. Clearly, neither Stage 1 on the left nor the k-th row on the left is rewound.

B One-One CCA Commitment for Long Tags from Parallel CCA Commitment for Short Tags

Lemma 5. *Let $r(\cdot)$ and $t(\cdot)$ be arbitrary functions such that $t(n) = O(\log n)$, and let* CCACom *be an $r(n)$-round commitment scheme that satisfies strong computational binding property and parallel CCA security for tags of length $t(n)$. Then, there exists an $r(n)$-round commitment scheme* $\mathsf{CCACom}^{1:1}$ *that satisfies strong computational binding property and one-one CCA security for tags of length $2^{t(n)-1}$. Furthermore, if* CCACom *uses the underlying one-way function only in a black-box way, then* $\mathsf{CCACom}^{1:1}$ *uses the underlying one-way function only in a black-box way.*

Proof. $\mathsf{CCACom}^{1:1}$ is shown in Figure 7. The strong computational binding property follows from that of CCACom. Thus, it remains to show that $\mathsf{CCACom}^{1:1}$ is one-one CCA secure.

We show that for any PPT adversary $\mathcal{A}$ that interacts with $\mathcal{O}$ only in a single session, the following are computationally indistinguishable:

Commit Phase

The committer C and the receiver R receive common inputs 1^n and $\mathsf{tag} \in \{0,1\}^{2^{t(n)-1}}$. To commit to $v \in \{0,1\}^n$, the committer C chooses random $v_1, \ldots, v_{2^{t(n)-1}} \in \{0,1\}^n$ such that $v = \bigoplus_j v_j$, and for each $j \in [2^{t(n)-1}]$ in parallel, $\mathcal{C}$ commits to v_j by using CCACom with tag (j, tag_j), where tag_j is the j-th bit of tag.

Decommit Phase

To decommit, C sends v to R and decommits all the CCACom commitments.

Figure 7: One-one CCA-secure commitment scheme $\mathsf{CCACom}^{1:1}$.

- $\left\{\mathsf{IND}_0(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)\right\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$
- $\left\{\mathsf{IND}_1(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)\right\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$

Without loss of generality, we assume that the tag that $\mathcal{A}$ chooses in the right session is always different from the tag that $\mathcal{A}$ chooses in the left session.

Assume for contradiction that there exist a PPT distinguisher $\mathcal{D}$ and a polynomial $p(\cdot)$ such that for infinitely many n, there exists $z \in \{0,1\}^*$ such that $\mathcal{D}$ distinguishes $\mathsf{IND}_1(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)$ from $\mathsf{IND}_0(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)$ with probability at least $1/p(n)$. In the following, we fix any such n and z.

Let us consider the following PPT adversary $\mathcal{B}$ against CCA security of CCACom. $\mathcal{B}$ internally invokes $\mathcal{A}$ and simulates $\mathrm{IND}_0(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)$ for $\mathcal{A}$ as follows. First, $\mathcal{B}$ chooses random $j^* \in [2^{t(n)-1}]$, and for each $j \in [2^{t(n)-1}] \setminus \{j^*\}$, $\mathcal{B}$ chooses random $v_j \in \{0,1\}^n$. Then, in the left session, when $\mathcal{A}$ outputs challenge values $m_0, m_1 \in \{0,1\}^n$ and tag $\mathsf{tag} = (\mathsf{tag}_1, \ldots, \mathsf{tag}_{2^{t(n)-1}})$, $\mathcal{B}$ sets $v_{j^*}^{(b)} := m_b \oplus \bigoplus_{j\neq j^*} v_j$ for each $b \in \{0,1\}$ and sends challenge $v_{j^*}^{(0)}, v_{j^*}^{(1)}$ and tag $(j^*, \mathsf{tag}_{j^*}) \in \{0,1\}^{t(n)}$ to the external left committer. When $\mathcal{B}$ receives a CCACom commitment from the left committer (the committed value is either $v_{j^*}^{(0)}$ or $v_{j^*}^{(1)}$), $\mathcal{B}$ forwards it to $\mathcal{A}$. At the same time, $\mathcal{B}$ generates CCACom commitments to $(v_j)_{j\neq j^*}$ and sends them to $\mathcal{A}$. In the right session, when $\mathcal{A}$ outputs tag $\widetilde{\mathsf{tag}}$, $\mathcal{B}$ terminates and outputs fail if $\mathsf{tag}_{j^*} = \widetilde{\mathsf{tag}}_{j^*}$. Otherwise, $\mathcal{B}$ forwards a $\mathsf{CCACom}^{1:1}$ commitment from $\mathcal{A}$ to $\mathcal{O}$ as $2^{t(n)-1}$ parallel commitments of CCACom with tags $\left\{(j, \widetilde{\mathsf{tag}}_j)\right\}_{j=1}^{2^{t(n)-1}}$. Then, $\mathcal{B}$ receives $(v_1, \ldots, v_{2^{t(n)-1}})$ from $\mathcal{O}$, and if $v_j \neq \bot$ for all $j \in [2^{t(n)-1}]$, $\mathcal{B}$ returns $v := \bigoplus_j v_j$ to $\mathcal{A}$. If $v_j = \bot$ for some j, $\mathcal{B}$ returns $\bot$ to $\mathcal{A}$. Finally, $\mathcal{B}$ outputs $\mathcal{D}(y)$, where y is the output of the simulated $\mathrm{IND}_0(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)$.

We reach a contradiction by showing that $\mathcal{B}$ breaks the CCA security of CCACom with probability $1/p(n)\mathsf{poly}(n)$. For $b \in \{0,1\}$, let abort_b be the event that $\mathcal{B}$ outputs fail in $\mathrm{IND}_b(\mathsf{CCACom}, \mathcal{B}, n, z)$. Then, from the hiding property of CCACom, we have

$$|\Pr[\mathsf{abort}_0] - \Pr[\mathsf{abort}_1]| \leq \mathsf{negl}(n) \ .$$

In addition, since we always have $\mathsf{tag} \neq \widetilde{\mathsf{tag}}$, for each $b \in \{0,1\}$ we have

$$\Pr[\neg\mathsf{abort}_b] \geq \frac{1}{2^{t(n)-1}} \geq \frac{1}{\mathsf{poly}(n)} \ .$$

If $\mathcal{B}$ does not output fail, $\mathcal{B}$ perfectly simulates $\mathrm{IND}_0(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)$ or $\mathrm{IND}_1(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)$. In addition, if $\mathcal{B}$ does not output fail, for each $j \in [2^{t(n)-1}]$ the tag $(j, \widetilde{\mathsf{tag}}_j)$, which was used on the right, is different from the tag $(j^*, \mathsf{tag}_{j^*})$, which was used on the left. Thus, we have

$$
\begin{aligned}
&\left| \begin{array}{l} \Pr\left[\mathsf{IND}_0(\mathsf{CCACom}, \mathcal{B}, n, z) = 1\right] \\ -\Pr\left[\mathsf{IND}_1(\mathsf{CCACom}, \mathcal{B}, n, z) = 1\right] \end{array} \right| \\
&= \left| \begin{array}{l} \Pr\left[\mathsf{IND}_0(\mathsf{CCACom}, \mathcal{B}, n, z) = 1 \bigwedge \neg\mathsf{abort}_0\right] \\ -\Pr\left[\mathsf{IND}_1(\mathsf{CCACom}, \mathcal{B}, n, z) = 1 \bigwedge \neg\mathsf{abort}_1\right] \end{array} \right| \\
&\geq \left| \begin{array}{l} \Pr\left[\mathsf{IND}_0(\mathsf{CCACom}, \mathcal{B}, n, z) = 1 \mid \neg\mathsf{abort}_0\right] \\ -\Pr\left[\mathsf{IND}_1(\mathsf{CCACom}, \mathcal{B}, n, z) = 1 \mid \neg\mathsf{abort}_1\right] \end{array} \right| \times \frac{1}{\mathsf{poly}(n)} \\
&= \left| \begin{array}{l} \Pr\left[\mathcal{D}(\mathsf{IND}_0(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)) = 1\right] \\ -\Pr\left[\mathcal{D}(\mathsf{IND}_1(\mathsf{CCACom}^{1:1}, \mathcal{A}, n, z)) = 1\right] \end{array} \right| \times \frac{1}{\mathsf{poly}(n)} \\
&\geq \frac{1}{p(n)\mathsf{poly}(n)} \ .
\end{aligned}
$$

□