A Unified Approach to Constructing Black-box UC Protocols in Trusted Setup Models

Susumu Kiyoshima[1], Huijia Lin[2], and
Muthuramakrishnan Venkitasubramaniam[3]

[1] NTT Secure Platform Laboratories, Tokyo, Japan
kiyoshima.susumu@lab.ntt.co.jp
[2] University of California, Santa Barbara, CA, USA
rachel.lin@cs.ucsb.edu
[3] University of Rochester, NY, USA
muthuv@cs.rochester.edu

Abstract. We present a unified framework for obtaining *black-box* constructions of *Universal Composable* (UC) protocol in trusted setup models. Our result is analogous to the unified framework of Lin, Pass, and Venkitasubramaniam [STOC'09, Asiacrypt'12] that, however, only yields *non-black-box* constructions of UC protocols. Our unified framework shows that to obtain black-box constructions of UC protocols, it suffices to implement a special purpose commitment scheme that is, in particular, concurrently extractable using a given trusted setup. Using our framework, we improve black-box constructions in the common reference string and tamper-proof hardware token models by weakening the underlying computational and setup assumptions.

1 Introduction

Secure multi-party computation (MPC) protocols enable a set of m mutually distrustful parties with private inputs $x_1, \cdots, x_m$ to jointly compute a function f, learn the output $f(x_1, \cdots, x_m)$ and nothing else. In the classical *stand-alone* setting, security of MPC protocols is analyzed where a single instance of a protocol runs in isolation. However, such analysis falls short of guaranteeing security in more realistic, *concurrent*, settings, where multiple instances of different protocols co-exist and are subject to coordinated attacks. To address this, Canetti formulated the *Universally Composable* (UC) framework [1] for reasoning about the security of protocols in arbitrary execution environments that dynamically interact with the analyzed protocol. The UC framework formulates, so far, the most stringent and realistic model of protocol execution, and provides a strong composability property — known as the *universal composition theorem* — that protocols shown secure in the UC framework remain secure when executed concurrently within arbitrary larger complex system.

Unfortunately, these strong properties come at a price: Many natural functionalities cannot be realized with UC security in the *plain model*, where the only setup provided is authenticated communication channels; some additional

trusted setup is necessary [2,3]. Following Canetti and Fischlin [2], Canetti, Lindell, Ostrovsky and Sahai [4] demonstrated the feasibility of UC-secure protocols realizing general functionalities, in the *Common Reference String* (CRS) Model, where a trusted entity samples a single CRS from a prescribed distribution that can be referenced to by all executions of the designed protocol. Since its conception, a long line of work have focused on designing UC secure protocols under various *trusted setups*, from CRS, to public key infrastructure, to tamper-proof hardware tokens, and many others (see for example [5,6,7,8,9,10,11]), and led to a comprehensive understanding on *what are the minimal trusted setups and computational assumptions needed for achieving UC security.*

Black-Box vs Non-Black-Box Construction: A basic distinction between cryptographic constructions is whether they make only black-box use of the underlying primitives or not. *Black-box constructions* only call the designated input/output interface of the underlying primitives, whereas *non-black-box constructions* depend on specifics of the code implementing the primitives. Typically, non-black-box constructions are more versatile for demonstrating feasibility of cryptographic tasks and minimizing underlying primitives. However, black-box constructions are more modular and tend to be more efficient. A natural theoretical direction seeks to narrow the gap between what is achieved via non-black-box and black-box constructions for important cryptographic tasks, under minimal assumptions (as done in, for example [12,13,14,15,16,17,18,19,20,21,22]), which leads to new constructions, techniques, and understanding.

For the task of achieving UC security with trusted setups, there still remain significant gaps between what is achievable via non-black-box and black-box constructions. First, generic approaches for achieving UC-security have been developed using non-black-box techniques. Lin, Pass, and Venkitasubramaniam [11,23] presented a unified framework for developing UC-secure protocols in general trusted setup models. In particular, they identified a (simple) "minimal primitive" called *UC-puzzles* that give non-black-box constructions of UC-secure protocols for general functionalities. At a high-level this primitive facilitates *concurrent simulation*, which is a necessary condition to achieve UC-security. Moreover, an important consequence of the unified framework was the weakening of trusted infrastructure and other assumptions in many models. It also significantly reduced the complexity of designing UC-secure protocols, as UC puzzles are often easy (if not trivial) to attain using trusted setups. Thus a natural question we ask in this work is,

Can we have a unified framework for developing
black-box constructions of UC-secure protocols, under general setup?

Thus far, no generic approach using black-box techniques exist, and, in fact, to the best of our knowledge, there are only a few black-box constructions [24,21,22] of UC-secure protocols for specific trusted-setup, namely the CRS and tamper-proof hardware tokens models, which fall short in the following ways:

In the CRS model, the state-of-the-art non-black-box constructions assume only the existence of semi-honest secure Oblivious Transfer (OT) protocols,

whereas black-box constructions are based on either the existence of enhanced trapdoor permutations [4], or specific assumptions, such as, Strong RSA, DDH, DLIN [25,26,27]. All these assumption imply CCA encryption and semi-honest OT. This raises the question:

Can we have black-box constructions of UC-secure protocols in the CRS model, from weaker assumptions?

Hazay and Venkitasubramaniam [21] gave partial answer to this question in the stronger "local CRS model". They gave black-box construction of UC-protocols from public-key encryption and semi-honest OT; however, every execution of their protocols needs to rely on an independently sampled *local* CRS. In contrast, the CRS model as defined originally [4] considers a *single* CRS that is shared by all concurrent executions. Clearly, having a trusted entity sampling a single CRS once and for all is a much more realistic setup than sampling a CRS for every protocol execution.

In the tamper-proof hardware token model, [4] unconditionally UC-secure protocols exist using *stateful* tokens [28,29]. When relying on much weaker (and more realistic) *stateless* tokens, computational assumptions are necessary [28]. Following a body of works [30,28,31,32,33,34], Hazay, Polychroniadou, and Venkitasubramaniam [22] showed that the minimal assumption of one-way functions suffices. However, all UC-protocols using stateless tokens require each instance of protocol execution to create a token that has specific information of the instance (namely, the session id) hardwired inside. This means parties must have the capability to create customized tokens. In this work, we consider a even weaker model of tokens, namely stateless and *instance-independent* tokens, which runs codes sampled from a universal distribution, independent of protocol instances. We believe that this model is more realistic as tokens with instance-independent codes may be obtained and distributed ahead of protocol execution, and can potentially be reused across different execution instances. We ask,

Can we have UC-secure protocols using stateless and instance-independent *tokens?*

1.1 Our Result

In this work, we present a unified framework for obtaining black-box construction of UC-secure protocols under general trusted setup, assuming semi-honest OT. At a high-level, our framework reduces the task of achieving UC-security to that of constructing a *UC-special-purpose commitment scheme* CECom with the following properties.

– CECom is *straight-line concurrently extractable w.r.t. opening*, that is, there is a straight-line extraction strategy E that can extract values from any concurrent committer C^* with the guarantee that C^* cannot successfully open to any values different from what E extracts.

[4] In the tamper-proof hardware model, parties are assumed to have tamper-proof hardware tokens that only provide input/output (i.e. black-box) access to the token holder.

– CECom is *hiding against resetting receivers*.

We observe that comparing with UC commitments, UC-special-purpose commitments are weaker in the sense that it does not guarantee simulation-extractability nor equivocation, but stronger in the sense that they are resettably hiding.

Given such a commitment scheme CECom under trusted setup $\mathcal{T}$, our unified framework shows how to construct general UC-secure protocols that make use of 4 independent instances of $\mathcal{T}$ and black-box use of CECom. We model the 4 independent instances of $\mathcal{T}$ as a single trusted-setup:

– The quadruple-$\mathcal{T}$ trusted setup $4\mathcal{T}$ simply runs 4 independent instances of $\mathcal{T}$ internally, and make them available to all parties.

In fact, for many specific trusted setups, 4 independent instances can be emulated using just a single instance. For example, in the CRS model, 4 reference strings can be concatenated into one. In the tamper proof token model, operations related to tokens are captured by an ideal functionality that allows parties to create an arbitrary number of tokens, transfer them, and execute them. One single such ideal functionality provides the same functionality as 4 of them. In these cases, our unified framework shows that to obtain black-box UC-secure protocols, it suffices to focus on constructing *UC-special-purpose commitment schemes*.

Theorem 1 (Main Theorem, Informal). *Let $\mathcal{T}$ be any trusted-setup. Assume the existence of a* UC-special-purpose commitment scheme CECom *under $\mathcal{T}$, and a constant-round semi-honest oblivious transfer protocol. Then, for every well-formed functionality $\mathcal{F}$, there is a black-box construction of a protocol π that UC-realizes $\mathcal{F}$ in the $4\mathcal{T}$-trusted setup model. Moreover, if* CECom *has r_{CEC} rounds, then π has $O(r_{\mathrm{CEC}})$ rounds.*

We remark that we rely on our setup in an "instance independent" way. In particular, in the CRS model, four references strings are sampled at the beginning and all instances rely on the same reference strings. Whereas in the token model, our result implies that we require tokens with "instance-independent" code. Technically, we follow the Joint Universal Composition (JUC) paradigm [1] and show that our protocol π when executed concurrently implement directly the multi-session extension $\hat{F}$ of the functionality $\mathcal{F}$, using a single instance of $4\mathcal{T}$.

Comparison with the LPV framework The unified framework (dubbed as the LPV framework) of [11,23] formulated the notion of *UC puzzles* and showed how to use them to obtain non-black-box constructions of UC-protocols. Roughly speaking, UC puzzle is a protocol between a sender and a receiver with two properties: i) soundness guarantees that the puzzle is hard to solve for an honest receiver, yet ii) *concurrent simulation* guarantees that the view of a concurrent sender can be simulated while obtaining all puzzle solutions. From there, the LPV framework shows how to use the UC puzzles to construct protocols that can be concurrently simulated by following the Feige-Shamir paradigm with the puzzle solutions as trapdoors.

In comparison, our unified framework requires constructing UC-special purpose commitment, which captures the capability of *concurrent extraction*. While it is known that using non-black-box techniques concurrent extraction can be achieved through concurrent simulation, as done implicitly in the LPV framework, these techniques often require the use of zero-knowledge or witness indistinguishable proofs, and are not suitable for black-box constructions. This is why in our framework, we directly require a concurrently extractable commitment scheme to start with.

Next, using the generic framework, we improve black-box construction of UC secure protocols in the CRS and tamper-proof hardware token model.

Comparison with the GUC Framework In this work, we follow the JUC framework [1] for modeling concurrent security of protocols. In particular, we show that for every functionality $\mathcal{F}$, the concurrent execution of our protocol $\pi_{\mathcal{F}}$ that implements $\mathcal{F}$ securely computes the multi-session extension of $\mathcal{F}$. This means that all instances of execution of $\pi_{\mathcal{F}}$ refer to the same trusted setup, for example, the same CRS. This model should be compared with the Global UC (GUC) framework of [7], where a trusted setup is not only available to all protocol instances, but also to the environment. This means the trusted setup can be shared between arbitrary, even potentially unknown, protocols. Therefore, protocols secure in the GUC framework provide stronger composition guarantees. However, this comes at a price, in particular, it is known that general GUC protocols in the CRS model is infeasible. In the tamper-proof hardware model, the protocols by [22] are secure in the GUC framework, but their tokens are instance-dependent.

Black-box UC Protocols in the CRS Model. In the CRS model, UC-special purpose commitment scheme is trivial to construct, simply use any public key encryption scheme. (In fact, even public key encryption with an *interactive* encryption phase suffices.) Thus, plugging into our unified framework, we immediately obtain black-box UC-protocols in the CRS model, from public key encryption and semi-honest OT.

Theorem 2. *Assuming the existence of a public-key encryption scheme and a semi-honest oblivious-transfer protocol, there exists a fully black-box construction of UC-secure protocols for general functionalities in the CRS model. Moreover, if both underlying primitives have constant rounds, then the UC-secure protocols also have constant rounds.*

Previous black-box constructions in the CRS model either relies on the existence of a trapdoor permutation [4], or specific algebraic or number theoretic assumptions, such as, DDH [27], Strong RSA [26], and DLin [26]. Note that all these assumptions imply CCA encryption, which is used in all previous constructions. In comparison, our construction only relies on a public key encryption scheme and a semi-honest OT protocol, which are not known to imply CCA encryption. Instead, in our construction, we use the public key encryption scheme to implement an *interactive* CCA encryption scheme, where the encryption phase is interactive (while the key generation and decryption procedures remain the

same). Our notion of interactive CCA encryption should be compared with that of Dodis and Fiore [35]. Our notion is stronger in the sense that the receiver in the interactive encryption phase does not need to know the secret key, whereas in the notion by [35], only receivers knowing the secret key can "receive" the encryption. In particular, their notion is insufficient for constructing UC-secure protocols in the CRS model.

On the other hand, comparing with non-black-box constructions, the best non-black-box construction assumes only the existence of semi-honest OT [11]. We thus narrow the gap in assumptions between non-black-box and black-box constructions, and leaving open the question whether public key encryption can be eliminated for black-box constructions.

Since the common reference string used in our protocols is simply the public keys of the encryption scheme, we obtain as a corollary UC secure protocols in the Uniform Reference String (URS) model assuming public key encryptions with pseudorandom public key (also referred to as dense public-key cryptosystems [36]), which also implies semi-honest OT [37].

Corollary 1. *Assuming the existence of an public-key encryption scheme with pseudorandom public keys, there exists a fully black-box construction of UC-secure protocols for general functionalities in the URS model. Moreover, if both underlying primitives have constant rounds, then the UC-secure protocols also have constant rounds.*

Using the same techniques, we believe we can also obtain black-box UC-secure protocols in the public key infrastructure model.

Black-box UC Protocols in the Tamper Proof Hardware Token Model. Extending the work of [22], we show how to construct a UC-special purpose commitment scheme using tamper-proof hardware tokens, with black-box use of a one-way function. The tokens used in our protocols are stateless and instance independent, in the sense, every token implements a stateless function that is sampled from a predefined distribution. Thus, plugging this commitment scheme into our unified framework, we immediately obtain black-box UC-protocols in the token model from semi-honest OT.

Theorem 3. *Assuming the existence of semi-honest oblivious-transfer. Then, there is black-box construction of UC-secure protocols for general functionalities in the tamper-proof hardware token model, using stateless and instance-independent hardware tokens.*

In contrast, previous works [28,34,22] either rely on *stateful* or *instance-dependent* tokens.

We believe that our framework will yield analogous improvements in other setups such as, PUF [38], global random oracle models [39], etc, and we leave it as future work to explore these instantiations.

1.2 Our Techniques

We now give an overview of our techniques. Recall that our main theorem states that for a given trusted setup $\mathcal{T}$, we can obtain black-box UC protocols in the

$4\mathcal{T}$ model from semi-honest OT and UC-special-purpose commitment schemes in the $\mathcal{T}$ model, where UC-special-purpose commitment schemes are concurrently extractable commitment schemes that are also resettably hiding. We prove this theorem in two steps. For simplicity of this overview, our discussion below will only use the concurrently extractability property of the commitments, and not the resettable hiding property. For the use of resettable hiding property, see Remark 2 at the bottom of this overview.

From CCA Commitment to Black-Box UC Protocols in Trusted Setup Models. We first show that a black-box construction of UC-secure protocols in $4\mathcal{T}$-model can be obtained from semi-honest OT and *CCA-secure commitment schemes* [40] in $4\mathcal{T}$-model. We recall here that CCA-secure commitment schemes are a stronger variant of non-malleable commitment schemes that additionally require the hiding property to hold even against adversaries that have access to the *committed-value oracle*, which can break arbitrary commitments sent by the adversary using brute force.

CCA-secure commitments were originally proposed for the purpose of constructing concurrent secure protocol in the *plain* model (without any trusted setups) that satisfy a weaker security notion called angel-based security [41] or UC with super-polynomial time helpers [40]. In these models, to circumvent the aforementioned impossibility results of UC security [2,3], the security definition is modified by allowing the adversary/simulator to have access to a *super-polynomial time* helper H or angel. Since the helper can be implemented in super-polynomial time, these models imply super-polynomial-time simulation security [42]. The security in these models can be realized in the plain model [41,43,40,19,44,20], and in particular black-box constructions of protocols satisfying UC-security with super-polynomial time helpers in the plain model can be obtained from CCA-secure commitment schemes and semi-honest OT protocol [19,20].

Our starting point is the work of [19] which builds upon techniques in [13,14,45], and show how to obtain UC-secure protocols with a super-polynomial time helper starting from semi-honest OT and CCA-secure commitments in a black-box way. We show that a direct extension of this yields an analogous result where we rely on CCA-secure commitments in $4\mathcal{T}$-model as opposed to CCA-commitments in the plain model. Moreover, the helper H is a super-polynomial machine that breaks CCACom commitments in $4\mathcal{T}$-model.

In our next step, we eliminate access to super-polynomial helpers to guarantee standard UC-security. Suppose that the CCACom is also straight-line concurrently extractable, i.e., there exists a (polynomial-time) extractor E that by simulating the $4\mathcal{T}$-setup for the concurrent committer can extract the committed values in a straight-line way, then we can simply remove the super-polynomial time helper H by simulating the trusted setup (in polynomial time), achieving UC-security. Then, we will be able to emulate H with standard UC-simulation in $4\mathcal{T}$-model.

From Concurrently Extractable Commitments to CCA-secure Commitments in Trusted Setup Models. We next show that a black-box CCA-

secure commitment scheme (with straight-line concurrent extractability as required in the above step) in $4\mathcal{T}$-model can be obtained from a straight-line concurrent extractable commitment scheme in $\mathcal{T}$-model.

Our high-level approach is to use the well-known Naor-Yung paradigm [46] that has been used to construct many CCA-secure encryption schemes. Recall that in the Naor-Yung technique, the sender encrypts a single message twice and proves "consistency" (i.e., the plaintext encrypted in both ciphertexts are equal) using a simulation-sound (non-interactive) zero-knowledge proof. Similarly, we consider a commitment scheme where, at a very high-level, the committer commits to a single message twice using a concurrently extractable commitment scheme and proves consistency. However, since our goal is to obtain black-box constructions, the committer of our protocol cannot use generic zero-knowledge proofs for proving consistency. We address this problem using the elegant technique of Choi et al. [45], developed in the context of constructing black-box non-malleable encryption from just public key encryption, and later extended to the context of constructing black-box non-malleable commitments by Wee [17]. Their techniques combine the cut-and-choose technique with Shamir's secret sharing scheme.

In more detail, we consider the following scheme as the starting point. Let CECom be a straight-line concurrently extractable commitment scheme in $\mathcal{T}$-model, and ECom be a straight-line (stand-alone) extractable commitment scheme in $\mathcal{T}$-model (ECom can be obtained from CECom trivially). Let v be the message to be committed, and $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2$ be three independent instances of $\mathcal{T}$.

Stage 1. The receiver R commits to a random subset $\Gamma \subset [10\lambda]$ of size λ using ECom and trusted setup $\mathcal{T}_0$.

Stage 2. The committer C computes a $(\lambda+1)$-out-of-10λ Shamir's secret sharing $\boldsymbol{s} = (s_1, \ldots, s_{10\lambda})$ of value v. Next, for each $j \in [10\lambda]$, C commits to s_j in parallel, using CECom and the setup $\mathcal{T}_1$. We will refer to commitments made in this stage as "commitments in the first row".

Stage 3. For each $j \in [10\lambda]$, C commits to s_j in parallel, using CECom and the setups $\mathcal{T}_2$. We will refer to commitments made in this stage as "commitments in the second row".

Stage 4 (Cut and Choose). R decommits the Stage 1 commitment to Γ. For each $j \in \Gamma$, C decommits both the j^{th} commitment in the first row and the j^{th} one in the second row, and R checks whether the two commitments are correctly decommitted to the same value s_j.

Decommitment: To decommit, simply decommit all commitments in the first row. If the shares $\boldsymbol{s} = (s_1, \ldots, s_{10\lambda})$ committed in the first row is 0.9-close to a valid codeword of v, then the committed value is v, otherwise, it is set to $\perp$.

Note that this scheme works in $3\mathcal{T}$-model since it uses three instances of $\mathcal{T}$. We remark that, similar to the scheme by Naor and Yung, this scheme satisfies the following two properties.

1. The committer is required to commit to the same value in the two rows of the commitments. Specifically, it is guaranteed by the values revealed in the

cut-and-choose stage (i.e, Stage 4) and the hiding of ECom, that the shares that are committed in the two rows are very "close" (that is, agree in most coordinates). This "closeness" ensures that there is a way of reconstructing the committed value from the shares committed in the second row. (We remark that this reconstruction works differently from that in the actual decommitment.)

2. The commitments made in the two rows are "independent" since they are generated using two independent instances of $\mathcal{T}$. When considering man-in-the-middle adversaries playing the roles of receiver and sender in the different executions, this independence will allow us to extract commitments made by the adversary from one row "correctly" while maintaining the "hiding" property of the commitments received by the adversary made in the other row.

Now, we rely on the following hybrid experiments to prove the CCA security.

H_0 The real experiment, where an adversary tries to break the hiding property of the above scheme in the "left" interaction while interacting with the committed-value oracle in the "right" interaction.

H_1 Follows the experiment as in H_0 with the following exceptions:

1. In the left interaction, the committed subset Γ is extracted from the adversary in Stage 1 using the extractability of ECom, and then $0^{|s_j|}$ is committed in the j^{th} commitment of the *second row* for every $j \notin \Gamma$
2. In the right interaction, the committed-value oracle is emulated in polynomial time as follows. All shares committed to in the *first row* are extracted relying on the extractability of the underlying CECom scheme and then the committed value is reconstructed from those extracted shares.

Notice that in this experiment, the setups $\mathcal{T}_0$ and $\mathcal{T}_1$ are simulated for extraction.

H_2 Follows experiment H_1 with the following exceptions:

1. In the left interaction, $0^{|s_j|}$ is committed in the j^{th} commitment of the *first row* for every $j \notin \Gamma$.
2. In the right interaction, the committed-value oracle is emulated in polynomial time by extracting shares from the *second row* and reconstructing the committed value.

In this experiment, the setups $\mathcal{T}_0$ and $\mathcal{T}_2$ are simulated for extraction. We notice that in this experiment, only $|\Gamma| = \lambda$ shares are set and revealed in the left execution for both rows. Hence from the perfect privacy of the underlying Shamir secret sharing scheme, the committed value in the left interaction is hidden.

Intuitively, H_0 and H_1 are indistinguishable because *i)* in H_1 the committed value oracle is emulated correctly using the shares extracted from the first rows, which defines the committed values, and thus *ii)* the only difference in the adversary's view are the values committed to in the second row on the left (which are committed using the setup $\mathcal{T}_2$), and the setup $\mathcal{T}_2$ is not simulated in these

hybrids. Additionally, at first sight, H_1 and H_2 also seem indistinguishable because *i)* in H_2 the committed-value oracle seems to be emulated correctly using values extracted from the second row thanks of the closeness, and thus *ii)* the only difference in the adversary's view is the values committed in the first row on the left (which are committed using the setup $\mathcal{T}_1$) and the setup $\mathcal{T}_1$ is not simulated in H_2.

Unfortunately, we cannot show the indistinguishability between the above hybrids since the above scheme does not guarantee *simulation soundness*. The problem is that if we simulate $\mathcal{T}_0$ on the left (as in H_1, H_2), we can no longer rely on the hiding property of ECom on the right, so we cannot show the closeness of the two rows on the right directly. This is problematic because when showing the indistinguishability between H_1 and H_2, we need to use the closeness of the two rows to argue that the committed-value oracle can be emulated correctly even from the second row.

To address this problem, we add a non-malleable commitment scheme into the above scheme. Specifically, we modify the scheme so that the second row is generated by using a commitment scheme that is both non-malleable and straight-line concurrently extractable, and additionally require the committer to commit to the decommitment of the first rows when generating the second row (i.e., we require the committer to commit to (s_j, d_j) in the second row, where d_j is the decommitment of the j^{th} commitment in the first row). With these modifications, we can prove the closeness of the two rows in H_1 as follows.

1. First, we observe that, since the decommitments of the first row are committed in the second row, the closeness of the two rows can be verified by seeing only the committed values of the second row. In particular, the closeness holds between the two rows if the second row is "consistent", meaning that a correct decommitment of the first row is committed in most coordinates.
2. Based on this observation, we show the closeness in H_1 as follows. First, we show the consistency of the second row in H_0 using the hiding property of ECom. (Recall that we do not break ECom in H_0 and can use its hiding property in H_0.) Next, when we move to H_1 from H_0, we use the non-malleability of the second row to argue that the committed values of the second row on the right does not change non-negligibly, which implies that the second row on the right remains consistent in H_1. (Here we use the ability to efficiently verify the consistency condition given the committed values of the second row). Now, since the consistency condition implies the closeness, we conclude that the closeness holds in H_1 as desired.

Given the closeness in H_1, we can show the indistinguishability between H_1 and H_2 as follows. Consider an intermediate hybrid where the left interaction is generated as in H_1 but the committed-value oracle is emulated using the second row as in H_2. Then, we first use the closeness in H_1 to argue that this intermediate hybrid is indistinguishable from H_1. Next, observing that the setup $\mathcal{T}_1$ is not simulated in this intermediate hybrid and H_2, we show that this intermediate hybrid is also indistinguishable from H_2 by using the hiding property of the second row on the left.

Finally, to complete the proof, we argue that the non-malleable commitment scheme that we use above (i.e., a commitment scheme that is both non-malleable and straight-line concurrently extractable) can be obtained without any additional assumptions. We know that constant-round black-box non-malleable commitments in the plain model can be obtained from one-way functions in a black-box way [18], which in turn can be obtained from semi-honest OT (which we assume to exist in the main theorem). Then, our idea is to combine this non-malleable commitments and CECom in $\mathcal{T}$-model in a similar manner as in the protocol above (i.e., by using secret sharing and cut-and-choose technique). Now, non-malleability follows analogous to the plain-model non-malleability of the underlying scheme and straight-line concurrent extractability follows from the properties of the latter. The resulting non-malleable commitment is proven secure in the $2\mathcal{T}$-model; thus, if this scheme is plugged into our first protocol (as the commitment used in the second row), the final protocol will be in the $4\mathcal{T}$-model.

Remark 1. We remark that several issues arise when making the preceding high-level argument formal. For example, one subtlety that we ignore is the case that the concurrently extractable commitment scheme that we use is only computationally binding (which is the case in our instantiation for the token model.) This subtlety makes the above argument complicated because the closeness of the two rows is hard to define if the shares that are committed in the rows are not uniquely determined. In our formal proof, we address this subtlety by defining the closeness property only w.r.t. the shares that are *extracted* from the rows.

Remark 2. As noted at the beginning of this overview, we actually assume the existence of concurrently extractable commitment scheme that is also resettably hiding. We use this requirement when constructing straight-line concurrently extractable non-malleable commitment schemes. Moreover, we obtain such schemes by combining a non-malleable commitment in the plain model and concurrently extractable commitment in $\mathcal{T}$-model. In the actual argument, we additionally use plain-model extractable commitments, and rely on its plain-model (i.e. rewinding based) extractability in the analysis. However relying on a rewinding analysis in the presence of trusted setups is subtle. Specifically, since the adversary might have an arbitrary unbounded-round interaction with the setups, the interaction with the setups can be rewound when the extractor rewinds the adversary. To circumvent this, we simply assume that the schemes in the setup models remain secure even when they are rewound (i.e., reset). In the two concrete setup models we consider, CRS and tamper-proof hardware model, we show that achieving resettable hiding is not hard.

2 Definitions of Commitments in Trusted-Setup Models

In this work, we consider commitment protocols that use trusted setups, meaning that the honest committer and receiver communicate with the setup $\mathcal{T}$ for

committing and decommitting, and the security of the commitment scheme relies on that $\mathcal{T}$ is never controlled by the adversary — we say such a protocol is in the trusted setup $\mathcal{T}$-model, or simply in $\mathcal{T}$-model.

For clarity, we indicate the parts related to trusted-setup models in red in the definitions we give below; removing them gives the definitions in the plain model.

2.1 Trusted Setups

We model a trusted-setup $\mathcal{T}$ as an *ideal functionality* in the UC model, which is simply given by an Interactive Turning Machine (ITM) $\mathcal{M}$. Different from UC, which models the execution of arbitrary protocol in arbitrary environment, for commitments, we only need to consider the execution of security games that define different properties, such as, hiding, binding, and CCA security. Therefore, below we describe a much simpler model of execution.

In a security game with setup $\mathcal{T}$, a set of m (honest or corrupted) parties $\{P_i\}_{i\in[m]}$, and an adversary A, the setup $\mathcal{T}$ can concurrently communicate with all entities following the rules described below:

- Whenever a party P_i, or a subroutine invoked by P_i, sends a message m to $\mathcal{T}$, $\mathcal{T}$ receives input (ID, m), where ID is the identifier of P_i or its subroutine. The identifiers of all parties and their subroutines are adaptively chosen by the adversary A at the beginning of their invocation.
- The adversary can communicate with $\mathcal{T}$ either directly according to the code of $\mathcal{T}$, or indirectly by acting as a party with identifier ID.
- All identifiers (of all parties and their subroutines and of parties acted by A) must be distinct.

2.2 Commitments in $\mathcal{T}$-Model

First we define the structure of a commitment scheme.

Definition 1 (Commitment Schemes). *A* commitment scheme *in $\mathcal{T}$-model is a pair of* PPT *ITMs $\langle C, R\rangle$ with the following properties:*

1. *The commitment scheme has two stages, a commit stage and a reveal stage, where C and R receive as common input a security parameter 1^n and C receives a private input $v \in \{0,1\}^n$ that is the string to be committed.*
2. *The commit stage results in a joint output c, called the commitment, a private output for C, d, called the decommitment string. In the commit stage, both C and R can access $\mathcal{T}$ using their respective identities $\mathsf{ID}_C \neq \mathsf{ID}_R$.*
3. *In the reveal stage, upon receiving pair (v, d), the receiver R decides to accept or reject deterministically, depending only on (c, v, d).*
   We let open *denote the function that verifies the validity of (v, d); the receiver accepts (v, d) if $\mathsf{open}(c, v, d) = 1$, and rejects otherwise,*

If C and R do not deviate from the protocol, then R should accept with probability 1 during the reveal stage.

We define the binding and hiding property of a commitment scheme in trusted setup models naturally as in the plain model.[5] (We provide their formal definition in the full version.) We say that a commitment c is *accepting* if R does not abort at the end of commit stage, and is *valid* if there exists an accepting decommitment.

Next we define the resettably hiding property of a commitment scheme. Roughly speaking, a commitment scheme in $\mathcal{T}$-setup model is resettably hiding if its hiding property holds even against any cheating receiver that can "reset" an honest committer and $\mathcal{T}$ and restart the interaction with them from an arbitrary point of the interaction.

Definition 2 (Resettably Hiding). *A commitment scheme $\langle C, R\rangle$ in $\mathcal{T}$-model is computationally (resp. statistically)* resettably hiding *if for every non-uniform* PPT *machine (resp. for every machine) R^*, the view of R^* in the following two games, Game 0 and Game 1, are computationally indistinguishable over $\lambda \in N$ (resp. statistically indistinguishable over $\lambda \in N$).*

- *Game b ($b \in \{0,1\}$): Let $C(b)$ be a committer that upon receiving (v_0, v_1) gives a commitment to v_b by using $\langle C, R\rangle$. Let F denote the forest of execution threads, initialized as empty. Then, in Game b, R^* can interact with $C(b)$ and $\mathcal{T}$ in an arbitrary number of interactions as below: R^* specifies a prefix ρ of execution in F, and starts interacting with $C(b)$ and $\mathcal{T}$ from ρ, where R^*, C and $\mathcal{T}$ use fresh randomness after ρ.*

In the rest of the paper, by default we refer to commitment schemes as ones that are statistically binding and computationally hiding, and will specify explicitly when considering commitment schemes that are computationally binding. In addition, we consider tag-based commitment schemes.

Definition 3 (Tag-based Commitment Schemes). *A commitment scheme $\langle C, R\rangle$ is* tag-based *w.r.t. $l(\lambda)$-bit identities if, in addition to the security parameter 1^λ, the committer and the receiver also receive a "tag"—a.k.a. identity—*id *of length $l(\lambda)$ as common input. In $\mathcal{T}$-model, the tag is set to the identity of the committer $\mathsf{id} = \mathsf{ID}_C$.*

2.3 Concurrent Non-malleable Commitments in $\mathcal{T}$-Model

Next we define the concurrent non-malleability of a commitment scheme. Roughly speaking, a commitment scheme is non-malleable if a man-in-the-middle adversary, who receives a commitment in the left interaction, cannot commit to a

[5] As described in Section 2.1, in the binding game with $\mathcal{T}$, R, and C^*, R can interact with the trusted setup $\mathcal{T}$ using an identity ID_R chosen by C^*, and C^* can interact with $\mathcal{T}$ directly according to $\mathcal{T}$'s code, or indirectly as any parties with identities different from ID_R. Similarly, in the hiding game with $\mathcal{T}$, C, and R^*, C can interact with the trusted setup $\mathcal{T}$ using an identity ID_C chosen by R^*, and R^* can interact with $\mathcal{T}$ directly according to $\mathcal{T}$'s code, or indirectly as any parties with identities different from ID_C.

value that is related to the values committed in the left interaction. A commitment scheme is concurrent non-malleable if it is non-malleable even when the man-in-the-middle adversary can give multiple commitments concurrently.

Formally, the concurrent non-malleability of a commitment scheme is defined as follows. Let $\langle C, R\rangle$ be a tag-based commitment scheme; recall that in $\mathcal{T}$-model, the tag of a commitment is set to the identity of the committer $\mathsf{id} = \mathsf{ID}_C$. Let M^* be a man-in-the-middle adversary and consider the following experiment. On input security parameter $\lambda \in \mathbb{N}$ and auxiliary input $z \in \{0,1\}^*$, M^* participates in one left and m right interactions simultaneously. In the left interaction, M^* interacts with the committer of $\langle C, R\rangle$ and receives a commitment to value v using identity $id \in \{0,1\}^\lambda$ of its choice, where both have access to $\mathcal{T}$. In the right interaction, M^* interacts with the receiver of $\langle C, R\rangle$ and gives commitments using identity $\widetilde{\mathsf{id}}_0, \ldots, \widetilde{\mathsf{id}}_m$ of its choice, where the commitments can be scheduled arbitrarily by M^*, and both M^* and the receiver have access to $\mathcal{T}$. Let $\widetilde{v}_1, \ldots, \widetilde{v}_m$ be the values that M^* commits to on the right. If any of the right commitments is invalid or undefined, its committed value is defined to be $\perp$. For any i, if $\mathsf{id} = \widetilde{\mathsf{id}}_i$, set $\widetilde{v}_i = \perp$. Let $\mathsf{c\text{-}mim}(\langle C, R\rangle, M^*, v, z)$ denote a random variable that describes $\widetilde{v}_1, \ldots, \widetilde{v}_m$ and the view of M^* in the above experiment.

Definition 4. *A commitment scheme $\langle C, R\rangle$ in $\mathcal{T}$-model is* ***concurrent non-malleable*** *if for any* PPT *man-in-the-middle adversary M^*, the following are computationally indistinguishable.*

- $\{\mathsf{c\text{-}mim}(\langle C, R\rangle, M^*, v_0, z)\}_{\lambda\in\mathbb{N}, v_0\in\{0,1\}^\lambda, v_1\in\{0,1\}^\lambda, z\in\{0,1\}^*}$
- $\{\mathsf{c\text{-}mim}(\langle C, R\rangle, M^*, v_1, z)\}_{\lambda\in\mathbb{N}, v_0\in\{0,1\}^\lambda, v_1\in\{0,1\}^\lambda, z\in\{0,1\}^*}$

We remark that the above definition captures "one-many" setting, where the adversary participates in one left and m right interactions simultaneously. We can easily generalize the definition so that it captures "many-many" setting, where the adversary participates in m left and m right interactions simultaneously. It is known that the "one-many" version of the definition implies the "many-many" one [47].

2.4 CCA Commitments in Trusted-Setup Models

The notion of CCA security for *statistically-binding and computationally hiding* tag-based commitment schemes was introduced in [40]. We here adapt the definition of CCA security in the plain model of [19] to trusted setup models.

Roughly speaking, a (statistically binding) commitment scheme is CCA secure if the commitment scheme retains its hiding property even if the receiver has access to a *committed-value oracle*. Let CCACom be a tag-based commitment scheme with $l(\lambda)$-bit identities; recall that in $\mathcal{T}$-model, the tag of a commitment is set to the identity of the committer $\mathsf{id} = \mathsf{ID}_C$. A committed-value oracle $\mathcal{O}_{\mathsf{CCACom}}$ of CCACom acts as follows in interaction with an adversary A, both with access to $\mathcal{T}$: It participates with A in many sessions of the commit phase of CCACom as an honest receiver, using identities chosen adaptively by A. At the end of each session, if the session is *accepting and valid*, it returns to A the

unique committed value in that session (by the statistical binding property of the commitment scheme, there exists such a unique value when the commitment is valid except with negligible probability; if not output $\perp$); otherwise, it sends $\perp$.

More precisely, let $\mathsf{IND}_b(\mathsf{CCACom}, A, \lambda, z)$, where $b \in \{0,1\}$, denote the output of the following probabilistic experiment: on common input 1^λ and auxiliary input z, $A^{\mathcal{O}_{\mathsf{CCACom}}}$ (adaptively) chooses a pair of challenge values $(v_0, v_1) \in \{0,1\}^\lambda$—the values to be committed to—and an identity id, and receives a commitment to v_b using identity id, where C and $A^{\mathcal{O}_{\mathsf{CCACom}}}$ all have access to $\mathcal{T}$. Finally, the experiment outputs the output y of $A^{\mathcal{O}_{\mathsf{CCACom}}}$; the output y is replaced by $\perp$ if the identity of the commitment that A receives is the same as the identity of any of the commitments that A sends to $\mathcal{O}_{\mathsf{CCACom}}$ (that is, any execution where the adversary queries the decommitment oracle on a commitment using the same identity as the commitment it receives, is considered invalid).

Definition 5 (CCA-security). *Let* CCACom *be a tag-based statistically binding commitment scheme in $\mathcal{T}$-model. We say that* CCACom *is* CCA-secure*, if for every* PPT *ITM A, the following ensembles are computationally indistinguishable:*

- $\{\mathsf{IND}_0(\mathsf{CCACom}, A, \lambda, z)\}_{\lambda \in N, z \in \{0,1\}^*}$
- $\{\mathsf{IND}_1(\mathsf{CCACom}, A, \lambda, z)\}_{\lambda \in N, z \in \{0,1\}^*}$

k-Robustness Roughly speaking, k-robustness states the committed-value oracle can be simulated efficiently for an attacker, without "disturbing" any k-round interaction that the attacker participates in.

Consider a man-in-the-middle adversary A that participates in an *arbitrary* left interaction with B of a *limited number of rounds*, while having access to a committed-value oracle $\mathcal{O}_{\mathsf{CCACom}}$; $A^{\mathcal{O}_{\mathsf{CCACom}}}$ has access to $\mathcal{T}$, but importantly B does not. CCACom is k-robust if the (joint) output of every k-round interaction, with an adversary having access to the oracle $\mathcal{O}_{\mathsf{CCACom}}$, can be simulated without the oracle. In other words, having access to the oracle does not help the adversary in participating in any k-round protocols that does not access $\mathcal{T}$.

Definition 6 (k-Robustness). *Let* CCACom *be a statistically binding commitment scheme in $\mathcal{T}$-model. We say that* CCACom *is* k-robust*, if for every* PPT *adversary A, there exists a* PPT *simulator S, such that, the following holds.*

Simulation: *For every* PPT *k-round ITM B that interacts only with A, the following two ensembles are computationally indistinguishable.*

- $\left\{\mathsf{output}_{B,A}[\langle B(1^\lambda, x), A^{\mathcal{O}_{\mathsf{CCACom}}, \mathcal{T}}(1^\lambda, z)\rangle]\right\}_{\lambda \in N, x, z \in \{0,1\}^{\mathrm{poly}(\lambda)}}$
- $\left\{\mathsf{output}_{B,S}[\langle B(1^\lambda, x), S(1^\lambda, z)\rangle]\right\}_{\lambda \in N, x, z \in \{0,1\}^{\mathrm{poly}(\lambda)}}$

where $\mathsf{output}_{X,Y}[\langle X(x), Y(y)\rangle]$ denote the joint output of an interaction between ITMs X and Y on private input x and y respectively, and with uniformly and independently chosen random inputs to each machine.

We say that CCACom *is* poly*-robust if it is k-robust against arbitrary polynomial $k(\lambda)$.*

2.5 Concurrent Extractability w.r.t. Commitment in $\mathcal{T}$-Model

We now define concurrent extractability w.r.t. commitment. Extraction w.r.t. commitment is defined only for statistically binding commitments and guarantees to extract from (malicious) committers the statistically defined committed values.

Definition 7 (Concurrent Extractability w.r.t. Commitment). *Let* CCACom *be a statistically binding commitment scheme in $\mathcal{T}$-model. We say that* CCACom *is* straight-line concurrently extractable w.r.t. commitment*, if there exists a universal* PPT *simulator S, such that,*

Simulation of Committed-value Oracle: *for every* PPT *adversary A, the following two ensembles are computationally indistinguishable.*

- $\left\{\langle(\mathcal{O}_{\mathsf{CCACom}}, \mathcal{T}),\ A(1^\lambda, z)\rangle\right\}_{\lambda\in N, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$
- $\left\{\langle S(1^\lambda), A(1^\lambda, z)\rangle\right\}_{\lambda\in N, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$

We say that CCACom *is* straight-line extractable w.r.t. commitment *if the above condition holds for attackers that sends only a single commitment to $\mathcal{O}_{\mathsf{CCACom}}$.*

Claim 1. *If a statistically binding commitment scheme* CCACom *in $\mathcal{T}$-model is straight-line concurrently extractable w.r.t. commitment, then it is also k-robust for any polynomial $k(\lambda)$.*

Due to space restrictions, we defer the proof of claim to the full version. At a very high level, straight-line concurrent extractability w.r.t. commitment implies poly-robustness as it essentially guarantees that $\mathcal{O}_{\mathsf{CCACom}}$ can be simulated in a straight-line, and straight-line simulation does not "disturb" the concurrent interaction with B, no matter how many rounds the interaction has.

2.6 Concurrent Extractability w.r.t. Opening in $\mathcal{T}$-Model

We now introduce the new notion of *straight-line concurrent extractability w.r.t. opening*. This notion is defined for any *computationally binding and computationally hiding* commitment scheme. Roughly speaking, it requires the commitment scheme to have an efficient extractor E satisfying the following two properties: 1) when interacting with any efficient attacker A acting as a concurrent committer, the value v that E extracts for each commitment that A sends is guaranteed to be consistent with the value v' that A opens to (i.e., $v' = \bot$ or $v = v'$), even if A receives the extracted values v's. 2) The messages that E send statistically emulate that of honest receivers, for even computationally unbounded attackers.

Definition 8. *Let* CECom *be any computationally hiding and computationally binding commitment scheme in a trusted-setup $\mathcal{T}$-model. We say that* CECom *is straight-line concurrently extractable w.r.t. opening if there exists a universal* PPT *extractor E with the following properties:*

Syntax and Statistical Emulation: *For any (potentially unbounded) adversary A, it holds that the view of A in the following real and simulated games are statistically close.*
- *In the real game, A (acting as a concurrent committer) interacts with honest receivers R in multiple sessions of* CECom*. At the end of each session, if A sends a decommitment, R replies with the decision of whether the decommitment is accepted. All parties have access to $\mathcal{T}$.*
- *In the simulated game, E emulates the honest receivers and trusted-setup for A in a straight-line.*

At the end of the commit stage of each session j, E outputs a value v_j on its special output tape.

Concurrent Extractability w.r.t. Opening: *For any PPT adversary A, consider another simulated game where A interacts with E as described above, and* at the end of the commit stage of each session j, it receives the value v_j that E outputs on its special output tape. *The probability that in any session j, A successfully decommits to a value $v'_j \neq \perp$ that is different from the value v_j that E outputs is negligible, that is, if $\mathsf{open}(c_j, v'_j, d_j) = 1$ then $v_j = v'_j$ with overwhelming probability.*

3 Robust CCACom from CECom w.r.t Opening

In this section, given a commitment scheme CECom that is straight-line concurrently extractable w.r.t. opening in $\mathcal{T}$-model, we construct a robust CCA-secure commitment scheme CCACom that uses a related trusted-setup $4\mathcal{T}$, called the quadruple-$\mathcal{T}$ trusted-setup, which runs four independent copies of $\mathcal{T}$ internally.

The $\times\mathcal{T}$ Trusted Setup: $\times\mathcal{T}$, parameterized by an integer x, is an ITM that upon invocation invokes internally x instances of $\mathcal{T}$—denoted as $\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_{x-1}$. In an experiment with $\times\mathcal{T}$, all parties and adversaries can interact with any instance, by pre-pending to every message to/from copy $\mathcal{T}_i$ with the index $i \in \{0, \cdots, x-1\}$. That is, upon receiving input $i||v$ from party P, $\times\mathcal{T}$ activates internally the copy $\mathcal{T}_i$ with input v from party P, and upon receiving output o from $\mathcal{T}_i$, returns $i||o$ to P. Additionally, each copy $\mathcal{T}_i$ can interact with the adversary as its code specifies, with all messages exchanged of form $i||\mathsf{mesg}$.

Theorem 4. *Let $\mathcal{T}$ be any trusted setup, and $4\mathcal{T}$ the corresponding quadruple-$\mathcal{T}$ trusted setup. There is a* fully black-box *construction of a* poly*-robust CCA-secure commitment scheme* CCACom *in the $4\mathcal{T}$-trusted setup model from any one-way function and any commitment scheme* CECom *in the $\mathcal{T}$-trusted setup model that is straight-line concurrently extractable w.r.t. opening and resettably hiding. Moreover, if* CECom *has r_{CEC} rounds, then* CCACom *has $O(r_{\text{CEC}})$ rounds.*

Proof. In our protocol CCACom, we use the following building blocks:

- A standard constant-round statistically-binding commitment scheme com in the plain model, which is known from one-way functions [48].

- A r_{CEC}-round commitment scheme CECom that is (straight-line) concurrently extractable w.r.t. opening in the $\mathcal{T}$-trusted setup model.
- A commitment scheme ECom that is straight-line extractable w.r.t. opening in the $\mathcal{T}$-model, which is implied by CECom in $\mathcal{T}$-model.
- A $O(r_{\text{CEC}})$-round concurrent non-malleable commitment scheme NMCom in the double-$\mathcal{T}$, $2\mathcal{T}$, trusted-setup model that is also straight-line concurrently extractable w.r.t. commitment.
  Such a commitment scheme can be constructed from any concurrent non-malleable commitment scheme in the plain model, and any commitment scheme in $\mathcal{T}$-model that is straight-line concurrently extractable w.r.t. opening and resettably hiding. (Note that the CCACom protocol itself does not directly rely on the resettable hiding property of CECom.) Due to space limitations, we provide our construction in the full version.

Next, we present our protocol CCACom formally.

Commit Phase of CCACom. On common inputs 1^λ and identities $\mathsf{ID}_C, \mathsf{ID}_R$, and private input $v \in \{0,1\}^\lambda$ to C, the committer C and receiver R interact with each other as follows:

Stage 1. R commits to a random subset $\mathit{\Gamma} \subset [10\lambda]$ of size λ using ECom and trusted setup $\mathcal{T}_0$. We will refer to $\mathcal{T}_0$ as the ECom-setup.

Stage 2 (The Com Row). C computes a $(\lambda+1)$-out-of-10λ Shamir's secret sharing $\boldsymbol{s} = (s_1, \ldots, s_{10\lambda})$ of value v. Next, for each $j \in [10\lambda]$, C commit to s_j in parallel, using com. We will refer to commitments made in this stage as "commitments in the com row".
Let ϕ_j and d_j be the commitment and decommitment for share s_j.

Stage 3 (The CECom Row). For each $j \in [10\lambda]$, C commits to (s_j, d_j) in parallel, using the protocol CECom and the setup $\mathcal{T}_1$. We will refer to commitments made in this stage as "commitments in the CECom row", and $\mathcal{T}_1$ as the CECom-setup.
Let ψ_j and e_j be the commitment and decommitment for (s_j, d_j).

Stage 4 (The NMCom Row). For each $j \in [10\lambda]$, C commits to (s_j, d_j, e_j) in parallel, using the protocol NMCom and the setups $\mathcal{T}_2$ and $\mathcal{T}_3$ to emulate the double-$\mathcal{T}$ setup $2\mathcal{T}$. We will refer to commitments made in this stage as "commitments in the NMCom row", and $\mathcal{T}_2, \mathcal{T}_3$ as the NMCom-setup.

Stage 5 (Cut and Choose). R decommits the Stage 1 commitment to $\mathit{\Gamma}$.
For each $j \in \mathit{\Gamma}$, C decommits the j^{th} commitment in the NMCom row to (s_j, d_j, e_j). R accepts if for every $j \in \mathit{\Gamma}$, the decommitment to the j^{th} NMCom commitment is accepting, and (s_j, d_j) is a valid decommitment to commitment ϕ_j in the Stage 2, and $((s_j, d_j), e_j)$ is a valid decommitment to ψ_j in Stage 3.

Decommit Phase. C sends v and the decommitments $(s_1, d_1), \cdots, (s_{10\lambda}, d_{10\lambda})$ to all com commitments $\phi_1, \cdots, \phi_{10\lambda}$. R checks all decommitments and does the following. If for any $i \in [10\lambda]$, the decommitment (s_i, d_i) is invalid w.r.t. ϕ_i,

set s_i to $\perp$. R accepts the decommitments if and only if $\mathsf{Value}(\boldsymbol{s}) = v$, where $\mathsf{Value}(\boldsymbol{s})$ for $\boldsymbol{s} = \{s_1, \cdots, s_{10\lambda}\}$ is defined as follows:

$$\mathsf{Value}(\boldsymbol{s}) = \begin{cases} & \boldsymbol{s} \text{ is 0.9-close to a valid codeword } \boldsymbol{w} = (w_1, \ldots, w_{10\lambda}), \\ v & \text{for each } j \in \Gamma, \, w_j \text{ equals the value revealed in Stage 5, and} \\ & \boldsymbol{w} \text{ decodes to } v \\ \perp & \text{otherwise} \end{cases} \tag{1}$$

Clearly, The round complexity of the above protocol is $O(r_{\text{CEC}})$. The statistical binding property of CCACom follows directly from that of com in Stage 2. Thus, it remains to show that CCACom is CCA secure and poly-robust; for the latter property, we show the stronger property of straight-line concurrently extractability w.r.t. commitment, which implies poly-robustness by Claim 1.

Proposition 1. CCACom *in the* $4\mathcal{T}$*-trusted setup model is CCA secure and straight-line concurrently extractable w.r.t. commitment.*

Due to space restrictions, we prove only CCA security below, and defer the proof of straight-line concurrently extractable w.r.t. commitment in the full version.

Proof of CCA Security For any PPT adversary A, we need to show that the outputs of the games IND_0 and IND_1 are indistinguishable (cf. Definition 5).

- $\{\mathsf{IND}_0(\mathsf{CCACom}, A, \lambda, z)\}_{\lambda \in N, z \in \{0,1\}^*}$
- $\{\mathsf{IND}_1(\mathsf{CCACom}, A, \lambda, z)\}_{\lambda \in N, z \in \{0,1\}^*}$

Towards showing the indistinguishability, for each $b \in \{0, 1\}$, we consider the following hybrid experiments $H_0^b \cdots H_7^b$; we use $\mathsf{H}_k^b(\lambda, z)$ to denote the random variable representing the view of A in the execution of $H_k^b(\lambda, z)$. Throughout the hybrids, we will keep the invariant that certain bad events do not happen except with negligible probabilities. Roughly speaking, we would like to maintain that in all hybrids, in every right session, the shares that A commits to in the com, CECom, and/or NMCom rows are "consistent", so that, we can efficiently emulate the $\mathcal{O}_{\mathsf{CCACom}}$ oracle by extracting from either the CECom rows or from the NMCom rows. Below, we first define these bad events.

Inconsistency condition: We say that a vector of shares $\tilde{\boldsymbol{s}}$ is inconsistent w.r.t. a transcript Trans of protocol CCACom, if

- Either, more than 0.1 fraction of $\tilde{\boldsymbol{s}}$ are $\perp$, that is, $|\Lambda_1 = \{j \mid \tilde{s}_j = \perp\}| > \lambda$.
- or, $\tilde{\boldsymbol{s}}$ is 0.8-close to a valid codeword $\boldsymbol{w}$, yet 0.1-far from it, that is, $|\Lambda_2 = \{j \mid \tilde{s}_j \neq w_j\}| > \lambda$, and additionally $\boldsymbol{w}$ agrees with the shares $\{s_j\}_{j \in \Gamma}$ opened to in Stage 5 in transcript Trans.

Event $\mathsf{Bad}_{\text{CEC}}$ is defined for hybrids below where the extractor $\mathcal{S}_{\mathsf{CECom}}$ of CECom is used to extract values from the CECom rows (i.e., H_1^b, H_2^b, H_3^b). Let $\{(\tilde{s}_j^k, d_j^k)\}_{j \in [10\lambda]}$

denote the values extracted by $\mathcal{S}_{\mathsf{CECom}}$ from the CECom commitment in right session k. Set

$$\tilde{s}_j^k = \widetilde{\mathsf{Extract}}(\tilde{s}_j^k, d_j^k) := \begin{cases} \tilde{s}_j^k & \text{if } (\tilde{s}_j^k, d_j^k) \text{ is a valid decommitment for } \phi_j^k \\ \bot & \text{otherwise} \end{cases} \quad (2)$$

Event $\mathrm{Bad}_{\mathrm{CEC}}$ occurs if there is an *accepting* right session k in which the shares $\{\tilde{s}_j^k\}_{10\lambda}$ extracted from the CECom row is *inconsistent* w.r.t. the transcript of this session.

EVENT $\mathrm{Bad}_{\mathrm{NM}}$ is defined for all hybrids below and concerns the values committed to in the NMCom commitments on the right. Let $\{((\hat{s}_j^k, d_j^k), e_j^k)\}_{j\in[10\lambda]}$ denote the values committed to in the NMCom row in right session k. (Since NMCom is statistically binding, the committed values are well-defined.) Set

$$\hat{s}_j^k = \widehat{\mathsf{Extract}}((\hat{s}_j^k, d_j^k), e_j^k) := \begin{cases} \hat{s}_j^k & \text{if } ((\hat{s}_j^k, d_j^k), e_j^k) \text{ is a valid decommitment for } \psi_j^k, \\ & \text{and } (\hat{s}_j^k, d_j^k) \text{ is a valid decommitment for } \phi_j^k \\ \bot & \text{otherwise} \end{cases} \quad (3)$$

where ψ_j^k and ϕ_j^k are respectively the j^{th} commitment in the CECom row and in the com row in the k^{th} right session. Event $\mathrm{Bad}_{\mathrm{NM}}$ occurs if there is an *accepting* right session k, in which the shares $\{\hat{s}_j^k\}_{10\lambda}$ extracted from the values committed in the NMCom row are *inconsistent* w.r.t. the transcript of this session.

Hybrid $H_0^b(\lambda, z)$ is the same as experiment $\mathsf{IND}_b(\mathsf{CCACom}, A, \lambda, z)$.

Hybrid $H_1^b(\lambda, z)$ is the same as $H_0^b(\lambda, z)$ except that on the right the $\mathcal{O}_{\mathsf{CCACom}}$ oracle is emulated efficiently using the extractor $\mathcal{S}_{\mathsf{CECom}}$ of CECom as follows:

1. Generate the receiver messages of CCACom honestly, except for messages in the CECom-rows.
2. Use $\mathcal{S}_{\mathsf{CECom}}$ to emulate i) the CECom receivers in the CECom-rows and in Stage 5 when A open some of the CECom commitments, and ii) the CECom-setup $\mathcal{T}_1$. By definition, at the end of each CECom-row, say in the right session k, $\mathcal{S}_{\mathsf{CECom}}$ outputs a vector of values $\{(\tilde{s}_j^k, d_j^k)\}_{j\in[10\lambda]}$ on its special output tape. Set $\tilde{s}_j^k = \widetilde{\mathsf{Extract}}(\tilde{s}_j^k, d_j^k)$, where $\widetilde{\mathsf{Extract}}$ is described in Equation (2).
3. At the end of each right session k, emulate the committed value that $\mathcal{O}_{\mathsf{CCACom}}$ returns, by returning the value $\tilde{v}^k = \widetilde{\mathsf{Value}}(\tilde{\boldsymbol{s}}^k)$ reconstructed from the shares $\tilde{\boldsymbol{s}}^k = \{\tilde{s}_j^k\}_{j\in[10\lambda]}$ where $\widetilde{\mathsf{Value}}$ is defined as

$$\widetilde{\mathsf{Value}}(\tilde{\boldsymbol{s}}) = \begin{cases} & \tilde{\boldsymbol{s}} \text{ is } 0.8\text{-close to a valid codeword } \boldsymbol{w} = (w_1, \ldots, w_{10\lambda}), \\ \tilde{v} & \forall\ j \in \Gamma,\ w_j \text{ equals the value revealed in Stage 5,} \\ & \text{and } \boldsymbol{w} \text{ decodes to } \tilde{v} \\ \bot & \text{otherwise} \end{cases} \quad (4)$$

We first show that bad events Bad_{NM} and Bad_{CEC} occur in H_1^b with negligible probability.

Lemma 1. *For every $b \in \{0,1\}$, it holds that, the probabilities that event Bad_{NM} and Bad_{CEC} occur are negligible in H_1^b.*

Proof. We first bound the probability of Bad_{CEC} occurring. Suppose for contradiction that there is an *accepting* right session k in which the shares $\{\tilde{s}_j^k = \widetilde{\mathsf{Extract}}(\tilde{s}_j^k, d_j^k)\}_{10\lambda}$ extracted from the CECom row is *inconsistent*. The inconsistency condition states that

- Either, $\tilde{\boldsymbol{s}}^k$ contains more than λ $\perp$, i.e., $|\Lambda_1 = \{j \mid \tilde{s}_j^k = \perp\}| \geq \lambda$.
- Or, $\tilde{\boldsymbol{s}}^k$ is 0.8-close to $\boldsymbol{w}^k$, yet 0.1-far from it, i.e., $|\Lambda_2 = \{j \mid \tilde{s}_j^k \neq w_j^k\}| > \lambda$, and $\boldsymbol{w}^k$ agree with the shares opened in Stage 5 of right session k.

In case 1, for this session to be accepting, it must happen that none of the locations in Λ_1 was opened in Stage 5, that is, $\Lambda_1 \cap \Gamma^k = \emptyset$, where Γ^k is the subset opened in Stage 5 of right session k; otherwise, the attacker must manage to open to a non-$\perp$ share for some $j \in \Lambda_1$, which contradicts with the concurrent extractability w.r.t. opening property of CECom. Similarly, in case 2, it must be that $\Lambda_2 \cap \Gamma^k = \emptyset$, as otherwise, the attacker must manage to open to a share $\tilde{s}_j^k = w_j^k$ for some $j \in \Lambda_2$. In both cases, A manage to form a set, Λ_1 or Λ_2, of size λ that does not intersect with Γ^k also of size λ, which violates hiding of the ECom commitment to Γ^k. (See the full version for a formal argument).

We next bound the probability of Bad_{NM} occurring in H_1^b by using the following hybrid G_1^b, G_2^b.

Hybrids G_1^b, G_2^b are identical to H_1^b except that on the right the values committed to in the NMCom commitments are extracted using the committed-value oracle $\mathcal{O}_{\mathsf{NMCom}}$ in G_1^b and using the extractor $\mathcal{S}_{\mathsf{NMCom}}$ in G_2^b. That is,

- On the right, G_1^b (resp. G_2^b) forwards all NMCom commitments to $\mathcal{O}_{\mathsf{NMCom}}$ (resp. $\mathcal{S}_{\mathsf{NMCom}}$). By definition, $\mathcal{O}_{\mathsf{NMCom}}$ (resp. $\mathcal{S}_{\mathsf{NMCom}}$) returns after every NMCom row the values committed to in this row. G_1^b (resp. G_2^b) ignores these values.

Since G_2^b is completely efficient, it follows from the same argument as above that event Bad_{NM} does not occur w.r.t. the values extracted by $\mathcal{S}_{\mathsf{NMCom}}$ except for negligible probability. Then, by the concurrent extractability w.r.t. commitment of NMCom, G_1^b and G_2^b are indistinguishable and hence Bad_{NM} does not occur w.r.t. the values returned by $\mathcal{O}_{\mathsf{NMCom}}$, except for negligible probability in G_1^b. Finally, since $\mathcal{O}_{\mathsf{NMCom}}$ emulates the receivers of NMCom perfectly for A, the views of A in H_1^b and G_1^b are identical. Thus, event Bad_{NM} (w.r.t. the values committed to in the NMCom commitments) occurs with only negligible probability in H_1^b. ∎

Now, we are ready to show the indistinguishability between H_0^b and H_1^b.

Lemma 2. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_0^b(\lambda, z)\right\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^{\text{poly}(\lambda)}} \approx \left\{\mathsf{H}_1^b(\lambda, z)\right\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^{\text{poly}(\lambda)}}$$

Proof. We show that both H_0^b and H_1^b are indistinguishable from the following simulated hybrid G^b.

Hybrid G^b is the same as $H_1^b(\lambda, z)$ except for the following:

- On the right, it emulates the right receiver messages as H_1^b does (i.e., the CECom receivers of commitments in CECom-rows and the CECom-setup are simulated using $\mathcal{S}_{\mathsf{CECom}}$, and other receiver messages are generated honestly). However, at the end of each right session k, committed value that $\mathcal{O}_{\mathsf{CCACom}}$ returns is emulated differently: It extracts the shares $\boldsymbol{s} = (s_1, \ldots, s_{10\lambda})$ committed to in the com row by brute force, and reply $\mathsf{Value}(\boldsymbol{s})$, where Value is defined in Equation (1). (That is, G^b returns to A the actual committed value in each right session.)

Claim 2. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_1^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \approx \left\{\mathsf{G}^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Proof. The only difference between these two hybrids lies in how the committed values of the right sessions are extracted: in H_1^b, they are reconstructed from the shares extracted from the CECom-rows, whereas in G^b, the actually committed value is extracted by brute-force. Thus it suffices to show that in H_1^b the values $\{\tilde{v}^k\}$ reconstructed from the shares extracted from the CECom-rows are the actual committed values $\{v^k\}$ with overwhelming probability. Since Lemma 1 gives that event $\mathrm{Bad}_{\mathrm{CEC}}$ occurs with negligible probability in H_1^b, it suffices to argue that when $\mathrm{Bad}_{\mathrm{CEC}}$ does not occur, $\tilde{v}^k = v^k$ for every right session k. Recall that if $\mathrm{Bad}_{\mathrm{CEC}}$ does not occur, in any accepting right session k the shares $\{\tilde{s}_j^k = \widetilde{\mathsf{Extract}}(\tilde{s}_j^k, d_j^k)\}_{10\lambda}$ extracted from the CECom row is consistent, so they satisfy the following condition.

1. $|\mathit{\Lambda}_1| \leq \lambda$, where $\mathit{\Lambda}_1 := \{j \mid \tilde{s}_j^k = \bot\}$, and
2. if $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ is 0.8-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10\lambda})$ such that w_j equals the value revealed in Stage 5 for each $j \in \mathit{\Gamma}^k$, then $|\mathit{\Lambda}_2| \leq \lambda$, where $\mathit{\Lambda}_2 := \{j \mid \tilde{s}_j^k \neq w_j\}$.

Let $\boldsymbol{s}^k = s^k_{j\,10\lambda}$ be the share that are committed to in the com row in the right session k. We consider two cases.

Case 1. $\boldsymbol{s}^k$ is 0.9-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10\lambda})$.
Since $|\mathit{\Lambda}_1| \leq \lambda$, $\boldsymbol{s}^k$ and $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ are 0.9-close (this follows from Equation (2)), so $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ is 0.8-close to $\boldsymbol{w}$. Hence, $\mathsf{Value}(\boldsymbol{s}^k) = \widetilde{\mathsf{Value}}(\tilde{\boldsymbol{s}}^{\boldsymbol{k}}) = \mathsf{Decode}(\boldsymbol{w})$ if w_j equals to the value revealed in Stage 5 for every $j \in \mathit{\Gamma}$, and $\mathsf{Value}(\boldsymbol{s}^k) = \widetilde{\mathsf{Value}}(\tilde{\boldsymbol{s}}^{\boldsymbol{k}}) = \bot$ otherwise.

Case 2. $\boldsymbol{s}^k$ is 0.1-far from any valid codeword.
We have $\mathsf{Value}(\boldsymbol{s}^k) = \widetilde{\mathsf{Value}}(\tilde{\boldsymbol{s}}^{\boldsymbol{k}}) = \bot$ if $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ is 0.2-far from any valid codeword, or is 0.8-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10\lambda})$ but w_j does not equal the value revealed in Stage 5 for some $j \in \mathit{\Gamma}^k$. Now, we argue that $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ cannot be 0.8-close to a valid codeword $\boldsymbol{w} = (w_1, \ldots, w_{10\lambda})$ such that w_j equals the value that is revealed in Stage 5 for every $j \in \mathit{\Gamma}^k$. Assume for contradiction that $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ is 0.8-close to such $\boldsymbol{w} = (w_1, \ldots, w_{10\lambda})$.

Then, since $|\Lambda_2| \leq \lambda$, it follows that $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ is actually 0.9-close to $\boldsymbol{w}$. However, since we have $\tilde{s}_j^k = s_j^k$ for every $j \in \{j \mid \tilde{s}_j^k \neq \perp\}$ (this follows from Equation (2)), and we have $\tilde{s}_j^k \neq \perp$ for every $j \in \{j \mid \tilde{s}_j^k = w_j\}$ (this is because $\boldsymbol{w}$ is a valid codeword), 0.9-closeness between $\tilde{\boldsymbol{s}}^{\boldsymbol{k}}$ and $\boldsymbol{w}$ implies that $\boldsymbol{s}^k$ is also 0.9-close to $\boldsymbol{w}$. This is contradiction because we assume that $\boldsymbol{s}^k$ is 0.1-far from any valid codeword.

Hence, we have $\mathsf{Value}(\boldsymbol{s}^k) = \widetilde{\mathsf{Value}}(\tilde{\boldsymbol{s}}^{\boldsymbol{k}})$, i.e., $v^k = \tilde{v}^k$, in both cases. ■

Claim 3. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_0^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \approx \left\{\mathsf{G}^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Proof. Note that the only difference between G^b and H_0^b is that in the former the CECom receivers and the CECom-setup are simulated by $\mathcal{S}_{\mathsf{CECom}}$, whereas in the latter, they are emulated honestly. It follows directly from the statistical emulation property of $\mathcal{S}_{\mathsf{CECom}}$ that these two hybrids are statistically close. (Note that the statistical emulation property of $\mathcal{S}_{\mathsf{CECom}}$ applies to even computationally unbound committers, which is the case here as hybrid H_0^b and G^b are not efficient.) ■

It follows from the above claims and a standard hybrid argument that hybrids H_0^b and H_1^b are indistinguishable. This concludes the proof of Lemma 2. ■

Hybrid $H_2^b(\lambda, z)$ is the same as $H_1^b(\lambda, z)$ except that on the left it uses the extractor $\mathcal{S}_{\mathsf{ECom}}$ of ECom to extract a subset Γ' from Stage 1 of the left session. More precisely,

- Use the extractor $\mathcal{S}_{\mathsf{ECom}}$ of ECom to emulate i) the receiver of ECom in Stage 1 of the left session and in Stage 5 when A opens this ECom commitment, as well as ii) the ECom-setup.
  By definition, at the end of Stage 1 in the left session, $\mathcal{S}_{\mathsf{ECom}}$ outputs a value Γ', interpreted as a subset, on its special output tape.
- Furthermore, in Stage 5, if A opens successfully to a set Γ and $\Gamma \neq \Gamma'$, abort and output ERR.

Lemma 3. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_1^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \overset{\mathcal{S}}{\approx} \left\{\mathsf{H}_2^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Proof. The only difference between these two hybrids lies in that in H_2^b, the receiver of ECom in the left session and the ECom-setup are emulated, and the hybrid aborts if the extracted subset Γ' disagree with the subset that A opens to. Since H_2^b is completely efficient, it follows from the extractability w.r.t. opening property of $\mathcal{S}_{\mathsf{ECom}}$ that the subset Γ that A opens to must agree with the extracted subset Γ' except for negligible probability. Moreover, conditioned on not aborting, since the extracted subset Γ' is never used otherwise, it follows from the statistical emulation property of $\mathcal{S}_{\mathsf{ECom}}$ that $\mathcal{S}_{\mathsf{ECom}}$ statistically emulates the receiver of ECom and the ECom-setup. Therefore, H_1^b and H_2^b are statistically close. ■

Then, since the two hybrids H_1^b and H_2^b are statistically close, and by Lemma 1, bad events $\mathrm{Bad}_{\mathrm{NM}}, \mathrm{Bad}_{\mathrm{CEC}}$ do not happen in H_1^b, they do not happen in H_2^b either.

Lemma 4. *For every $b \in \{0,1\}$, it holds that, the probabilities that event $\mathrm{Bad}_{\mathrm{NM}}$ and $\mathrm{Bad}_{\mathrm{CEC}}$ occur are negligible in H_2^b.*

Hybrid $H_3^b(\lambda, z)$ is the same as $H_2^b(\lambda, z)$ except that in the NMCom-row on the left, the left committer commits to 0 instead of $((s_j, d_j), e_j)$ for every $j \notin \Gamma$. Note that both H_2^b and H_3^b are completely efficient. Thus, it follows directly from the hiding property of the left NMCom commitments that H_2^b and H_3^b are indistinguishable.

Lemma 5. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_2^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \approx \left\{\mathsf{H}_3^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Moreover, we argue that the bad events $\mathrm{Bad}_{\mathrm{NM}}$ and $\mathrm{Bad}_{\mathrm{CEC}}$ do not occur in H_3^b either.

Lemma 6. *For every $b \in \{0,1\}$, it holds that, the probabilities that event $\mathrm{Bad}_{\mathrm{NM}}$ and $\mathrm{Bad}_{\mathrm{CEC}}$ occur are negligible in H_3^b.*

It follows from the hiding property of the left NMCom commitments that if event $\mathrm{Bad}_{\mathrm{CEC}}$ does not occur in H_2^b, then it does not occur in H_3^b either. Furthermore, it follows from the concurrent non-malleability property of NMCom that the values committed to in the NMCom commitments are indistinguishable in H_2^b and H_3^b. Therefore, it follows from Lemma 4 that $\mathrm{Bad}_{\mathrm{NM}}$ almost never occurs in H_3^b.

Hybrid $H_4^b(\lambda, z)$ is the same as $H_3^b(\lambda, z)$ except that on the right A interacts with the $\mathcal{O}_{\mathsf{CCACom}}$ oracle.
The only difference between H_4^b and H_3^b lies in that in the former A interacts with $\mathcal{O}_{\mathsf{CCACom}}$ on the right, whereas in the latter $\mathcal{O}_{\mathsf{CCACom}}$ is emulated using the extractor $\mathcal{S}_{\mathsf{CECom}}$ of CECom. This difference is the same as that between H_0^b and H_1^b. Furthermore, as in H_1^b, event $\mathrm{Bad}_{\mathrm{CEC}}$ does not occur in hybrid H_3^b by Lemma 6. Thus, it follows from the same proof that H_3^b and H_4^b are statistically close.

Lemma 7. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_3^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \overset{s}{\approx} \left\{\mathsf{H}_4^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Given that H_3^b and H_4^b are statistically close, it follows from Lemma 6 that event $\mathrm{Bad}_{\mathrm{NM}}$ occurs with only negligible probability in H_4^b.

Lemma 8. *For every $b \in \{0,1\}$, it holds that, the probability that event $\mathrm{Bad}_{\mathrm{NM}}$ occur is negligible in H_4^b.*

Hybrid $H_5^b(\lambda, z)$ is the same as $H_4^b(\lambda, z)$ except that on the right, it uses the committed-value oracle $\mathcal{O}_{\mathsf{NMCom}}$ of NMCom to emulate $\mathcal{O}_{\mathsf{CCACom}}$ as follows:

1. Emulate the receivers of CCACom honestly for A, except that all NMCom commitments are forwarded to $\mathcal{O}_{\mathsf{NMCom}}$, which emulates the receivers of NMCom perfectly. By definition of $\mathcal{O}_{\mathsf{NMCom}}$, at the end of each NMCom-row, say in the right session k, $\mathcal{O}_{\mathsf{NMCom}}$ returns the vector of committed values, parsed as $\{(\hat{s}_j^k, d_j^k), e_j^k\}_{j\in[10\lambda]}$. Set $\hat{s}_j^k = \widehat{\mathsf{Extract}}(\hat{s}_j^k, d_j^k)$, where $\widehat{\mathsf{Extract}}$ is described in Equation (3).
2. At the end of each right session k, emulate the committed value that $\mathcal{O}_{\mathsf{CCACom}}$ returns, by returning the value $\hat{v}^k = \widetilde{\mathsf{Value}}(\hat{\boldsymbol{s}}^k)$, where $\widetilde{\mathsf{Value}}$ is defined in Equation (4).

Lemma 9. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_4^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \overset{s}{\approx} \left\{\mathsf{H}_5^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Proof. Note that in H_5^b, the receivers of CCACom are emulated perfectly for A. Therefore, the only difference between H_5^b and H_4^b lies in how the committed values of the right sessions are extracted: in H_5^b they are reconstructed from the values committed to in the NMCom-rows, whereas in H_4^b, the actually committed values are extracted by brute-force by $\mathcal{O}_{\mathsf{CCACom}}$. Thus it suffices to show that in H_5^b the values $\{\hat{v}^k\}$ reconstructed from the values committed to in the NMCom-rows are the actual committed values $\{v^k\}$ with overwhelming probability. By Lemma 8, event $\mathrm{Bad}_{\mathrm{NM}}$ does not occur in H_4^b, except for negligible probability. Then, it follows from the same argument as in the proof of Claim 2 that when $\mathrm{Bad}_{\mathrm{NM}}$ does not occur, $\hat{v}^k = v^k$ for every right session k. ∎

Given that H_5^b and H_4^b are statistically close, it follows from Lemma 8 that event $\mathrm{Bad}_{\mathrm{NM}}$ occurs with negligible probability in H_5^b.

Lemma 10. *For every $b \in \{0,1\}$, it holds that, the probability that event $\mathrm{Bad}_{\mathrm{NM}}$ occur is negligible in H_5^b.*

Hybrid $H_6^b(\lambda, z)$ is the same as $H_5^b(\lambda, z)$ except that on the right, it uses the universal simulator $\mathcal{S}_{\mathsf{NMCom}}$ of NMCom to emulate $\mathcal{O}_{\mathsf{NMCom}}$.
It follows directly from the concurrent extractability w.r.t. commitment property of NMCom that H_6^b and H_5^b are indistinguishable.

Lemma 11. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_5^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \approx \left\{\mathsf{H}_6^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Hybrid $H_7^b(\lambda, z)$ is the same as $H_6^b(\lambda, z)$ except that on the left, the left committer i) commits to 0 instead of s_j for every $j \notin \Gamma$ in the com-row, and ii) commits to 0 instead of (s_j, d_j) for every $j \notin \Gamma$ in the CECom-row.
Note that both H_6^b and H_7^b are completely efficient. Thus, it follows directly from the hiding property of the left com commitments and CECom commitments that H_7^b and H_6^b are indistinguishable.

Lemma 12. *For every $b \in \{0,1\}$, it holds that,*

$$\left\{\mathsf{H}_6^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} \approx \left\{\mathsf{H}_7^b(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Finally, notice that in hybrid H_7^b, in the left session, the committer commits to 0 in all of the com, CECom, and NMCom rows. This means A receives no information about whether v_0 or v_1 is committed in $H_7^b(\lambda, z)$. Thus, the views of A in $\mathsf{H}_7^0(\lambda, z)$ and $\mathsf{H}_7^1(\lambda, z)$ are identically distributed.

Lemma 13. *It holds that,*

$$\left\{\mathsf{H}_7^0(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}} = \left\{\mathsf{H}_7^1(\lambda, z)\right\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^{\mathrm{poly}(\lambda)}}$$

Given the lemmas, it follows from a hybrid argument that for every b, $\{\mathsf{H}_0^b(\lambda, z)\} \approx \{\mathsf{H}_7^b(\lambda, z)\}$. Furthermore, given that $\mathsf{H}_7^0(\lambda, z)$ and $\mathsf{H}_7^1(\lambda, z)$ are identically distributed, we conclude that $\{\mathsf{H}_0^0(\lambda, z)\} \approx \{\mathsf{H}_0^1(\lambda, z)\}$ and thus the CCACom protocol is CCA-secure.

4 From CCA Commitments to UC Secure Protocols

We assume familiarity with the models of UC, Externalized UC (EUC), and Angel-based security / UC with super-polynomial helpers. See the full version for more details on these models.

4.1 The General Transformation

In this session, we show that given any commitment scheme CECom in $\mathcal{T}$-model that is straight-line concurrently extractable w.r.t. opening, we can UC-realize every functionality in the $4\mathcal{T}$-trusted-setup model. Formally,

Theorem 5 (UC-secure Protocols in $4\mathcal{T}$-trusted-setup model from CECom in $\mathcal{T}$-model). *Let $\mathcal{T}$ be any trusted-setup, and $4\mathcal{T}$ the corresponding quadruple-$\mathcal{T}$ setup. Then, for every well-formed functionality $\mathcal{F}$, there is a fully black-box construction of a protocol π that UC-realizes $\mathcal{F}$ in the $4\mathcal{T}$-trusted setup model, from the following primitives:*

- *a $O(1)$-round semi-honest secure oblivious transfer protocol, and*
- *a commitment scheme CECom in $\mathcal{T}$-model that is straight-line concurrently extractable w.r.t. opening and resettably hiding.*

Moreover, if CECom has r_{CEC} rounds, π has $O(r_{\mathrm{CEC}})$ rounds.

We achieve the theorem in three steps.

Step 1: Starting from a r_{CEC}-round commitment scheme CECom that is straight-line concurrently extractable w.r.t. opening and resettably hiding in the $\mathcal{T}$-model, by Theorem 4, there is fully black-box construction of a CCA-secure commitment scheme CCACom in $4\mathcal{T}$-model that is also straight-line concurrently extractable w.r.t. commitment, and the scheme has $r_{\mathrm{CCA}} = O(r_{\mathrm{CEC}})$ rounds. Recall that by Claim 1, such a scheme is also poly-robust.

Shared Functionality $\bar{\mathcal{H}}_{\mathsf{CCACom}}$

Initialize a session with $\mathcal{O}_{\mathsf{CCACom}}$: Upon receiving an input $(\texttt{Init}, P_i, sid, k)$ from party P_i in the protocol instance sid, if there is a previously recorded session (P_i, sid, k), ignore this message; otherwise, initialize a session of CCACom with $\mathcal{O}_{\mathsf{CCACom}}$ using identity (P_i, sid), and record session (P_i, sid, k).

Access $\mathcal{O}_{\mathsf{CCACom}}$: Upon receiving an input $(\texttt{Mesg}, P_i, sid, k, m)$ from party P_i in the protocol instance sid, if there is no previously recorded session (P_i, sid, k), ignore the message; otherwise, forward m to $\mathcal{O}_{\mathsf{CCACom}}$ in the k^{th} session that uses identity (P_i, sid), obtain a reply m', and return $(\texttt{Mesg}, P_i, sid, k, m')$ to P_i.

Accessing $4\mathcal{T}$: Upon receiving an input $(\texttt{setup}, P_i, sid, m)$ from party P_i in the protocol instance sid, forward $((P_i, sid), m)$ to $\mathcal{T}$, obtain a reply m', and return $(\texttt{setup}, P_i, sid, m')$ to P_i.

Fig. 1. The ideal shared functionality $\bar{\mathcal{H}}_{\mathsf{CCACom}}$

Step 2: Given a poly-robust CCA-secure commitment scheme CCACom, it follows from the work of Lin and Pass [19] that every well-formed functionality $\mathcal{F}$ can be EUC-realized w.r.t. the shared functionality $\bar{\mathcal{H}}_{\mathsf{CCACom}}$ defined by CCACom. Roughly speaking, $\bar{\mathcal{H}}_{\mathsf{CCACom}}$ runs the committed-value oracle of CCACom for every party with the restriction that when invoked by a party with identity $\mathsf{ID} = (P_i, sid)$ (consisting of party ID P_i and session ID sid), it only breaks CCACom commitments with exactly the same identity ID. Since we here consider robust CCA-secure CCACom in $4\mathcal{T}$-model, all parties also have access to $4\mathcal{T}$. Thus, we let $\bar{\mathcal{H}}_{\mathsf{CCACom}}$ run also the setup $4\mathcal{T}$. A formal description of the functionality is in Figure 1.
Therefore, honest parties interact with $\bar{\mathcal{H}}_{\mathsf{CCACom}}$ to access $4\mathcal{T}$, while corrupted parties, adversaries A/S, and environment Z can interact with $\bar{\mathcal{H}}_{\mathsf{CCACom}}$ to access both $4\mathcal{T}$ and the committed-value oracles $\mathcal{O}_{\mathsf{CCACom}}$. We note that since the work of [19] considers CCA secure commitment schemes in the plain model, their helper functionalities only run the committed-value oracle, and the honest parties never access the helper functionality. Their construction and security proof extends directly to our case where the honest parties access the helper functionality for $4\mathcal{T}$ only, but not $\mathcal{O}_{\mathsf{CCACom}}$.

Theorem 6 ([19]). *Assume the existence of a r_{CCA}-round* poly*-robust CCA-secure commitment scheme CCACom in the $4\mathcal{T}$-trusted-setup model, and a constant-round semi-honest secure oblivious transfer protocol. Then, for every well-formed functionality $\mathcal{F}$, there is a fully black-box construction of a $O(r_{\text{CEC}})$-round protocol π that $\bar{\mathcal{H}}_{\mathsf{CCACom}}$-EUC-realizes $\mathcal{F}$.*

Step 3: Finally, we move from EUC-security w.r.t. shared functionality $\bar{\mathcal{H}}_{\mathsf{CCACom}}$ back to UC-security w.r.t.$4\mathcal{T}$-trusted-setup, by crucially relying on the fact that CCACom is straight-line concurrently extractable w.r.t. commitment.

Theorem 7. *Let* CCACom *be any commitment scheme that is CCA-secure and straight-line concurrently extractable w.r.t. commitment in the* $4\mathcal{T}$*-trusted-setup model. For every well-formed functionality* $\mathcal{F}$*, if protocol* π $\bar{\mathcal{H}}_{\mathsf{CCACom}}$*-EUC-realizes* $\mathcal{F}$*, then* π *UC-realizes* $\mathcal{F}$ *in the* $4\mathcal{T}$*-trusted-setup model.*

To show that π UC-realizes $\mathcal{F}$ in the $4\mathcal{T}$-trusted-setup model, we need to show that its multi-session extension $\hat{\pi}$ UC-realizes the multi-session extension $\hat{\mathcal{F}}$ of $\mathcal{F}$ in the $4\mathcal{T}$-hybrid model. This follows from the following two simple observations.

First, combining the universal composition theorem of EUC with Theorem 6 gives that $\hat{\pi}$ $\bar{\mathcal{H}}_{\mathsf{CCACom}}$-EUC-realizes $\hat{F}$. That is, for any PPT adversary $\mathcal{A}$, there exists a PPT simulator S, such that, for every PPT environment Z, it holds that

$$\text{EXEC}^{\bar{\mathcal{H}}}_{\hat{\pi},\mathcal{A},Z} \approx \text{EXEC}^{\bar{\mathcal{H}}}_{\mathcal{I}_{\hat{\mathcal{F}}},S,Z} \ ,$$

where $\bar{\mathcal{H}}$ is a short hand for $\bar{\mathcal{H}}_{\mathsf{CCACom}}$.

By definition of EUC, the above indistinguishability holds for arbitrary A and Z that may or may not access the shared functionality $\bar{\mathcal{H}}$. Consider the special case where $\mathcal{A}$ never accesses $\mathcal{O}_{\mathsf{CCACom}}$ in $\bar{\mathcal{H}}$ (but may access $4\mathcal{T}$ in $\bar{\mathcal{H}}$), and Z never accesses $\bar{\mathcal{H}}$ at all. In this case, in the real execution, honest parties of $\hat{\pi}$ and $\mathcal{A}$ may access $4\mathcal{T}$ in $\bar{\mathcal{H}}$, and no party accesses $\mathcal{O}_{\mathsf{CCACom}}$ in $\bar{\mathcal{H}}$. Note that this is simply an execution $\text{EXEC}_{\hat{\pi},\mathcal{A},Z}(\lambda, z)$ of $\hat{\pi}$ with adversary $\mathcal{A}$ and environment Z in the $4\mathcal{T}$-hybrid-model. On the other hand, in the ideal execution, only the simulator S interacts with $\bar{\mathcal{H}}$ and no other party interacts with $\bar{\mathcal{H}}$ at all.

Next, to show that π UC-realizes $\mathcal{F}$ in $4\mathcal{T}$ trusted-setup model, we need to show that $\hat{\pi}$ UC-emulates the ideal protocol $\mathcal{I}_{\hat{\mathcal{F}}}$ of $\hat{F}$ in the $4\mathcal{T}$-hybrid model. That is, for any PPT adversary $\mathcal{A}$, there exists a PPT simulator S', such that, for any PPT environment Z, it holds that

$$\text{EXEC}_{\hat{\pi},\mathcal{A},Z} \approx \text{EXEC}_{\mathcal{I}_{\hat{\mathcal{F}}},S',Z} \ .$$

As discussed above, for any $\mathcal{A}$, Z, λ and z, the experiments $\text{EXEC}_{\hat{\pi},\mathcal{A},Z}(\lambda, z)$ and $\text{EXEC}^{\bar{\mathcal{H}}}_{\hat{\pi},\mathcal{A},Z}(\lambda, z)$ are identically distributed. We now use the simulator S for $\mathcal{A}$ in the EUC model to construct the a simulator S' for $\mathcal{A}$ in the UC model satisfying that

$$\text{EXEC}_{\mathcal{I}_{\hat{\mathcal{F}}},S',Z} \approx \text{EXEC}^{\bar{\mathcal{H}}}_{\mathcal{I}_{\hat{\mathcal{F}}},S,Z}$$

The only difference between these two ideal executions is that in the former Z interacts with S' and in the latter Z interacts with S who interacts with $\bar{\mathcal{H}}$ (no other party accesses $\bar{\mathcal{H}}$). Construct S' as follows: It internally runs S and emulates (the committed-value oracle of CCACom and the setup $4\mathcal{T}$ in) $\bar{\mathcal{H}}$ for S, using the simulator $\mathcal{S}_{\mathsf{CCACom}}$ of CCACom. It follows directly from the concurrent extractability w.r.t. commitment property of CCACom that the simulation is indistinguishable and so are the above two experiments. It then follows from a hybrid argument that S' is a valid simulator for $\mathcal{A}$ in the UC model. Therefore, we conclude that π UC-realizes $\mathcal{F}$ in the $4\mathcal{T}$-trusted setup model.

Combining the above three steps gives a protocol π that UC-realizes an arbitrary functionality $\mathcal{F}$ in the $4\mathcal{T}$-trusted setup model. In addition, it is easy to see that the protocol has $O(r_{\text{CCA}}) = O(r_{\text{CEC}})$ rounds. This concludes Theorem 5.

4.2 Instantiation of CECom in the CRS model

In this section we present our CECom in the $\mathcal{F}_{\mathsf{CRS}}$-hybrid model.
Protocol $\mathsf{CECom}_{\mathsf{CRS}}$ We will require a *perfectly-correct* semantically-secure public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ for this construction.

Common Reference String. The common reference string is set to pk where $(\mathsf{pk}, \mathsf{sk})$ is sampled according to $\mathsf{Gen}(1^\kappa)$.
Input. C and R have as common input 1^λ and identities sid, and C has private input $v \in \{0,1\}^\lambda$.
Commit Phase of $\mathsf{CECom}_{\mathsf{CRS}}$. Sen queries $\mathcal{F}_{\mathsf{CRS}}$ to obtain the $\mathsf{CRS} = \mathsf{pk}$. Then it samples randomness r and sends $c = \mathsf{Enc}_{\mathsf{pk}}(v; r)$ to the receiver.
Decommitment Phase. The sender simply reveals v, r.

From semantic security and correctness of the underlying encryption scheme, $\mathsf{CECom}_{\mathsf{CRS}}$ is statistically binding, computationally hiding, straight-line concurrently extractable w.r.t. commitment, and resettably-hiding in the $\mathcal{F}_{\mathsf{CRS}}$-model. Therefore we have the following lemma:

Lemma 14. *Assume the existence of public-key encryption scheme. Then, there exists a computationally-hiding statistically-binding commitment scheme that is (1) Straight-line concurrently extractable w.r.t commitment, and (2) Resettably-hiding in the Common Reference String Model.*

Instantiation of CECom in the Uniform Reference String model An immediate corollary to our instantiation in the CRS model is an instantiation in the uniform reference string (URS) model. Recall that in the URS model, the reference string is sampled as uniformly random. We can rely on the same construction as in the CRS model if the we rely on a dense public-key encryption scheme where additionally the distribution of the sampled public-keys are pseudorandom. More precisely we have the following corollary

Corollary 1 *Assume the existence of a dense public-key encryption scheme. Then, there exists a computationally-hiding statistically-binding commitment scheme that is (1) Straight-line concurrently extractable w.r.t commitment, and (2) Resettably-hiding in the Uniform Reference String Model.*

4.3 Instantiation of CECom in the Tamper Proof Hardware Model

We assume familiarity of the global tamper proof model of [22], where operations related to tokens are captured by the ideal functionality $\mathcal{F}_{\mathsf{gwrap}}$.

A simple extractable commitment based on tokens can be achieved as follows: The receiver chooses a function F from a pseudorandom function family that

maps $\{0,1\}^m$ to $\{0,1\}^n$ bits where $m >> n$, and incorporates it into a token that it sends to the sender. Next, the sender commits to its input b by first sampling a random string $u \in \{0,1\}^m$ and querying the PRF token on u to receive the value v. It sends as its commitment the string $\mathsf{com}_b = (\mathsf{Ext}(u;r) \oplus b, r, v)$ where $\mathsf{Ext}(\cdot,\cdot)$ is a strong randomness extractor. Hiding follows from the fact that the PRF is highly compressing, while binding follows from the pseudorandomness of the underlying PRF. Extraction on the other hand can be achieved by allowing the simulator to observe the queries made by the sender to the token and waiting for a query to give the answer v. First, we remark that this commitment only achieves commitment w.r.t opening as the extraction procedure does not know when the commitment is correct. This is however not an issue as our general framework can rely on CECom that has straight-line extractability w.r.t opening. A larger issue however is to handle resettability of tokens. A resetting receiver can leak information by creating a stateful token and rewinding the committer. We tackle this problem by observing that resettable hiding of our protocol can be solved by using a commitment scheme with "reusable" tokens. Such a scheme was presented in [22] and we here rely on a milder variant of this protocol.

Protocol $\mathsf{CECom}_{\mathsf{TK}}$

Input. C and R receive as common inputs 1^λ and identity $(\mathsf{sid}, \mathsf{ssid})$, and individual inputs pid_C and pid_R respectively. C also receives as private input $v \in \{0,1\}^\lambda$.

Commit Phase of $\mathsf{CECom}_{\mathsf{TK}}$.

Round 1. The Receiver creates the following tokens and sends it to the sender.
- For every $l \in [2\kappa]$, Receiver chooses a random PRF keys $\gamma_{b,l}$ ($l \in [\kappa], b \in \{0,1\}$) from a PRF family $\mathcal{F}$ from 5κ bits to κ. Then, for every (b,l), the Receiver creates the tokens $\mathsf{TK}^{\mathsf{PRF},l}$ by sending the message $\{\mathsf{create}, \mathsf{sid}, \mathsf{ssid}, \mathrm{Rec}, \mathrm{Sen}, \mathsf{tid}_{b,l}, M_{b,l}\}$, that on input x, outputs $\mathsf{PRF}_{\gamma_{b,l}}(x)$, where $M_{b,l}$ is the functionality.

Round 2. Sen $\to$ Rec: Sen picks κ random bits $h_1, \ldots, h_k$. For every $i \in [\kappa]$, run TK_{i,h_i} on input u and check if all token output the same value v. It they don't output the receiver halts. Otherwise, it commits by transmitting $(\mathrm{Ext}(u) \oplus m, v)$ to the sender, where $\mathrm{Ext} : \{0,1\}^{5\kappa} \times \{0,1\}^d \to \{0,1\}$ is a $(2\kappa + 1, 2^{-\kappa})$ randomness extractor and the seed has length d (for simpler exposition we drop the seed in the expression above).

Decommitment Phase: The sender simply reveals u and m.

The following properties follow directly from the pseudorandomness of the underlying PRF and the fact that the function is highly compressing. We provide formal proofs in the full version.

Proposition 1 $\mathsf{CECom}_{\mathsf{TK}} = \langle C, R \rangle$ *presented above is a computationally binding commitment scheme in the* $\mathcal{F}_{\mathsf{gwrap}}$*-model.*

Proposition 2 $\mathsf{CECom}_{\mathsf{TK}}$ *is statistically hiding commitment scheme in the* $\mathcal{F}_{\mathsf{gwrap}}$*-model.*

We can further show a stronger hiding property.

Proposition 3 $\mathsf{CECom}_{\mathsf{TK}}$ *is straight-line concurrently extractable w.r.t. opening in* $\mathcal{F}_{\mathsf{gwrap}}$*-model.*

Proposition 4 $\mathsf{CECom}_{\mathsf{TK}}$ *is resettably-hiding in* $\mathcal{F}_{\mathsf{gwrap}}$*-model.*

Acknowledgments. We sincerely thank the anonymous reviewers for their incredibly helpful and insightful comments and suggestions.

Huijia Lin was supported by NSF grants CNS-1528178, CNS-1514526, CNS-1652849 (CAREER), a Hellman Fellowship, the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under a subcontract No. 2017-002 through Galois. Muthuramakrishnan Venkitasubramaniam is supported by a Google Faculty Research Grant and NSF Awards CNS-1526377 and CNS-1618884. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, the U.S. Government or Google.

References

1. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS. (2001) 136–145
2. Canetti, R., Fischlin, M.: Universally composable commitments. In: CRYPTO. (2001) 19–40
3. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: EUROCRYPT. (2003) 68–86
4. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: STOC. (2002)
5. Groth, J., Ostrovsky, R.: Cryptography in the multi-string model. In: CRYPTO. (2007) 323–341
6. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: FOCS. (2004) 186–195
7. Canetti, R., Dodis, Y., Pass, R., Walfish, S.: Universally composable security with global setup. IACR Cryptology ePrint Archive **2006** (2006) 432
8. Kalai, Y.T., Lindell, Y., Prabhakaran, M.: Concurrent composition of secure protocols in the timing model. J. Cryptology **20**(4) (2007) 431–492
9. Katz, J.: Universally composable multi-party computation using tamper-proof hardware. In: EUROCRYPT. (2007) 115–128
10. Canetti, R., Pass, R., Shelat, A.: Cryptography from sunspots: How to use an imperfect reference string. In: FOCS. (2007) 249–259
11. Lin, H., Pass, R., Venkitasubramaniam, M.: A unified framework for concurrent security: universal composability from stand-alone non-malleability. In: STOC. (2009) 179–188
12. Kilian, J.: Founding cryptography on oblivious transfer. In: 20th ACM STOC, ACM Press (May 1988) 20–31
13. Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions for secure computation. In: STOC. (2006) 99–108

14. Haitner, I.: Semi-honest to malicious oblivious transfer - the black-box way. In: TCC. (2008) 412–426
15. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In Wagner, D., ed.: CRYPTO 2008. Volume 5157 of LNCS., Springer, Heidelberg (August 2008) 572–591
16. Pass, R., Wee, H.: Black-box constructions of two-party protocols from one-way functions. In: TCC. (2009) 403–418
17. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: 51st FOCS, IEEE Computer Society Press (October 2010) 531–540
18. Goyal, V., Lee, C., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: FOCS. (2012) 51–60
19. Lin, H., Pass, R.: Black-box constructions of composable protocols without set-up. In: CRYPTO. (2012) 461–478
20. Kiyoshima, S.: Round-efficient black-box construction of composable multi-party computation. In Garay, J.A., Gennaro, R., eds.: CRYPTO 2014, Part II. Volume 8617 of LNCS., Springer, Heidelberg (August 2014) 351–368
21. Hazay, C., Venkitasubramaniam, M.: On black-box complexity of universally composable security in the CRS model. In: ASIACRYPT. (2015) 183–209
22. Hazay, C., Polychroniadou, A., Venkitasubramaniam, M.: Composable security in the tamper-proof hardware model under minimal complexity. In: Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I. (2016) 367–399
23. Pass, R., Lin, H., Venkitasubramaniam, M.: A unified framework for UC from only OT. In Wang, X., Sako, K., eds.: ASIACRYPT 2012. Volume 7658 of LNCS., Springer, Heidelberg (December 2012) 699–717
24. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: CRYPTO. (2000) 432–450
25. Damgård, I., Groth, J.: Non-interactive and reusable non-malleable commitment schemes. In: STOC. (2003) 426–437
26. Choi, S.G., Katz, J., Wee, H., Zhou, H.: Efficient, adaptively secure, and composable oblivious transfer with a single, global CRS. In: PKC. (2013) 73–88
27. Lindell, Y.: Highly-efficient universally-composable commitments based on the DDH assumption. In: EUROCRYPT. (2011) 446–466
28. Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding cryptography on tamper-proof hardware tokens. In: Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings. (2010) 308–326
29. Moran, T., Segev, G.: David and goliath commitments: UC computation for asymmetric parties using tamper-proof hardware. In: EUROCRYPT. (2008) 527–544
30. Chandran, N., Goyal, V., Sahai, A.: New constructions for UC secure computation using tamper-proof hardware. In: EUROCRYPT. (2008) 545–562
31. Döttling, N., Kraschewski, D., Müller-Quade, J.: Unconditional and composable security using a single stateful tamper-proof hardware token. In: TCC. (2011) 164–181
32. Döttling, N., Mie, T., Müller-Quade, J., Nilges, T.: Implementing resettable uc-functionalities with untrusted tamper-proof hardware-tokens. In: TCC. (2013) 642–661
33. Döttling, N., Kraschewski, D., Müller-Quade, J., Nilges, T.: General statistically secure computation with bounded-resettable hardware tokens. In: TCC. (2015) 319–344

34. Choi, S.G., Katz, J., Schröder, D., Yerukhimovich, A., Zhou, H.: (efficient) universally composable oblivious transfer using a minimal number of stateless tokens. In: TCC. (2014) 638–662
35. Dodis, Y., Fiore, D.: Interactive encryption and message authentication. In Abdalla, M., Prisco, R.D., eds.: SCN 14. Volume 8642 of LNCS., Springer, Heidelberg (September 2014) 494–513
36. De Santis, A., Persiano, G.: Zero-knowledge proofs of knowledge without interaction (extended abstract). In: 33rd FOCS, IEEE Computer Society Press (October 1992) 427–436
37. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: 41st FOCS, IEEE Computer Society Press (November 2000) 325–335
38. van Dijk, M., Rührmair, U.: Physical unclonable functions in cryptographic protocols: Security proofs and impossibility results. IACR Cryptology ePrint Archive **2012** (2012) 228
39. Canetti, R., Jain, A., Scafuro, A.: Practical UC security with a global random oracle. In: CCS. (2014) 597–608
40. Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: FOCS. (2010) 541–550
41. Prabhakaran, M., Sahai, A.: New notions of security: achieving universal composability without trusted setup. In: STOC. (2004) 242–251
42. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: EUROCRYPT. (2003) 160–176
43. Malkin, T., Moriarty, R., Yakovenko, N.: Generalized environmental security from number theoretic assumptions. In: TCC. (2006) 343–359
44. Goyal, V., Lin, H., Pandey, O., Pass, R., Sahai, A.: Round-efficient concurrently composable secure computation via a robust extraction lemma. In Dodis, Y., Nielsen, J.B., eds.: TCC 2015, Part I. Volume 9014 of LNCS., Springer, Heidelberg (March 2015) 260–289
45. 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. Volume 4948 of LNCS., Springer, Heidelberg (March 2008) 427–444
46. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC, ACM Press (May 1990) 427–437
47. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent non-malleable commitments from any one-way function. In: TCC. (2008) 571–588
48. Naor, M.: Bit commitment using pseudorandomness. Journal of Cryptology **4** (1991) 151–158