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 $r_{\text{OT}}(n)$, the round complexity of our protocol is $\max(\widetilde{O}(\log^2 n), O(r_{\text{OT}}(n)))$. 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 $O(\max(n^{\epsilon}, r_{\text{OT}}(n)))$-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. [11], 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 [17], constructing black-box constructions is important for both theoretical and practical reasons. Theoretically, it is important because understanding whether non-black-box use of cryptographic primitives is necessary for a

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

J.A. Garay and R. Gennaro (Eds.): CRYPTO 2014, Part II, LNCS 8617, pp. 351–368, 2014.

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. [17] 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 [15], 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 [16]. Subsequently, Wee [30] reduced the round complexity of [17] to $O(\log^* n)$, and Goyal [12] 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 [17,30,12] 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. 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 they 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 [2] 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. [6] 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 [18] and in the $\mathcal{F}_{\mathrm{COM}}$-hybrid model [8] (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 [3,4].

To achieve composable security in the plain model, Prabhakaran and Sahai [29] 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* [26,1,10,27]. 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 [29] presented a general MPC protocol that satisfies angel-based UC security in the plain model based on new assumptions. Subsequently, Malkin et al. [24] 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. [5] constructed a polynomial-round general MPC protocol in angel-based UC security assuming the existence of enhanced trapdoor permutations. Subsequently, Lin [20] and Goyal et al. [14] 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, however, non-black-box. That is, in the protocols of [5,20,14], the underlying primitives are used in a non-black-box way.

Black-Box Constructions of Composable Protocols. Lin and Pass [22] 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 $O(\max(n^\epsilon, r_{\mathrm{OT}}(n)))$, where $\epsilon > 0$ is an arbitrary constant and $r_{\mathrm{OT}}(n)$ 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, Kiyoshima et al. [19] constructed a constant-round protocol from constant-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

- logarithmic-round non-black-box constructions under a standard polynomial-time hardness assumption [20,14],

- a polynomial-round black-box construction under a standard polynomial-time hardness assumption [22], and
- constant-round black-box or non-black-box constructions under standard super-polynomial-time hardness assumptions [20,14,19].

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 (logarithmic rounds [20,14]) and that of the black-box protocols (polynomial rounds [22]). 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 greatly 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* [5,20,22,14,19] 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 [22] 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 [22].

1.2 Outline

In Section 2, we give an overview of our CCA secure commitment scheme. Due to lack of space, we defer formal proofs to the full version.

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, we can show 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. Then, 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 [25].

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 [7,8,30,22,19]).

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. Then, 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.

[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*.

4. The receiver accepts the commitment if and only if the decommitments 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).

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 wExtCom that was introduced by Goyal et al. [13]. The commit phase of 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 wExtCom is extractable only in a weak sense—extractions may fail with probability at most $1/2$—but 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 wExtCom satisfies the following property: After the `commit`

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

stage, if the 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 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 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 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 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 wExtCom of at most $9n$ columns, then there are n columns in which the wExtCom commitment is accepted with probability at most $1/2 + \mathsf{negl}(n)$. Thus, the probability that all wExtCom commitments are accepted is negligible, and therefore the commitment is accepted with at most negligible probability.)
- Thus, from the property of wExtCom, we can extract the committed values of wExtCom 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 wExtCom commitment is valid and its committed value is a valid decommitment of the corresponding CECom commitment. Note that since the committed values of wExtCom commitments can be extracted without over-extraction, we can show that the cheating committer cannot give invalid 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.

A formal description of our concurrently extractable commitment scheme CECom′ is shown in Fig. 1. (For technical reasons, we set the number of columns to $40n$.) In Appendix A, we give a formal proof for the fact that in most columns on an accepted commitment, the wExtCom commitment is valid and its committed value is a valid decommitment of the CECom commitment. The formal proof is more complicated than the above proof sketch because we execute the wExtCom commitments in parallel and thus the columns are not independent of each other. The proof of this fact is the most complicated part of the analysis of CECom′: Given this fact, we can show the concurrent extractability by using the technique used in the previous works [7,8,30,22,19].

To commit to $v \in \{0,1\}^n$, 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 v. 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.

Fig. 1. A concurrently commitment scheme CECom'

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.[4]

We construct a black-box $O(\log n)$-round one-one CCA-secure commitment scheme by simplifying the CCA-secure commitment scheme of [22] and using the *DDN* $\log n$ *trick* [9,23], 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 [22]. Roughly speaking, the scheme of [22] consists of polynomially-many *rows*—each row is a parallel execution of (a part of) the trapdoor commitment scheme of [28]—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 [22], 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

[4] 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.*

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.
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 normal 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 [5,22,20,14] 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 [19].)

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 [5,22,20], 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 [21]. 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 actual argument 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. For details, see the full version.

Comparison with the CCA-secure commitment scheme of [19]. The above CCA-secure commitment scheme is based on the CCA-secure commitment scheme of [19], which is constructed from one-way functions that are secure against subexponential-time adversaries. The scheme of [19] 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 [19] 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. (Note that 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 [19] heavily depends on the subexponentially hard security of the underlying commitment schemes. Roughly speaking, we weaken the assumption of [19] 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 [19] and ours, the details of the security proofs have a lot of difference.

References

1. Barak, B., Sahai, A.: How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In: FOCS, pp. 543–552 (2005)
2. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS, pp. 136–145 (2001)
3. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (2001)
4. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 68–86. Springer, Heidelberg (2003)
5. Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: FOCS, pp. 541–550 (2010)
6. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: STOC, pp. 494–503 (2002)
7. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Black-box construction of a non-malleable encryption scheme from any semantically secure one. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 427–444. Springer, Heidelberg (2008)
8. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Simple, black-box constructions of adaptively secure protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 387–402. Springer, Heidelberg (2009)
9. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM J. Comput. 30(2), 391–437 (2000)
10. Garg, S., Goyal, V., Jain, A., Sahai, A.: Concurrently secure computation in constant rounds. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 99–116. Springer, Heidelberg (2012)
11. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC. pp. 218–229 (1987)
12. Goyal, V.: Constant round non-malleable protocols using one way functions. In: STOC, pp. 695–704 (2011)
13. Goyal, V., Lee, C.K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: FOCS, pp. 51–60 (2012)
14. Goyal, V., Lin, H., Pandey, O., Pass, R., Sahai, A.: Round-efficient concurrently composable secure computation via a robust extraction lemma. Cryptology ePrint Archive, Report 2012/652 (2012), `http://eprint.iacr.org/`
15. Haitner, I.: Semi-honest to malicious oblivious transfer—the black-box way. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 412–426. Springer, Heidelberg (2008)
16. Haitner, I., Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions of protocols for secure computation. SIAM J. Comput. 40(2), 225–266 (2011)
17. Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions for secure computation. In: STOC, pp. 99–108 (2006)
18. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008)
19. Kiyoshima, S., Manabe, Y., Okamoto, T.: Constant-round black-box construction of composable multi-party computation protocol. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 343–367. Springer, Heidelberg (2014)
20. Lin, H.: Concurrent Security. Ph.D. thesis, Cornell University (2011)

21. Lin, H., Pass, R.: Concurrent non-malleable zero knowledge with adaptive inputs. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 274–292. Springer, Heidelberg (2011)
22. Lin, H., Pass, R.: Black-box constructions of composable protocols without set-up. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 461–478. Springer, Heidelberg (2012)
23. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent non-malleable commitments from any one-way function. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 571–588. Springer, Heidelberg (2008)
24. Malkin, T., Moriarty, R., Yakovenko, N.: Generalized environmental security from number theoretic assumptions. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 343–359. Springer, Heidelberg (2006)
25. Micciancio, D., Ong, S.J., Sahai, A., Vadhan, S.P.: Concurrent zero knowledge without complexity assumptions. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 1–20. Springer, Heidelberg (2006)
26. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (2003)
27. Pass, R., Lin, H., Venkitasubramaniam, M.: A unified framework for UC from only OT. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 699–717. Springer, Heidelberg (2012)
28. Pass, R., Wee, H.: Black-box constructions of two-party protocols from one-way functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 403–418. Springer, Heidelberg (2009)
29. Prabhakaran, M., Sahai, A.: New notions of security: achieving universal composability without trusted setup. In: STOC, pp. 242–251 (2004)
30. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: FOCS, pp. 531–540 (2010)

A Formal Proof

In this section, we give a formal proof for the fact that in most columns on an accepted commitment of CECom', the $\mathsf{wExtCom}$ commitment is valid and its committed value is a valid decommitment of the CECom commitment. This is the most complicated part of the analysis of CECom': Given this fact, we can show the concurrent extractability by using the technique used in the previous works [7,8,30,22,19].

Lemma 1. *Let $\mathcal{C}^*$ be any cheating committer that concurrently executes many sessions of the commit phase of CECom'. Then, the following holds except with negligible probability: In more than $38n$ columns on every accepted session, the $\mathsf{wExtCom}$ commitment is valid and its committed value is a valid decommitment of the CECom commitment.*

Proof. 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 CECom commitment; otherwise, the column is *inconsistent*. We say that C^* *cheats* in a session if (i) every 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 the lemma, it suffices to show that in every session the probability that C^* cheats is negligible.

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 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 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 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 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 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

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

(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

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

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

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

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

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

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

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

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 $\Pr[\mathsf{accept}] \geq \Pr[\mathsf{cheat}]$ from the definitions.) Then, for any $j \in [40n]$ and $b \in \{0,1\}$,

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

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

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

Let

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

Then we have

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

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 $\Pr[ch_j = b] = 1/2$, 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

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

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

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

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.

□