Adaptive and Concurrent Secure Computation from New Adaptive, Non-malleable Commitments

Dana Dachman-Soled[1,⋆], Tal Malkin[2], Mariana Raykova[3,⋆⋆], and Muthuramakrishnan Venkitasubramaniam[4]

[1] University of Maryland, College Park, MD 20742, USA
danadach@ece.umd.edu

[2] Columbia University, New York, NY 10027, USA
tal@cs.columbia.edu

[3] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA, and
SRI, Menlo Park, CA 94025, USA
mariana@cs.columbia.edu

[4] University of Rochester, Rochester, NY 14627, USA
muthuv@cs.rochester.edu

Abstract. We present a unified approach for obtaining general secure computation that achieves adaptive-Universally Composable (UC)-security. Using our approach we essentially obtain all previous results on adaptive concurrent secure computation, both in relaxed models (e.g., quasi-polynomial time simulation), as well as trusted setup models (e.g., the CRS model, the imperfect CRS model). This provides conceptual simplicity and insight into what is required for adaptive and concurrent security, as well as yielding improvements to set-up assumptions and/or computational assumptions in known models. Additionally, we provide the first constructions of concurrent secure computation protocols that are adaptively secure in the timing model, and the non-uniform simulation model. As a corollary we also obtain the first adaptively secure multiparty computation protocol in the plain model that is secure under bounded-concurrency.

Conceptually, our approach can be viewed as an adaptive analogue to the recent work of Lin, Pass and Venkitasubramaniam [STOC '09], who considered only non-adaptive adversaries. Their main insight was that the non-malleability requirement could be decoupled from the simulation requirement to achieve UC-security. A main conceptual contribution of this work is, quite surprisingly, that it is still the case even when considering adaptive security.

A key element in our construction is a commitment scheme that satisfies a strong definition of non-malleability. Our new primitive of *concurrent equivocal non-malleable commitments*, intuitively, guarantees that even when a man-in-the-middle adversary observes concurrent equivocal commitments and decommitments, the binding property of the commitments continues to hold for commitments made by the adversary. This definition is stronger than previous ones, and may be of independent interest. Previous constructions that satisfy our definition have been constructed in setup models, but either require existence of stronger encryption schemes such as CCA-secure encryption or require independent "trapdoors" provided by the setup for every pair of parties to ensure non-malleability. A main technical contribution of this work is to provide a construction that eliminates these requirements and requires *only* a single trapdoor.

[⋆] Most of this work was done while at Columbia University and Microsoft Research.

[⋆⋆] Supported by NSF Grant No.1017660.

K. Sako and P. Sarkar (Eds.) ASIACRYPT 2013 Part I, LNCS 8269, pp. 316–336, 2013.

1 Introduction

The notion of *secure multi-party computation* allows mutually distrustful parties to securely compute a function on their inputs, such that only the (correct) output is obtained, and no other information is leaked, even if the adversary controls an arbitrary subset of parties. This security is formalized via the real/ideal simulation paradigm, requiring that whatever the adversary can do in a real execution of the protocol, can be simulated by an adversary ("simulator") working in the ideal model, where the parties submit their inputs to a trusted party who then computes and hands back the output. Properly formalizing this intuitive definition and providing protocols to realize it requires care, and has been the subject of a long line of research starting in the 1980s.

In what is recognized as one of the major breakthroughs in cryptography, strong feasibility results were provided, essentially showing that *any function that can be efficiently computed, can be efficiently computed securely,* assuming the existence of enhanced trapdoor permutations (eTDP) [46,27]. However, these results were originally investigated in the *stand-alone setting*, where a single instance of the protocol is run in isolation. A stronger notion is that of *concurrent security*, which guarantees security even when many different protocol executions are carried out concurrently. In this work, we focus on the strongest (and most widely used) notion of concurrent security, namely universally-composable (UC) security [6]. This notion guarantees security even when an unbounded number of different protocol executions are run concurrently in an arbitrary interleaving schedule and is critical for maintaining security in an uncontrolled environment that allows concurrent executions (e.g., the Internet). Moreover, this notion also facilitates modular design and analysis of protocols, by allowing the design and security analysis of small protocol components, which may then be composed to obtain a secure protocol for a complex functionality.

Unfortunately, achieving these strong notions of concurrent security is far more challenging than achieving stand-alone security, and we do not have general feasibility results for concurrently secure computation of every function. In fact, there are lower bounds showing that concurrent security (which is implied by UC security) cannot be achieved for general functions, unless trusted setup is assumed [8,9,35]. Previous works overcome this barrier either by using some trusted setup infrastructure [8,11,2,7,30,12], or by relaxing the definition of security [39,45,3,10,25] (we will see examples below).

Another aspect of defining secure computation, is the power given to the adversary. A *static* (or non-adaptive) adversary is one who has to decide which parties to corrupt at the outset, before the execution of the protocol begins. A stronger notion is one that allows for an *adaptive* adversary, who may corrupt parties at any time, based on its current view of the protocol. It turns out that achieving security in the adaptive setting is much more challenging than in the static one. The intuitive reason for this is that the simulator needs to simulate messages from uncorrupted parties, but may later need to explain the messages (i.e. produce the randomness used to generate those messages) when that party is corrupted. Moreover, the simulator must simulate messages from uncorrupted parties *without knowing their inputs*, but when corrupted, must explain the messages according to the actual input that the party holds. On the other hand, in the real protocol execution, messages must information-theoretically determine the actual inputs of the party, both for correctness as well as to ensure that an adversary is committed to its inputs and cannot cheat. We note that although the setting of adaptive corruptions *with erasures* has been considered in the literature, in our work we assume adaptive corruptions *without erasures*. Here we assume that honest parties cannot reliably erase randomness used to generate messages of the protocol and thus when corrupted, the adversary learns the randomness used by that party to generate previous protocol messages. Clearly, this is the more general and challenging setting. Canetti, Lindell, Ostrovsky and Sahai [11] provided the first constructions of UC-secure protocols with static and adaptive security in the *common reference string* model (CRS)[1]. Subsequently, several results were obtained for both the static and adaptive case in other trusted-setup models and relaxed-security models. The techniques for achieving security against adaptive adversaries are generally quite different than the techniques needed to achieve security against static adversaries, and many results for concurrent secure computation do not readily extend to the adaptive setting. In fact, several of the previous results allowing general concurrent secure computation (e.g., using a trusted setup) were only proved for the static case [33,34,42,40,22,30], and extending them to the adaptive setting has remained an open problem.

In this paper we focus on the strongest notions of security, and study their fundamental power and limitations. The main question we ask is:

Under which circumstances is adaptive concurrent security generally feasible?

In particular, we refine this question to ask:

What is the minimum setup required to achieve adaptive concurrent security?

We address these questions on both a conceptual and technical level. In particular, we unify and generalize essentially all previous results in the generic adaptive concurrent setting, as well as providing completely new results (constructions with weaker trusted setup requirements, weaker computational assumptions, or in relaxed models of security), conceptual simplicity, and insight into what is required for adaptive and concurrent secure computation. Our main technical tool is a new primitive of equivocal non-malleable commitment. We describe our results in more detail below.

1.1 Our Results

We extend the general framework of [33], to obtain a composition theorem that allows us to establish adaptive UC-security in models both with, and without, trusted set-up. With this theorem, essentially all general UC-feasibility results for adaptive adversaries follow as simple corollaries, often improving the set-up and/or complexity theoretic assumptions; moreover, we obtain adaptive UC secure computation in new models (such as the timing model). Additionally, our work is the first to achieve bounded-concurrent adaptively-secure multiparty computation without setup assumptions. As such, similar to [33], our theorem takes a step towards characterizing those models in which adaptive UC security is realizable, and also at what cost.

Although technically quite different, as mentioned previously, our theorem can be viewed as an adaptive analogue of the work of Lin, Pass and Venkitasubramaniam

[1] In the CRS model, all parties have access to public reference string sampled from a prespecified distribution.

[33], who study the *static* case. Their work puts forward the very general notion of a "UC-puzzle" to capture the models (or setup assumptions) that admit general static UC-security. More precisely, they prove that if we assume the existence of enhanced trapdoor permutations and stand-alone non-malleable commitments, static UC-security is achievable in any model that admits a UC-puzzle. In this work, we establish an analogous result for the more difficult case of *adaptive* UC-security, as we outline below.

We start by introducing the notion of an *Adaptive UC-Puzzle*. Next, we define the new primitive (which may be of independent interest), *equivocal non-malleable commitment* or EQNMCom, which is a commitment with the property that a man-in-the-middle observing concurrent equivocal commitments and decommitments cannot break the binding property. We then present a construction of equivocal non-malleable commitment for any model that admits an adaptive UC-puzzle (thus, requiring this primitive does not introduce an additional complexity-theoretic assumption). Finally, we rely on a computational assumption that is known to imply adaptively secure OT (analogous to the eTDP used by [33], which implies statically secure OT). Specifically, we use *simulatable public key encryption* [18,13]. Although a weaker assumption, *trapdoor* simulatable public key encryption is known to imply semi-honest adaptively secure OT, it is unknown how to achieve malicious, adaptive, UC secure OT (in any setup model) from only trapdoor simulatable public key encryption. We remark here that, more recently, for the static case, Lin et al. show how to extend their framework and rely on the minimal assumptions of stand-alone semi-honest oblivious-transfer and static UC-puzzle [41]. More concretely, we show the following:

Theorem 1 (Main Theorem (Informal)). *Assume the existence of an adaptive UC-secure puzzle Σ using some setup $\mathcal{T}$, the existence of an* EQNMCom *primitive, and the existence of a simulatable public-key encryption scheme. Then, for every m-ary functionality f, there exists a protocol Π using the same set-up $\mathcal{T}$ that adaptively, UC-realizes f.*

As an immediate corollary of our theorem, it follows that to establish feasibility of adaptive UC-secure computation in any set-up model, it suffices to construct an adaptive UC-puzzle in that model. Complementing the main theorem, we show that in many previously studied models, adaptive UC-puzzles are easy to construct. Indeed, in many models the straightforward puzzle constructions for the static case (cf. [33]) are sufficient to obtain adaptive puzzles; some models require puzzle constructions that are more complex (see the full version [17] for details). We highlight some results below.

Adaptive UC in the "imperfect" String Model. Canetti, Pass and shelat [12] consider adaptive UC security where parties have access to an "imperfect" reference string–called a "sunspot"–that is generated by an arbitrary efficient min-entropy source (obtained e.g., by measurement of some physical phenomenon). The CPS-protocol requires m communicating parties to share m reference strings, each of them generated using fresh entropy. We show that a *single* reference string is sufficient for UC and adaptively-secure MPC (regardless of the number of parties m).

Adaptive UC in the Timing Model. Dwork, Naor and Sahai [22] introduced the *timing model* in the context of concurrent zero-knowledge, where all players are assumed to have access to clocks with a certain drift. Kalai, Lindell and Prabhakaran [30] subsequently presented a concurrent secure computation protocol in the timing model;

whereas the timing model of [22] does not impose a maximal upper-bound on the clock drift, the protocol of [30] requires the clock-drift to be "small"; furthermore, it requires extensive use of delays (roughly $n\Delta$, where Δ is the latency of the network). Finally, [33] showed that UC security against *static* adversaries is possible also in the *unrestricted* timing model (where the clock drift can be "large"); additionally, they reduce the use of delays to only $O(\Delta)$. To the best of our knowledge, our work is the first to consider security against adaptive adversaries in the timing model, giving the first feasibility results for UC and adaptively-secure MPC in the timing model; moreover, our results also hold in the unrestricted timing model.

Adaptive UC with Quasi-polynomial Simulation. Pass [39] proposed a relaxation of the standard simulation-based definition of security, allowing for super polynomial-time or Quasi-polynomial simulation (QPS). In the static and adaptive setting, Prabhakaran and Sahai [45] and Barak and Sahai [3] obtained general MPC protocols that are concurrently QPS-secure without any trusted set-up, but rely on strong complexity assumptions. We achieve adaptive security in the QPS model under relatively weak complexity assumptions. Moreover, we achieve a stronger notion of security, which (in analogy with [39]) requires that indistinguishability of simulated and real executions holds for all of quasi-polynomial time; in contrast, [3] only achieves indistinguishability w.r.t. distinguishers with running-time smaller than that of the simulator.

Adaptive UC with Non-uniform Simulation. Lin et al. [33] introduced the non-uniform UC model, which considers environments that are $\mathcal{PPT}$ machines and ideal-model adversaries that are non-uniform $\mathcal{PPT}$ machines and prove feasibility of MPC in the same model. Relying on the same assumptions as those introduced by [33] to construct a puzzle in non-uniform model (along with the assumption of the existence of simulatable PKE), we show feasibility results for secure MPC in the adaptive, non-uniform UC model.

Adaptive Bounded-Concurrent Secure Multiparty Computation. Several works [34,42,40] consider the notion of bounded-concurrency for general functionalities where a single secure protocol Π implementing a functionality f is run concurrently, and there is an *a priori* bound on the number of concurrent executions. In our work, we show how to construct an adaptive puzzle in the bounded-concurrent setting (with no setup assumptions). Thus, we achieve the first results showing feasibility of bounded-concurrency of general functionalities under adaptive corruptions.

In addition to these models, we obtain feasibility of adaptive UC in existing models such as the common reference string (CRS) model [11], uniform reference string (URS) model [11], key registration model [2], tamper-proof hardware model [31], and partially isolated adversaries model [20] (see the full version [17]). For relaxed security models, we obtain UC in the quasi-polynomial time model [39,45,3].

Beyond the specific instantiations, our framework provides conceptual simplicity, technical insight, and the potential to facilitate "translation" of results in the static setting into corresponding (and much stronger) adaptive security results. For example, in recent work of Garg et al. [24], one of the results—constructing UC protocols using multiple setups when the parties share an arbitrary belief about the setups—can be translated to the adaptive model by replacing (static) puzzles with our notion of adaptive puzzles. Other results may require more work to prove, but again are facilitated by our framework.

1.2 Technical Approach and Comparison with Previous Work

There are two basic properties that must be satisfied in order to achieve adaptive UC secure computation: (1) concurrent simulation and (2) concurrent non-malleability. The former requirement amounts to providing the simulator with a trapdoor while the latter requirement amounts to establishing independence of executions. The simulation part is usually "easy" to achieve. Consider, for instance, the *common random string* (CRS) or *Uniform Reference String* (URS) model where the players have access to a public reference string that is sampled uniformly at random. A trapdoor can be easily provided to the simulator as the inverse of the reference string under a pseudo-random generator. Concurrent non-malleability on the other hand is significantly harder to achieve. For the specific case of the CRS model, Canetti et al. [11] and subsequent works [23,37] show that adaptive security can be achieved using a single trapdoor. However, more general setup models require either strong computational assumptions, or provide the simulator with *different* and *independent* trapdoors for different executions. For example, in the URS model, [11] interpret the random string as a public-key for a CCA-secure encryption scheme, and need to assume dense cryptosystems, while in the imperfect random string (sunspot) model, [12] require multiple trapdoors. Other models follow a similar pattern, where concurrent non-malleability is difficult.

In the static case, [33] provided a framework that allowed to decouple the concurrent simulation requirement from the concurrent non-malleability. More precisely, they show that providing a (single) trapdoor to achieve concurrent simulation is sufficient, and once a trapdoor is established concurrent non-malleability can be obtained for free. This allows them to obtain significant improvement in computational/set-up assumptions since no additional assumptions are required to establish non-malleability.

A fundamental question is whether the requirement of concurrent simulation and concurrent non-malleability can be decoupled in the case of adaptive UC-security. Unfortunately, the techniques used in the static case are not applicable in the adaptive case. Let us explain the intuition. [33] and subsequent works rely on *stand-alone non-malleable* primitives to achieve concurrent non-malleability. An important reason this was possible in the static case is because non-malleable primitives can be constructed in the plain-model (i.e. assuming no trapdoor). Furthermore, these primitives inherently admit black-box simulation, i.e. involve the simulator *rewinding* the adversary. Unfortunately, in the adaptive case both these properties are difficult to achieve. First, primitives cannot be constructed in the plain model since adaptive security requires the simulator to be able to simultaneously *equivocate* the simulated messages for honest parties for different inputs and demostrate their validity at any point in the execution by revealing the random coins for the honest parties consistent with the messages. Second, as demostrated in [26], black-box rewinding techniques cannot be employed since the adversary can, in between messages, corrupt an arbitrary subset of the players (some not even participating in the execution) whose inputs are not available to the simulator.

In this work, we show, somewhat surprisingly that a single trapdoor is still sufficient to achieve concurrent non-malleability. Although we do not decouple the requirements, this establishes that even for the case of adaptive security no additional setup, and therefore, no additional assumptions, are required to achieve concurrent non-malleability, thereby yielding similar improvements to complexity and set-up assumptions to [33].

The basic approach we take resembles closely the unified framework of [33]. By relying on previous works [40,42,35,11,27], Lin et. al in [33] argue that to construct a UC protocol for realizing any multi-party functionality, it suffices to construct a zero-knowledge protocol that is concurrently simulatable and concurrently simulation-sound[2]. To formalize concurrent-simulation, they introduce the notion of a UC-puzzle that captures the property that no adversary can successfully complete the puzzle and also obtain a trapdoor, but a simulator exists that can generate (correctly distributed) puzzles together with trapdoors. To achieve simulation-soundness, they introduce the notion of strong non-malleable witness indistinguishability and show how a protocol satisfying this notion can be based on stand-alone non-malleable commitments.

A first approach for the adaptive case, would be to extend the techniques from [33], by replacing the individual components with analogues that are adaptively secure and rely on a similar composition theorem. While the notion of UC-puzzle can be strengthened to the adaptive setting, the composition theorem does not hold for stand-alone non-malleable commitments. This is because, in the static case, it is enough to consider a commitment scheme that is statistically-binding for which an indistinguishability-based notion of non-malleability is sufficient; such a notion, when defined properly, is concurrently composable. However, when we consider adaptive security, commitments need to be equivocable (i.e., the simulator must be capable of producing a fake commitment transcript and inputs for honest committers that allow the transcript to be decommitted to both 0 and 1) and such commitments cannot be statistically-binding. Therefore, we need to consider a stronger simulation-based notion of non-malleability. Furthermore, as mentioned before, an equivocal commitment, even in the stand-alone case, requires the simulator to have a trapdoor, which in turn requires some sort of a trusted set-up.

Our approach here is to consider a “strong” commitment scheme, one that is both equivocable and concurrently non-malleable at the same time, but relies on a UC-puzzle (i.e. single trapdoor) and then establish a new composition theorem that essentially establishes feasibility of UC-secure protocol in any setup that admits a UC-puzzle. While the core contribution of [33] was in identifying the right notion of UC-puzzle and providing a modular analysis, in this work, the main technical novelty is in identifying the right notion of commitment that will allow feasibility with a single trapdoor. Once this is established the results from [33] can be extended analogously by constructing an adaptively secure UC-puzzle for each model. In fact, in most of the models considered in this work, the puzzle constructions are essentially the same as the static case and thus we obtain similar corollaries to [33]. While the general framework for our work resembles [33], as we explain in the next section, the commitment scheme and the composition theorem are quite different and requires an intricate and subtle analysis.

1.3 Main Tool: Equivocal Non-malleable Commitments

We define and construct a new primitive called *equivocal non-malleable commitments* or $\mathsf{EQNMCom}$. Such commitments have previously been defined in the works of [15,16] but only for the limited case of bounded concurrency and non-interactive commitments. In our setting, we consider the more general case of unbounded concurrency as well as

[2] Simulation-soundness is a stronger property that implies and is closely related to non-malleability.

interactive commitments. Intuitively, the property we require from these commitments is that even when a man-in-the-middle receives concurrent equivocal commitments and concurrent equivocal decommitments, the man-in-the-middle cannot break the binding property of the commitment. Thus, the man-in-the-middle receives equivocal commitments and decommitments, but cannot equivocate himself. Formalizing this notions seems to be tricky and has not been considered in literature before. Previously, non-malleability of commitments has been dealt with in two scenarios:

Non-malleability w.r.t commitment:[21,43,32] This requires that no adversary that receives a commitment to value v be able to commit to a related value (even without being able to later decommit to this value).

Non-malleability w.r.t decommitment (or opening):[15,43,19] This requires that no adversary that receives a commitment and decommitment to a value v be able to commit and decommit to a related value.

While the former is applicable only in the case the of statistically-binding commitments the latter is useful even for statistically-hiding commitments. In this work, we need a definition that ensures independence of commitments schemes that additionally are equivocable and adaptively secure. Equivocability means that there is a way to commit to the protocol without knowing the value being committed to and later open to any value. Such a scheme cannot be statistically-binding. Furthermore, since we consider the setting where the adversary receives concurrent equivocal decommitments, our definition needs to consider non-malleability w.r.t decommitment. Unfortunately, current definitions for non-malleability w.r.t decommitment in literature are defined only in the scenario where the commitment phase and decommitment phases are decoupled, i.e. in a first phase, a man-in-the-middle adversary receives commitments and sends commitments, then, in a second phase, the adversary requests decommitments of the commitments received in the first phase, followed by it decommitting its own commitments. For our construction, we need to define concurrent non-malleability w.r.t decommitments and such a two phase scenario is not applicable as the adversary can arbitrarily and adaptively decide when to obtain decommitments. Furthermore, it is not clear how to extend the traditional definition to the general case, as at any point, only a subset of the commitments received by the adversary could be decommitted and the adversary could selectively decommit based on the values seen so far and hence it is hard to define a "related" value.

We instead propose a new definition, along the lines of *simulation-extractability* that has been defined in the context of constructing non-malleable zero-knowledge proofs [44]. Loosely speaking, an interactive protocol is said to be simulation extractable if for any man-in-the-middle adversary A, there exists a probabilistic polynomial time machine (called the simulator-extractor) that can simulate both the left and the right interaction for A, while outputting a witness for the statement proved by the adversary in the right interaction. Roughly speaking, we say that a tag-based commitment scheme (i.e., commitment scheme that takes an identifier—called the tag—as an additional input) is *concurrent non-malleable w.r.t opening* if for every man-in-the-middle adversary A that participates in several interactions with honest committers as a receiver (called *left* interactions) as well as several interactions with honest receivers as a committer (called *right* interactions), there exists a simulator S that can simulate the left interactions, while extracting the commitments made by the adversary in the right interactions (whose identifiers are different from all the left identifiers) before the adversary decommits.

A related definition in literature is that of *simulation-sound trapdoor commitments* from [23,37] which considers *non-interactive* equivocable commitments and require that no adversary be able to equivocate even when it has access to an oracle that provides equivocal commitments and decommitments. This can be thought of as the CCA analogue for equivocal commitments. We believe that such a scheme would suffice for our construction, however, it is not clear how to construct such commitments from any trapdoor (i.e. any set-up) even if we relax the definition to consider interactive commitments.

It is not hard to construct equivocal commitments using trusted set-up. The idea here is to provide the simulator with a trapdoor with which it can equivocate as wells as extract the commitments on the right. (by e.g., relying on encryption). However, to ensure non-malleability, most constructions in literature additionally impose CCA-security or provide independent trapdoors for every interaction. Our main technical contribution consists of showing how to construct a concurrent non-malleable commitment scheme in any trusted set-up by providing the simulator with just one trapdoor, i.e. we show how to construct a concurrent non-malleable commitment scheme w.r.t opening using any UC-puzzle. We remark here that, in the static case, a stand-alone non-malleable commitment was sufficient, since the indistinguishability based notion of non-malleability allowed for some form of concurrent composition. However, in the adaptive case, it is not clear if our definition yields a similar composition and hence we construct a scheme and prove non-malleability directly in the concurrent setting.

Although our main application of equivocal non-malleable commitments is achieving UC-security, these commitments may also be useful for other applications such as concurrent non-malleable zero knowledge secure under adaptive corruptions. We believe that an interesting open question is to explore other applications of equivocal non-malleable commitments and non-malleable commitments with respect to decommitment.

2 Equivocal Non-malleable Commitments

In this section, we define Equivocal Non-malleable Commitments. Intuitively, these are equivocal commitments such that even when a man-in-the-middle adversary receives equivocal commitments and openings from a simulator, the adversary himself remains unable to equivocate. Since we are interested in constructing equivocal commitments from any trapdoor (i.e. setup), we will capture trapdoors, more generally, as witnesses to NP-statements. First, we provide definitions of language-based commitments.

Language-Based Commitment Schemes: We adopt a variant of language-based commitment schemes introduced by Lindell et. al [36] which in turn is a variant of [4,29]. Roughly speaking, in such commitments the sender and receiver share a common input, a statement x from an NP language L. The properties of the commitment scheme depend on the whether x is in L or not and the binding property of the scheme asserts that any adversary violating the binding can be used to extract an NP-witness for the statement. We present the formal definition below.

Definition 1 (Language-Based Commitment Schemes). *Let L be an NP-Language and $\mathcal{R}$, the associated NP-relation. A language-based commitment scheme (LBCS) for L is commitment scheme $\langle S, R\rangle$ such that:*

Computational hiding: *For every (expected) PPT machine R^*, it holds that, the following ensembles are computationally indistinguishable over $n \in N$.*

- $\{\mathsf{sta}^{R^*}_{\langle S,R\rangle}(x, v_1, z)\}_{n\in N, x\in\{0,1\}^n, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*}$
- $\{\mathsf{sta}^{R^*}_{\langle S,R\rangle}(x, v_2, z)\}_{n\in N, x\in\{0,1\}^n, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*}$

where $\mathsf{sta}^{R^*}_{\langle S,R\rangle}(x, v, z)$ denotes the random variable describing the output of $R^*(x, z)$ after receiving a commitment to v using $\langle S, R\rangle$.

Computational binding: *The binding property asserts that, there exists an polynomial-time witness-extractor algorithm Ext, such that for any cheating committer S^*, that can decommit a commitment to two different values v_1, v_2 on common input $x \in \{0,1\}^n$, outputs w such that $w \in \mathcal{R}(x)$.*

We now extend the definition to include equivocability.

Definition 2 (Language-Based Equivocal Commitments). *Let L be an* NP*-Language and $\mathcal{R}$, the associated* NP*-relation. A language-based commitment scheme $\langle S, R\rangle$ for L is said to be equivocal, if there exists a tuple of algorithms $(\tilde{S}, \mathsf{Adap})$ such that the following holds:*

Special-Hiding: *For every (expected) PPT machine R^*, it holds that, the following ensembles are computationally indistinguishable over $n \in N$.*

- $\{\mathsf{sta}^{R^*}_{\langle S,R\rangle}(x, v_1, z)\}_{n\in N, x\in L\cap\{0,1\}^n, w\in\mathcal{R}(x), v_1\in\{0,1\}^n, z\in\{0,1\}^*}$
- $\{\mathsf{sta}^{R^*}_{\langle \tilde{S},R\rangle}(x, w, z)\}_{n\in N, x\in L\cap\{0,1\}^n, w\in\mathcal{R}(x), v_1\in\{0,1\}^n, z\in\{0,1\}^*}$

where $\mathsf{sta}^{R^*}_{\langle \tilde{S},R\rangle}(x, w, z)$ denotes the random variable describing the output of $R^*(x, z)$ after receiving a commitment using $\langle \tilde{S}, R\rangle$.

Equivocability: *Let τ be the transcript of the interaction between R and $\tilde{S}$ on common input $x \in L \cap \{0,1\}^n$ and private input $w \in \mathcal{R}(x)$ and random tape $r \in \{0,1\}^*$ for $\tilde{S}$. Then for any $v \in \{0,1\}^n$, $\mathsf{Adap}(x, w, r, \tau, v)$ produces a random tape r' such that (r', v) serves as a valid decommitment for C on transcript τ.*

2.1 Definition of Equivocal Non-malleable Commitments

Let $\langle C, R\rangle$ be a commitment scheme, and let $n \in N$ be a security parameter. Consider man-in-the-middle adversaries that are participating in left and right interactions in which $m = \text{poly}(n)$ commitments take place[3]. We compare between a *man-in-the-middle* and a *simulated* execution. In the man-in-the-middle execution, the adversary A is simultaneously participating in m left and right interactions. In the left interactions the man-in-the-middle adversary A interacts with C receiving commitments to values $v_1, \ldots, v_m$, using identities $\mathsf{id}_1, \ldots, \mathsf{id}_\mathsf{m}$ of its choice. It must be noted here that values $v_1, \ldots, v_m$ are provided to committer on the left prior to the interaction. In the right interaction A interacts with R attempting to commit to a sequence of related values again

[3] We may also consider relaxed notions of concurrent non-malleability: one-many, many-one and one-one secure non-malleable commitments. In a one-one (i.e., a stand-alone secure) non-malleable commitment, we consider only adversaries A that participate in one left and one right interaction; in one-many, A participates in one left and many right, and in many-one, A participates in many left and one right.

using identities of its choice $\tilde{\mathsf{id}}_1, \ldots, \tilde{\mathsf{id}}_m$; $\tilde{v}_i$ is set to the value decommitted by A in the j^{th} right interaction. If any of the right commitments are invalid its committed value is set to $\perp$. For any i such that $\tilde{\mathsf{id}}_i = \mathsf{id_j}$ for some j, set $\tilde{v}_i = \perp$—i.e., any commitment where the adversary uses the same identity as one of the honest committers is considered invalid. Let $\mathsf{MIM}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)$ denote a random variable that describes the values $\tilde{v}_1, \ldots, \tilde{v}_m$ and the view of A, in the above experiment.

In the simulated execution, a simulator S interacts only with receivers on the right as follows:

1. Whenever the commitment phase of j^{th} interaction with a receiver on the right is completed, S outputs a value $\tilde{v}_j$ as the alleged committed value in a special-output tape.
2. During the interaction, S may output a partial view for a man-in-the-middle adversary whose right-interactions are identical to S's interaction so far. If the view contains a left interaction where the i^{th} commitment phase is completed and the decommitment is requested, then a value v_i is provided as the decommitment.
3. Finally, S outputs a view and values $\tilde{v}_1, \ldots, \tilde{v}_m$. Let $\mathsf{sim}^S_{\langle C,R\rangle}(1^n, v_1, \ldots, v_m, z)$ denote the random variable describing the view output by the simulation and values $\tilde{v}_1, \ldots, \tilde{v}_m$.

Definition 3. *A commitment scheme $\langle C, R\rangle$ is said to be* concurrent non-malleable w.r.t. opening *if for every polynomial $p(\cdot)$, and every probabilistic polynomial-time man-in-the-middle adversary A that participates in at most $m = p(n)$ concurrent executions, there exists a probabilistic polynomial time simulator S such that the following ensembles are computationally indistinguishable over $n \in N$:*

$$\left\{\mathsf{MIM}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)\right\}_{n\in N, v_1,\ldots,v_m\in\{0,1\}^n, z\in\{0,1\}^*}$$

$$\left\{\mathsf{sim}^S_{\langle C,R\rangle}(1^n, v_1, \ldots, v_m, z)\right\}_{n\in N, v_1,\ldots,v_m\in\{0,1\}^n, z\in\{0,1\}^*}$$

A slightly relaxed definition considers all the values committed to the adversary in the left interaction to be sampled independently from an arbitrary distribution D. We show how to construct a commitment satisfying only this weaker definition. However, this will be sufficient to establish our results.

Definition 4. *A commitment scheme $\langle C, R\rangle$ is said to be* concurrent non-malleable w.r.t. opening with independent and identically distributed (i.i.d.) commitments *if for every polynomial $p(\cdot)$ and polynomial time samplable distribution D, and every probabilistic polynomial-time man-in-the-middle adversary A that participates in at most $m = p(n)$ concurrent executions, there exists a probabilistic polynomial time simulator S such that the following ensembles are computationally indistinguishable over $n \in N$:*

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{MIM}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)\right\}_{n\in N, z\in\{0,1\}^*}$$

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{sim}^S_{\langle C,R\rangle}(1^n, v_1, \ldots, v_m, z)\right\}_{n\in N, z\in\{0,1\}^*}$$

Remark 1. Any scheme that satisfies our definition with a straight-line simulator in essence realizes the ideal commitment functionality with UC-security as it acheives equivocation and straight-line extraction. If the simulator is not straight-line, then the requirement that the left commitments are sampled from i.i.d distributions is seemingly inherent. This is because our definition implies security against *selective openning attacks* (SOA) and as proved in [38], achieving fully concurrent SOA-security with (black-box) rewinding techniques is impossible when the distributions of the commitments are not sampleable (or unknown).

Finally, we consider commitment schemes that are both non-malleable w.r.t opening and language-based equivocal. In a *setup model*, the simulator will obtain a trapdoor via the setup procedure and the witness relation will satisfy that language requirement.

Definition 5. *A commitment scheme $\langle C, R\rangle$ is said to be an* equivocal non-malleable commitment scheme *if it is both a language-based equivocal commitment scheme (see Definition 2) and is concurrent non-malleable w.r.t. opening (see Definition 4).*

3 Adaptive UC-Puzzles

Informally, an adaptive UC-puzzle is a protocol $\langle \mathsf{S}, \mathsf{R}\rangle$ between two players–a *sender* S and a *receiver* R – and a PPT computable relation $\mathcal{R}$, such that the following two properties hold:

Soundness: No efficient receiver R^* can successfully complete an interaction with S and also obtain a "trapdoor" y, such that $\mathcal{R}(\mathsf{TRANS}, y) = 1$, where TRANS is the transcript of the interaction.

Statistical UC-simulation with adaptive corruptions: For every efficient adversary $\mathcal{A}$ participating in a polynomial number of concurrent executions with receivers R (i.e., $\mathcal{A}$ is acting as a puzzle sender in all these executions) and at the same time communicating with an environment $\mathcal{Z}$, there exists a simulator $\mathcal{S}$ that is able to statistically simulate the view of $\mathcal{A}$ for $\mathcal{Z}$, while at the same time outputting trapdoors to all successfully completed puzzles. Moreover, $\mathcal{S}$ successfully simulates the view even when $\mathcal{A}$ may adaptively corrupt the receivers.

We provide a formal definition in the full version [17]. In essence, it is the same definition as in [33] with the additional requirement of adaptive security in the simulation. We remark that our analysis will require the puzzle to be **straight-line simulatable**. In fact, for almost all models considered in this work, this is indeed the case, with the exception of the timing and partially-isolated adversaries model (for which we argue the result independently). Using the result of [26], it is possible to argue that straight-line simulation is necessary to achieve adaptive security (except when we consider restricted adversaries, such as the timing or partially-isolated adversaries model).

4 Achieving Adaptive UC-Security

In this section, we give a high-level overview of the construction of an $\mathsf{EQNMCom}$ scheme and the proof of Theorem 1, which relies on the existence of an $\mathsf{EQNMCom}$

scheme. For the formal construction and analysis of our $\mathsf{EQNMCom}$ scheme, see the full version [17]. A formal proof of Theorem 1 can be found in the full version [17].

By relying on previous results [11,18,28,14,13], the construction of an adaptive UC-secure protocol for realizing any multiparty functionality can be reduced to the task of constructing a UC-commitment assuming the existence of simulatable PKE. First, we show how to construct an equivocal non-malleable commitment scheme based on any adaptive UC-puzzle. Then combining the equivocal non-malleable commitment scheme with a simulatable PKE scheme we show how to realize the UC-commitment.

4.1 Constructing EQNMCom Based on Adaptive UC-Puzzles

Our protocol on a very high-level is a variant of the non-malleable commitment protocol from [32] which in turn is a variant of the protocol from [21]. While non-malleability relies on the message-scheduling technique of [21,32] protocol, the equivocability is obtained by relying on a variant of Feige-Shamir's trapdoor commitment scheme[4] and adaptively secure witness-indistinguishable proof of knowledge (WIPOK) protocol (see the full version [17]) for a formal definition and construction) of Lindell-Zarosim[36]. More precisely, our protocol proceeds in two phases: in the preamble phase, the Committer and Receiver exchange a UC-puzzle where the Receiver is the sender of the puzzle and the Committer is the receiver of the puzzle (this phase establishes a trapdoor through which an equivocal commitment can be generated). This is followed by the commitment phase: here the Committer first commits to its value using a language-based (non-interactive) equivocal commitment scheme, where the NP-language is the one corresponding to the UC-puzzle and the particular statement is the puzzle exchanged in the preamble (this relies on the Feige-Shamir trapdoor commitment scheme). This is followed by several invocations of an (adaptively-secure) WIPOK where the Committer proves the statement that either it knows the value committed to in phase 2 or possesses a solution to the puzzle from phase 1. Here we rely on the *adaptively-secure* (without erasures) WIPOK of [36] where the messages are scheduled based on the Committers id using the scheduling of [21]. This phase allows for any Committer that possess a solution to the puzzle from the preamble phase to generate a commitment that can be equivocated (i.e. later be opened to any value). Conversely, any adversary that can equivocate the non-interactive commitment of the second phase can be used to obtain a solution to the puzzle. The decommitment information is simply the value and the random tape of an honest committer consistent with the commitment phase. More specifically, our protocol proceeds as follows:

1. In the Preamble Phase, the Committer and Receiver exchange a UC-puzzle where the Receiver is the sender of the puzzle and the Committer is the receiver of the puzzle. Let x be the transcript of the interaction.
2. In the Committing Phase, the Committer sends $c = \mathsf{EQCom}^x(v)$, where EQCom^x is a language-based equivocal commitment scheme as in Definition 2 with common input x. This is followed by the Committer proving that c is a valid commitment

[4] Let x be an NP-statement. The sender commits to bit b by running the honest-verifier simulator for Blum's Hamiltonian Circuit protocol [5] on input the statement x and the verifier message b, generating the transcript (a, b, z), and finally outputting a as its commitment. In the decommitment phase, the sender reveals the bit b by providing both b, z.

for v. This is proved by 4ℓ invocations of an adaptively-secure (without erasures) WIPOK where the messages are scheduled based on the id (as in [21,32]). More precisely, there are ℓ rounds, where in round i, the schedule $\mathsf{design}_{\mathsf{id}_i}$ is followed by $\mathsf{design}_{1-\mathsf{id}_i}$ (See Figure 1).

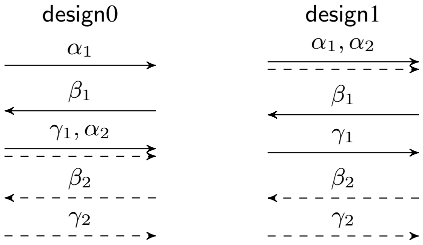

Fig. 1. Message schedule in a round in adaptively-secure WIPOK

While the protocol is an adaptation of the [32] commitment scheme, where the individual components are replaced by adaptively-secure alternatives, proving security requires a substantially different analysis. It is easy to see that concurrent equivocability of our scheme follows from the UC-Puzzle simulation. However proving concurrent non-malleability w.r.t opening with i.i.d commitments is the hard part and the core of our contribution. Recall that, achieving this, essentially entails constructing a simulator for any man-in-the-middle adversary, that while equivocating all commitments to the adversary (in the left interactions), can extract all the values the value committed to by the adversary (in the right interactions) before the decommitment phase.

Towards extracting from the right interactions, we first recall the basic idea in [32,21]. Their scheduling ensures that for every right interaction with a tag that is different from a left interaction, there exists a point—called a safe-point—from which we can *rewind* the right interaction (and extract the committed value), without violating the hiding property of the left interaction. It now follows from the hiding property of the left interaction that the values committed to on the right do not depend on the value committed to on the left. However, this technique only allows for extraction from a right interaction without violating the hiding property of *one* left interaction. However, here we need to extract without violating the hiding property of all the left interactions.

Our simulator-extractor as follows: In a main execution with the man-in-the-middle adversary, the simulator simulates all puzzles to obtain trapdoors and equivocates the left interactions using the solution of the puzzle and simulates the right interactions honestly. Whenever a decommitment on the left is requested, the simulator obtains a value externally (a value sampled independently from distribution D) which it decommits to the adversary (this is achieved since the protocol is adaptively secure). After the adversary completes the commitment phase of a right interaction in the main execution, the simulator switches to a rewinding phase, where it tries to extract the value committed to by the adversary in that right interaction. Towards this, it chooses a random WIPOK (instead of a safe point) from the commitment phase and rewinds the adversary to obtain the witness used in the WIPOK (using the proof-of-knowledge extractor). In the rewinding phase, the left interactions are now simulated using the honest committer strategy (as opposed to equivocating using the solution to the puzzle). More precisely, in the rewinding phase, for every left interaction that has already been opened (i.e. decommitment phase has occurred in the main execution), the simulator has a value and random tape for an honest committer and for those that have not yet been opened, using the adaptive-security of the protocol, the simulator simply samples a random value from distribution D (since we consider only i.i.d. values for left interactions) and generates a random tape for an honest committer consistent with the transcript so far. This stands in contrast of extracting only from safe-points as in [32].

The proof proceeds using a hybrid argument, where in hybrid experiment H_i all puzzle interactions are simulated and the first i left commitments to complete the preamble phase is equivocated. It will follow from the soundness of the UC-puzzle and statistical simulation that the simulation is correct H_0. First, we show that in H_0, the value extracted in any particular right interaction from a random WIPOK is the value decommitted to by the adversary. This follows from the fact that for the adversary to equivocate, it must know the solution to the UC-puzzle and this violates the statistical simulation and soundness condition of the puzzle. We show the following properties for every i, and the proof of correctness follows using a standard hybrid argument.

- *If the value extracted in any particular right interaction from a random WIPOK is the value decommitted to by the adversary in H_{i-1}, then the value extracted from a random WIPOK and the safe point of that right interaction w.r.t to i^{th} left interaction are the same and equal to the decommitment.* We show this by carefully considering another sequence of hybrids that yields an adversary that violates the soundness of the UC-puzzle in an execution where the puzzles are not simulated. This will rely on fact that the simulator simulates the left interactions in the rewindings using the honest committer strategy and the pseudo-randomness of the non-interactive commitment scheme used in the Commitment phase.
- *If the value extracted from the safe point is the decommitment in H_{i-1} then the same holds in H_i.* We rely on the proof technique of [32] through safe-points to establish this. In slightly more detail, we show that for any particular right interaction, the value extracted from the safe-point w.r.t i^{th} left interaction does not change when the i^{th} left commitment is changed from an honest commitment to an equivocal commitment. Recall that a safe-point can be used to extract the value committed to in the right without rewinding the particular left interaction. Since, the non-interactive commitment scheme used has pseudo-random commitments, an adversary cannot distinguish if it is receiving an honest or equivocal commitment in the i^{th} interaction.
- *If the value extracted in the right interaction from the safe point is the value decommitted to by the adversary in H_i, then the value extracted from a random WIPOK and the safe point are the same and equal to the decommitment in H_i.* This is established exactly as the first property.

See the full version [17] for the formal construction and proof.

4.2 Adaptive UC-Secure Commitment Scheme

We now provide the construction of a UC-commitment scheme. First, we recall the construction of the adaptive UC-secure commitment in the common reference string model (CRS) from [11] to motivate our construction. In the [11] construction, the CRS contains two strings. The first string consists of a random image $y = f(x)$ of a one-way function f and the second string consists of a public key for a cca-secure encryption scheme. The former allows a simulator to equivocate the commitment when it knows x and the public key allows the simulator to extract committed values from the adversary using its knowledge of the corresponding secret-key. The additional CCA requirement ensures non-malleability.

Our construction follows a similar approach, with the exception that instead of having a common reference string generated by a trusted party, we use the equivocal non-malleable commitment to generate two common-reference strings between every pair of parties: one for equivocation and the other for extraction. This is achieved by running the following "non-malleable" coin-tossing protocol between an initiator and a responder. Let $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ be a concurrent equivocal non-malleable commitment scheme and $\langle \mathsf{S}_{\mathsf{puz}}, \mathsf{R}_{\mathsf{puz}} \rangle$ be a UC-puzzle.

1. The initiator commits to a random string r^0 using $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ to the responder.
2. The responder chooses a random string r^1 and sends to the Initiator.
3. The initiator opens its commitment and reveals r^0.
4. The output of the coin toss is: $r = r^0 \oplus r^1$.

The coin-tossing protocol is run between an initiator and responder and satisfies the following two properties: (1) For all interactions where the initiator is honest, there is a way to simulate the coin-toss. This follows directly from the equivocability of the commitment scheme $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$. (2) For all interactions where the initiator is controlled by the adversary, the coin-toss generated is uniformly-random. This follows from the simulation-extractability of the commitment scheme.

Using the coin-tossing protocol we construct an adaptive UC-commitment scheme. First, the sender and receiver interact in two coin-tossing protocols, one where the sender is the initiator, with outcome $coin_1$ and the other, where the receiver is the initiator, with outcome $coin_2$. Let x be the statement that $coin_1$ is in the image of a pseudorandom generator G. Also let, $\text{PK} = \mathsf{oGen}(coin_2)$ be a public key for the simulatable encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{oGen}, \mathsf{oRndEnc}, \mathsf{rGen}, \mathsf{rRndEnc})$. To commit to a string β, the sender sends a commitment to β using the non-interactive language-based commitment scheme with statement x along with strings S_0 and S_1 where one of the two strings (chosen at random) is an encryption of decommitment information to β and the other string is outputted by $\mathsf{oRndEnc}$. In fact, this is identical to the construction in [11], with the exception that a simulatable encryption scheme is used instead of a CCA-secure scheme.

Binding follows from the soundness of the adaptive UC-puzzle and hiding follows from the hiding property of the non-interactive commitment scheme and the semantic security of the encryption scheme. It only remains to show that the scheme is concurrently equivocable and extractable. To equivocate a commitment from a honest committer, the simulator manipulates $coin_1$ (as the honest party is the initiator) so that $coin_1 = G(s)$ for a random string s and then equivocates by equivocating the non-interactive commitment and encrypting the decommitment information for both bits 0 and 1 in S_b and S_{1-b} (where b is chosen at random). To extract a commitment made by the adversary, the simulator manipulates $coin_2$ so that $coin_2 = \mathsf{rGen}(r)$ and

$(\text{PK}, \text{SK}) = \mathsf{Gen}(r)$ for a random string r. Then it extracts the decommitment information in the encryptions sent by the adversary using SK.

The procedure described above works only if the adversary does not encrypt the decommitment information for both 0 and 1 even when the simulator is equivocating. On a high-level, this follows since, if the coin-toss $coin_1$ cannot be manipulated by the adversary when it is the initiator, then the $coin_1$ is not in the range of G with very high probability and hence the adversary cannot equivocate (equivocating implies a witness can be extracted that proves that $coin_1$ is in the range of G). Proving this turns out to be subtle and an intricate analysis relying on the simulation-extractability of the $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$-scheme is required.

We use a "non-malleable" coin-toss protocol to generate two keys, one for equivocation and another for extraction. Such an idea has been pursued before, for instance, in [19], they use a coin-toss to generate keys for extraction and equivocation. However, they use a single coin-toss and depending on which party is corrupt, the simulation yields an extraction or equivocation key. In recent and independent work, Garg and Sahai [26], show how to achieve stand-alone adaptively-secure multiparty computation in the plain model (assuming no-setup) using non black-box simulation. They rely on a coin-tossing protocol using equivocal commitments to generate a common random string and then rely on previous techniques used in the uniform reference string model [11] to securely realize any functionality. An important difference between their approach and ours is that while our construction relies on a single trapdoor they require the trapdoors to be non-malleable.[5] See Figure 2 for a formal description of our protocol (For further details and the proof, we refer the reader to the full version [17]).

5 Puzzle Instantiations

By Theorem 1, it suffices to present an adaptive UC puzzle in a given model to demonstrate feasibility of adaptive and UC secure computation. We first give some brief intuition on the construction of adaptive UC-puzzles in various models. Formal constructions and proofs are found in the full version [17].

In the Common reference string (CRS) model, the Uniform reference string (URS) model and the Key registration model the puzzles are identical to the ones presented in [33] for the static case, where the puzzle interactions essentially consists of a call to the corresponding ideal setup functionalities. Thus, in these models, the simulator is essentially handed the trapdoor for the puzzle via its simulation of the ideal functionality and the puzzles are non-interactive. In the Timing model and the Partially Isolated Adversaries model, we rely on essentially the same puzzles as [33], however, we need to modify the simulator to accommodate adaptive corruption by the adversary.

Constructing adaptive UC-puzzles in the Sunspots model is less straightforward and so we give more detail here. Simulated reference strings r in the Sunspots model have Kolmogorov complexity smaller than k. Thus, as in [33], the puzzle sender and receiver exchange 4 strings (v_1, c_2, v_2, c_2). We then let Φ' denote the statement that c_1, c_2 are commitments to messages p_1, p_2 such that (v_1, p_1, v_2, p_2) is an accepting transcript of

[5] In [19], they use separate keys for each party and in [26], the trapdoors admit a "simulation-soundness" property.

Protocol $\langle \mathsf{S}, \mathsf{R} \rangle$: Input: The sender S has a bit β to be committed to.

Preamble:

- An adaptive UC-Puzzle interaction $\langle \mathsf{S}_{\mathsf{puz}}, \mathsf{R}_{\mathsf{puz}} \rangle$ on input 1^n where R is the receiver and S is the sender. Let TRANS_1 be the transcript of the messages exchanged.
- An adaptive UC-Puzzle interaction $\langle \mathsf{S}_{\mathsf{puz}}, \mathsf{R}_{\mathsf{puz}} \rangle$ on input 1^n where S is the receiver and R is the sender. Let TRANS_2 be the transcript of the messages exchanged.

Commit phase:

Stage 1: S and R run a coin-tossing protocol to agree on strings PK and CRS:

Coin-toss to generate PK:

1. The parties run protocol $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ with common input TRANS_1. R plays the part of sender with input a random string r^0_{R}.
2. S chooses a random string r^0_{S} and sends to R.
3. R opens its commitment and reveals r^0_{R}.
4. The output of the coin toss is: $r = r^0_{\mathsf{S}} \oplus r^0_{\mathsf{R}}$. S and R run $\mathsf{oGen}(r)$ to obtain public key PK.

Coin-toss to generate CRS:

1. The parties run protocol $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ with common input TRANS_2. S plays the part of sender with input a random string r^1_{S}.
2. R chooses a random string r^1_{R} and sends to S.
3. S opens its commitment and reveals r^1_{S}.
4. The output of the coin-toss is: $x = r^1_{\mathsf{S}} \oplus r^1_{\mathsf{R}}$.

Stage 2:

1. The parties run $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ with common input x to generate a commitment $C = \mathsf{EQCom}^x(\beta; r)$ where S plays the part of S_{eq} with input bit β.
2. S chooses $b \in \{0, 1\}$ at random and sends to R the strings (S_0, S_1) to where:
   - S_b is an encryption of the decommitment information of C (to bit β) under PK.
   - S_{1-b} is generated by running $\mathsf{oRndEnc}(\text{PK}, r_{\mathsf{Enc}})$ where r_{Enc} is chosen uniformly at random.

Reveal phase:

1. S sends β, b, and the randomness used to generate S_0, S_1 to R.
2. R checks that S_0, S_1 can be reconstructed using β, b and the randomness produced by S.

Fig. 2. The Adaptive Commitment Protocol $\langle \mathsf{S}, \mathsf{R} \rangle$

a Universal argument of the statement $\Phi = \mathsf{KOL}(r) \leq k$. Note that since we require *statistical* and *adaptive* simulation of puzzles, the commitment scheme used must be both statistically-hiding and ”obliviously samplable” (i.e. there is a way to generate strings that are statistically indistinguishable from commitments, without ”knowing” the committed value).

To construct an adaptive puzzle for the bounded-concurrent model we follow an approach similar to the sunspots model combined with the bounded-concurrent non black-box zero-knowledge protocol of Barak[1]. In fact this is inspired by the stand alone adaptive secure multiparty computation construction of Garg, et al, [26].

References

1. Barak, B.: How to go beyond the black-box simulation barrier. In: FOCS, pp. 106–115 (2001)
2. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: FOCS, pp. 186–195 (2004)
3. Barak, B., Sahai, A.: How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In: FOCS, pp. 543–552 (2005)
4. Bellare, M., Micali, S., Ostrovsky, R.: The (true) complexity of statistical zero knowledge. In: STOC, pp. 494–502 (1990)
5. Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians, pp. 1444–1451 (1986)
6. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS, pp. 136–145 (2001)
7. Canetti, R., Dodis, Y., Pass, R., Walfish, S.: Universally composable security with global setup. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 61–85. Springer, Heidelberg (2007)
8. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (2001)
9. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 68–86. Springer, Heidelberg (2003)
10. Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: FOCS, pp. 541–550 (2010)
11. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: STOC, pp. 494–503 (2002)
12. Canetti, R., Pass, R., Shelat, A.: Cryptography from sunspots: How to use an imperfect reference string. In: FOCS, pp. 249–259 (2007)
13. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Improved non-committing encryption with applications to adaptively secure protocols. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 287–302. Springer, Heidelberg (2009)
14. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Simple, black-box constructions of adaptively secure protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 387–402. Springer, Heidelberg (2009)
15. Di Crescenzo, G., Ishai, Y., Ostrovsky, R.: Non-interactive and non-malleable commitment. In: STOC, pp. 141–150 (1998)
16. Di Crescenzo, G., Katz, J., Ostrovsky, R., Smith, A.: Efficient and non-interactive non-malleable commitment. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, p. 40. Springer, Heidelberg (2001)
17. Dachman-Soled, D., Malkin, T., Raykova, M., Venkitasubramaniam, M.: Adaptive and concurrent secure computation from new notions of non-malleability. IACR Cryptology ePrint Archive, 2011:611 (2011)
18. Damgård, I.B., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000)
19. Damgård, I.B., Nielsen, J.B.: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 581–596. Springer, Heidelberg (2002)
20. Damgård, I., Nielsen, J.B., Wichs, D.: Universally composable multiparty computation with partially isolated parties. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 315–331. Springer, Heidelberg (2009)

21. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM J. Comput. 30(2), 391–437 (2000)
22. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. In: IN 30TH STOC, pp. 409–418 (1999)
23. Garay, J.A., MacKenzie, P.D., Yang, K.: Strengthening zero-knowledge protocols using signatures. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 177–194. Springer, Heidelberg (2003)
24. Garg, S., Goyal, V., Jain, A., Sahai, A.: Bringing people of different beliefs together to do UC. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 311–328. Springer, Heidelberg (2011)
25. Garg, S., Goyal, V., Jain, A., Sahai, A.: Concurrently secure computation in constant rounds. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 99–116. Springer, Heidelberg (2012)
26. Garg, S., Sahai, A.: Adaptively secure multi-party computation with dishonest majority. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 105–123. Springer, Heidelberg (2012)
27. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC, pp. 218–229 (1987)
28. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008)
29. Itoh, T., Ohta, Y., Shizuya, H.: A language-dependent cryptographic primitive. J. Cryptology 10, 37–50 (1997)
30. Kalai, Y.T., Lindell, Y., Prabhakaran, M.: Concurrent composition of secure protocols in the timing model. J. Cryptology 20(4), 431–492 (2007)
31. Katz, J.: Universally composable multi-party computation using tamper-proof hardware. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 115–128. Springer, Heidelberg (2007)
32. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent non-malleable commitments from any one-way function. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 571–588. Springer, Heidelberg (2008)
33. Lin, H., Pass, R., Venkitasubramaniam, M.: A unified framework for concurrent security: universal composability from stand-alone non-malleability. In: STOC, pp. 179–188 (2009)
34. Lindell, Y.: Protocols for bounded-concurrent secure two-party computation. Chicago J. Theor. Comput. Sci. (2006)
35. Lindell, Y.: Bounded-concurrent secure two-party computation without setup assumptions. In: STOC, pp. 683–692 (2003)
36. Lindell, Y., Zarosim, H.: Adaptive zero-knowledge proofs and adaptively secure oblivious transfer. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 183–201. Springer, Heidelberg (2009)
37. MacKenzie, P.D., Yang, K.: On simulation-sound trapdoor commitments. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 382–400. Springer, Heidelberg (2004)
38. Ostrovsky, R., Rao, V., Scafuro, A., Visconti, I.: Revisiting lower and upper bounds for selective decommitments. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 559–578. Springer, Heidelberg (2013)
39. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (2003)
40. Pass, R.: Bounded-concurrent secure multi-party computation with a dishonest majority. In: STOC, pp. 232–241 (2004)

41. Pass, R., Lin, H., Venkitasubramaniam, M.: A unified framework for UC from only OT. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 699–717. Springer, Heidelberg (2012)
42. Pass, R., Rosen, A.: Bounded-concurrent secure two-party computation in a constant number of rounds. In: FOCS, pp. 404–413 (2003)
43. Pass, R., Rosen, A.: Concurrent non-malleable commitments. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005, pp. 563–572 (2005)
44. Pass, R., Rosen, A.: New and improved constructions of non-malleable cryptographic protocols. In: STOC, pp. 533–542 (2005)
45. Prabhakaran, M., Sahai, A.: New notions of security: achieving universal composability without trusted setup. In: STOC, pp. 242–251 (2004)
46. Yao, A.C.-C.: How to generate and exchange secrets (extended abstract). In: FOCS, pp. 162–167 (1986)