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

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.

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) [49, 29]. 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, 37]. Previous works overcome this barrier either by using some trusted setup infrastructure [8, 11, 2, 7, 32, 12], or by relaxing the definition of security [42, 48, 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

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

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 [35, 36, 45, 43, 22, 32], 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 [35], 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 [35], 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 [35], 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 [35], which implies statically secure OT). Specifically, we use *simulatable public key encryption* [17, 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 [44]. 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. [35]) are sufficient to obtain adaptive puzzles; some models require puzzle constructions that are more complex (see Appendix E 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 [32] 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 [32] 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, [35] 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 [42] 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 [48] 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 [42]) 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. [35] 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 [35] 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 [36, 45, 43] 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 [33], and partially isolated adversaries model [20] (see Appendix E). For relaxed security models, we obtain UC in the quasi-polynomial time model [42, 48, 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, 39] 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, [35] 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. [35] 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 [35].

The basic approach we take resembles closely the unified framework of [35]. By relying on previous

works [43, 45, 37, 11, 29], Lin et. al in [35] 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 [35], 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 [35] 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 [35] 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 [35]. While the general framework for our work resembles [35], 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 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 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, 46, 34] 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).

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

Non-malleability w.r.t decommitment (or opening):[15, 46, 18] 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 [47]. 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, 39] 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 [38] which in turn is a variant of [4, 31]. 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 $m = \mathrm{poly}(n)$ 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 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 $\tilde{\mathsf{id}}_i = \mathsf{id}_\mathsf{j}$ 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.

[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.

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 [41], 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 Appendix B. In essence, it is the same definition as in [35] 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 EQNMCom scheme and the proof of Theorem 1, which relies on the existence of an EQNMCom scheme. For the formal construction and analysis of our EQNMCom scheme, see Appendix C. A formal proof of Theorem 1 can be found in Appendix D.

By relying on previous results [11, 17, 30, 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 [34] which in turn is a variant of the protocol from [21]. While non-malleability relies on the message-scheduling technique of [21, 34] 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 Appendix G for a formal definition and construction) of Lindell-Zarosim[38]. 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 [38] 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 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, 34]). 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).

While the protocol is an adaptation of the [34] 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

[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.

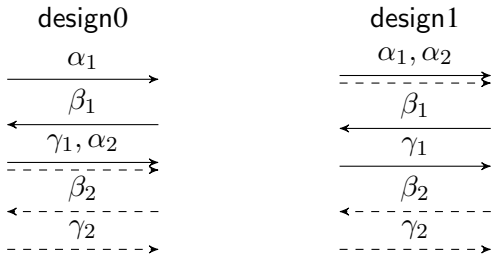

Figure 1: Message schedule in a round in adaptively-secure WIPOK

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 [34, 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 [34].

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 decom-*

mitted 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 [34] 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 Appendix C 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 pseudo-random 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 [18], 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] For details of the construction and proof, see Figure 2 and Appendix D.

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 follow.

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 [35] 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 [35], however, we need to modify the simulator to accommodate adaptive corruption by the adversary (see Section E.8 for more details).

Constructing adaptive UC-puzzles in the Sunspots model is less straightforward and so we give more

[5] In [18], they use separate keys for each party and in [26], the trapdoors admit a "simulation-soundness" property.

detail here. Simulated reference strings r in the Sunspots model have Kolmogorov complexity smaller than k. Thus, as in [35], 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 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). See Section E.6 for details.

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]. See Section E.7 for details.

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 $\mathsf{oRndEnc}(\mathrm{PK}, r_{\mathsf{Enc}})$ 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.

Figure 2: The Adaptive Commitment Protocol $\langle \mathsf{S}, \mathsf{R} \rangle$

References

[1] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001.

[2] Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *FOCS*, pages 186–195, 2004.

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

[4] M. Bellare, S. Micali, and R. Ostrovsky. The (true) complexity of statistical zero knowledge. In *STOC*, pages 494–502, 1990.

[5] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pages 1444–1451, 1986.

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

[7] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In *TCC*, pages 61–85, 2007.

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

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

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

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

[12] Ran Canetti, Rafael Pass, and Abhi Shelat. Cryptography from sunspots: How to use an imperfect reference string. In *FOCS*, pages 249–259, 2007.

[13] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In *ASIACRYPT*, pages 287–302, 2009.

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

[15] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *STOC*, pages 141–150, 1998.

[16] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Efficient and non-interactive non-malleable commitment. In *EUROCRYPT*, pages 40–59, 2001.

[17] Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *CRYPTO*, pages 432–450, 2000.

[18] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*, pages 581–596, 2002.

[19] Ivan Damgård, Jesper Buus Nielsen, and Daniel Wichs. Isolated proofs of knowledge and isolated zero knowledge. In *EUROCRYPT*, pages 509–526, 2008.

[20] Ivan Damgård, Jesper Buus Nielsen, and Daniel Wichs. Universally composable multiparty computation with partially isolated parties. In *TCC*, pages 315–331, 2009.

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

[22] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *IN 30TH STOC*, pages 409–418, 1999.

[23] Juan A. Garay, Philip D. MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. In *EUROCRYPT*, pages 177–194, 2003.

[24] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Bringing people of different beliefs together to do uc. In *TCC*, pages 311–328, 2011.

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

[26] Sanjam Garg and Amit Sahai. Adaptively secure multi-party computation with dishonest majority. In *CRYPTO*, pages 105–123, 2012.

[27] Oded Goldreich. *Foundations of Cryptography, vol. 1: Basic Tools*. Cambridge University Press, Cambridge, UK, 2001.

[28] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. In *Studies in Complexity and Cryptography*, pages 30–39. 2011.

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

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

[31] Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *J. Cryptology*, 10:37–50, 1997.

[32] Yael Tauman Kalai, Yehuda Lindell, and Manoj Prabhakaran. Concurrent composition of secure protocols in the timing model. *J. Cryptology*, 20(4):431–492, 2007.

[33] Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In *EUROCRYPT*, pages 115–128, 2007.

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

[35] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. A unified framework for concurrent security: universal composability from stand-alone non-malleability. In *STOC*, pages 179–188, 2009.

[36] Yehuda Lindell. Protocols for bounded-concurrent secure two-party computation. *Chicago J. Theor. Comput. Sci.*, 2006.

[37] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *STOC*, pages 683–692, 2003.

[38] Yehuda Lindell and Hila Zarosim. Adaptive zero-knowledge proofs and adaptively secure oblivious transfer. In *TCC*, pages 183–201, 2009.

[39] Philip D. MacKenzie and Ke Yang. On simulation-sound trapdoor commitments. In *EUROCRYPT*, pages 382–400, 2004.

[40] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for *np* using any one-way permutation. *J. Cryptology*, 11(2):87–108, 1998.

[41] Rafail Ostrovsky, Vanishree Rao, Alessandra Scafuro, and Ivan Visconti. Revisiting lower and upper bounds for selective decommitments. In *TCC*, pages 559–578, 2013.

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

[43] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *STOC*, pages 232–241, 2004.

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

[45] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *FOCS*, pages 404–413, 2003.

[46] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 563–572, 2005.

[47] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *STOC*, pages 533–542, 2005.

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

[49] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

A Definitions and Background

A.1 Commitment Schemes

Commitment schemes are used to enable a party, known as the *sender*, to commit itself to a value while keeping it secret from the *receiver* (this property is called hiding). Furthermore, in a later stage when the commitment is opened, it is guaranteed that the "opening" can yield only a single value determined in the committing phase (this property is called binding). In this work, we consider commitment schemes that are statistically-binding, namely while the hiding property only holds against computationally bounded (non-uniform) adversaries, the binding property is required to hold against unbounded adversaries. More precisely, a pair of PPT machines $\langle S, R \rangle$ is said to be a commitment scheme if the following two properties hold.

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

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

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

Statistical binding: Informally, the statistical-binding property asserts that, with overwhelming probability over the coin-tosses of the receiver R, the transcript of the interaction fully determines the value committed to by the sender. We refer the reader to [27] for more details.

We say that a commitment is *valid* if there exists a unique committed value that a (potentially malicious) committer can open to successfully.

A.2 Simulatable Encryption Schemes

Definition 6. *A* ℓ-bit simulatable encryption scheme *consists of an encryption scheme* $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ *augmented with* $(\mathsf{oGen},\mathsf{oRndEnc},\mathsf{rGen},\mathsf{rRndEnc})$. *Here,* oGen *and* $\mathsf{oRndEnc}$ *are the oblivious sampling algorithms for public keys and ciphertexts, and* rGen *and* $\mathsf{rRndEnc}$ *are the respective inverting algorithms,* rGen *(resp.* $\mathsf{rRndEnc}$*) takes* r_{G} *(resp.* $(\mathrm{PK},r_{\mathrm{E}},m)$*) as the trapdoor information. We require that, for all messages* $m\in\{0,1\}^\ell$*, the following distributions are computationally indistinguishable:*

$$\{\mathsf{rGen}(r_{\mathrm{G}}),\mathsf{rRndEnc}(\mathrm{PK},r_{\mathrm{E}},m),\mathrm{PK},c \mid (\mathrm{PK},\mathrm{SK})=\mathsf{Gen}(1^k;r_{\mathrm{G}}),c=\mathsf{Enc}_{\mathrm{PK}}(m;r_{\mathrm{E}})\}$$
$$\textit{and } \{\hat{r}_{\mathrm{G}},\hat{r}_{\mathrm{E}},\hat{\mathrm{PK}},\hat{c} \mid (\hat{\mathrm{PK}},\perp)=\mathsf{oGen}(1^k;\hat{r}_{\mathrm{G}}),\hat{c}=\mathsf{oRndEnc}_{\hat{\mathrm{PK}}}(1^k;\hat{r}_{\mathrm{E}})\}$$

It follows from the definition that a trapdoor simulatable encryption scheme is also semantically secure.

A.3 Traditional UC

Environment. The model of execution includes a special entity called the UC-environment (or environment) Z. The environment "manages" the whole execution: it invokes all the parties at the beginning of the execution, generates all inputs and reads all outputs, and finally produces an output for the whole concurrent execution. Intuitively, the environment models the "larger world" in which the concurrent execution takes place (e.g., for a distributed computing task over the Internet, the environment models all the other activities occurring on the Internet at the same time).

Adversarial behavior. The model of execution also includes a special entity called the adversary, that represents adversarial activities that are directly aimed at the protocol execution under consideration. We consider an *adaptive* adversary, who may corrupt any party at any point during the executions, and as a function of what he sees. When a party is corrupted, it shares all its tapes with the adversary and follows the instructions from the adversary for all its future actions.

While honest parties only communicate with the environment through the input/output of the functions they compute, the adversary is also able to exchange messages with the environment in an arbitrary way through out the computation[6]. Furthermore, the adversary controls the scheduling of the delivery of all messages exchanged between parties (messages sent by the environment is delivered directly). Technically, this is modeled by letting the adversary read the outgoing message tapes of all parties and decide whether or not and when (if at all) to deliver the message to the recipient, therefore the communication is asynchronous and lossy. However, the adversary cannot insert messages and claim arbitrary sender identity. In other words, the communication is authenticated.

[6]Through its interaction with the environment, the adversary is also able to influence the inputs to honest parties indirectly.

Protocol execution. The *execution of a protocol π with the environment Z, adversary A and trusted party $\mathcal{G}$* proceeds as follows. The environment is the first entity activated in the execution, who then activates the adversary, and invokes other honest parties. At the time an honest party is invoked, the environment assigns it a unique identifier, and inquiries the adversary whether it wants to corrupt the party or not. To start an execution of the protocol π, the environment initiates a *protocol execution session*, identified by a session identifier sid, and activates all the participants in that session. An honest party activated starts executing the protocol π thereafter and has access to the trusted party $\mathcal{G}$. We remark that in the UC model, the environment only initiates one protocol execution session.

Invoking parties. The environment invokes an honest party by passing input (invoke, P_i) to it. P_i is the globally unique identity for the party, and is picked dynamically by the environment at the time it is invoked. Immediately after that, the environment notifies the adversary of the invocation of P_i by sending the message (invoke, P_i) to it, who can then choose to corrupt the party by replying (corrupt, P_i). Note that here as the adversary is static, parties are corrupted only when they are "born" (invoked).

Session initiation. To start an execution of protocol π, the environment selects a subset U of parties that has been invoked so far. For each party $P_i \in U$, the environment activates P_i by sending a start-session message (start-session, P_i, sid, $c_{i,sid}$, $x_{i,sid}$) to it, where sid is a session id that identifies this execution. We remark that in the UC model, the environment starts only one session, and hence all the parties activated have the same session id.

Honest party execution. An honest party P_i, upon receiving (start-session, P_i, sid, $c_{i,sid}$, $x_{i,sid}$), starts executing its code $c_{i,sid}$ input $x_{i,sid}$. During the execution,

- the environment can read P_i's output tape and at any time may pass additional inputs to P_i;
- according to its code, P_i can send messages (delivered by the adversary) to other parties in the session, in the format (P_i, P_j, s, content)[7], where P_j is the identity of the receiver;
- according to its code, P_i can send input to the trusted party in the format (P_i, $\mathcal{F}$, s, input).

Adversary execution. After activation, the adversary may perform one of the following activities at any time during the execution.

- The adversary can read the outgoing communication tapes of all honest parties and decides to deliver some of the messages.
- A can exchange arbitrary messages with the environment.
- The adversary can read the inputs, outputs, incoming messages of a corrupted party, and instruct the corrupted party for any action.
- The adversary can decide to corrupt any party from the set of honest parties at the moment.

Output. The environment outputs a final result for the whole execution in the end.

In the execution of protocol π with security parameter $n \in N$, environment Z, adversary A and trusted party $\mathcal{G}$, we define $\mathsf{Exec}^{\mathcal{G}}_{\pi,\mathsf{A},\mathsf{Z}}(\mathsf{n})$ to be the random variable describing the output of the environment Z, resulting from the execution of the above procedure.

Let $\mathcal{F}$ be an ideal functionality; we denote by π_{ideal} the protocol accessing $\mathcal{F}$, called as the ideal protocol. In π_{ideal} parties simply interacts with $\mathcal{F}$with their private inputs, and receives their corresponding outputs from the functionality at the end of the computation. Then the ideal model execution of the functionality

[7]The session id in the messages enables the receiver to correctly de-multiplexing a message to its corresponding session, even though the receiver may involve in multiple sessions simultaneously.

$\mathcal{F}$is just the execution of the ideal protocol π_{ideal} with environment Z, adversary A' and trusted party $\mathcal{F}$. The output of the execution is thus $\mathsf{Exec}^{\mathcal{F}}_{\pi_{\mathsf{ideal}},\mathsf{A}',\mathsf{Z}}(\mathsf{n})$. On the other hand, the real model execution does not require the aid of any trusted party. Let π be a multi-party protocol implementing $\mathcal{F}$. Then, the real model execution of π is the execution of π with security parameter n, environment Z and adversary A, whose output is the random variable $\mathsf{Exec}_{\pi,\mathsf{A},\mathsf{Z}}(\mathsf{n})$. Additionally, the $\mathcal{G}$-Hybrid model execution of a protocol π is the execution of π with security parameter n, environment Z and adversary A and ideal functionality $\mathcal{G}$.

Security as emulation of a real model execution in the ideal model. Loosely speaking, a protocol securely realizes an ideal functionality if it securely emulates the ideal protocol π_{ideal}. This is formulated by saying that for every adversary A in the real model, there exists an adversary A' (a.k.a. *simulator*) in the ideal model, such that no environment Z can tell apart if it is interacting with A and parties running the protocol, or A' and parties running the ideal protocol π_{ideal}.

Definition 7. (Adaptive UC security) *Let $\mathcal{F}$ and π_{ideal} be defined as above, π be a multi-party protocol in the $\mathcal{G}$-hybrid model. The protocol π is said to realize $\mathcal{F}$ with adaptive UC security in $\mathcal{G}$-hybrid model, if for every uniform $\mathcal{PPT}$ adaptive adversary A, there exists a uniform $\mathcal{PPT}$ simulator A', such that, for every non-uniform $\mathcal{PPT}$ environment Z, the following two ensembles are indistinguishable.*

$$\left\{\mathsf{Exec}^{\mathcal{G}}_{\pi,\mathsf{A},\mathsf{Z}}(\mathsf{n})\right\}_{\mathsf{n}\in\mathsf{N}} \approx \left\{\mathsf{Exec}^{\mathcal{F}}_{\pi_{\mathsf{ideal}},\mathsf{A}',\mathsf{Z}}(\mathsf{n})\right\}_{\mathsf{n}\in\mathsf{N}}$$

Multi-session extension of ideal functionalities Note that the UC model only considers a single session of the protocol execution. (The environment is only allowed to open one session). To consider multiple concurrent executions, we focus on the multi-session extension of ideal functionalities [6, 11]. More specifically, let $\hat{\mathcal{F}}$ be the multi-session extension of $\mathcal{F}$. $\hat{\mathcal{F}}$ runs multiple copies of $\mathcal{F}$, where each copy will be identified by a special "sub-session identifier". Every k parties, trying access $\mathcal{F}$ together, share a sub-session identifier, $ssid$. To compute, each party simply sends its private input together with $ssid$ to $\hat{\mathcal{F}}$. $\hat{\mathcal{F}}$ upon receiving all the inputs, activates the appropriate copy of $\mathcal{F}$ identified by $ssid$ (running within $\hat{\mathcal{F}}$), and forwards the incoming messages to that copy. (If no such copy of $\mathcal{F}$exists then a new copy is invoked and is given that ssid.) Outputs generated by the copies of $\mathcal{F}$are returned to corresponding parties by $\hat{\mathcal{F}}$.

A.4 A Generalized Version of UC

In the UC model, the environment is modeled as a non-uniform $\mathcal{PPT}$ machine and the ideal-model adversary (or simulator) as a (uniform) $\mathcal{PPT}$ machines. We consider a generalized version (in analogy with [42, 48]) where we allow them to be in arbitrary complexity classes. Note, however, that the adversary is still $\mathcal{PPT}$. Additionally, we "strengthen" the definition by allowing the environment to output a bit string (instead of a single bit) at the end of an execution. In the traditional UC definition, it is w.l.o.g. enough for the environment to output a single bit [6]; in our generalized version this no longer holds and we are thus forced to directly consider the more stringent version.

We represent a generalized UC model by a 2-tuple $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$, where $\mathcal{C}_{\mathsf{env}}$ and $\mathcal{C}_{\mathsf{sim}}$ are respectively the classes of machines the environment and the simulator of the general model belong to. We consider only classes, $\mathcal{C}_{\mathsf{env}}$ and $\mathcal{C}_{\mathsf{sim}}$, that are closed under probabilistic polynomial time computation. For a model $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$, let $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$ denote the complexity class that includes all computations by $\mathcal{PPT}$ oracle Turing machines M with oracle access to $\mathcal{C}_{\mathsf{env}}$ and $\mathcal{C}_{\mathsf{sim}}$.

Definition 8 ($(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-Adaptive UC adaptive security)**.** *Let $\mathcal{F}$ and π_{ideal} be, as defined above, and π be a multi-party protocol. The protocol π is said to realize $\mathcal{F}$ with $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-adaptive UC security, if for every $\mathcal{PPT}$ machine A, there exists a machine $A' \in \mathcal{C}_{\mathsf{sim}}$, such that, for every $Z \in \mathcal{C}_{\mathsf{env}}$, the following two ensembles are indistinguishable w.r.t $\mathcal{C}_{\mathsf{sim}}$.*

$$\left\{\mathsf{Exec}_{\pi,\mathsf{A},\mathsf{Z}}(\mathsf{n})\right\}_{\mathsf{n}\in\mathsf{N}} \approx \left\{\mathsf{Exec}^{\mathcal{F}}_{\pi_{\mathsf{ideal}},\mathsf{A}',\mathsf{Z}}(\mathsf{n})\right\}_{\mathsf{n}\in\mathsf{N}}$$

Using the above notation, traditional UC is equivalent to (n.u.$\mathcal{PPT}$,$\mathcal{PPT}$)-UC-security. We let QPS-UC denote (n.u.$\mathcal{PPT}$,$\mathcal{PQT}$)-UC-security[8] (where $\mathcal{PQT}$ denotes probabilistic quasi-polynomial time algorithms), and Non-uniform UC denote ($\mathcal{PPT}$,n.u.$\mathcal{PPT}$)-UC-security.

B 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.

Formally, let $n \in N$ be a security parameter and $\langle \mathsf{S}, \mathsf{R} \rangle$ be a protocol between two parties, the sender S and the receiver R. We consider a **concurrent puzzle execution** for an adversary $\mathcal{A}$. In a **concurrent puzzle execution**, $\mathcal{A}$ exchanges messages with a puzzle-environment $\mathcal{Z} \in \mathcal{C}_{\mathsf{env}}$ and participates as a sender concurrently in $m = \mathrm{poly}(n)$ puzzles with honest receivers $\mathsf{R}_1, \ldots, \mathsf{R}_m$. At the onset of a concurrent execution, $\mathcal{Z}$ outputs a session identifier sid that all receivers in the concurrent puzzle execution receive as input. Thereafter, the puzzle-environment is allowed to exchange messages only with the adversary $\mathcal{A}$. We compare a *real* and an *ideal* execution.

Real execution. In the real execution, the adversary $\mathcal{A}$ on input 1^n, interacts with a puzzle-environment $\mathcal{Z} \in \mathcal{C}_{\mathsf{env}}$ and participates as a sender in m interactions using $\langle \mathsf{S}, \mathsf{R} \rangle$ with honest receivers that receive input sid (decided by $\mathcal{Z}$). The adversary $\mathcal{A}$ is allowed to exchange arbitrary messages with environment $\mathcal{Z}$ when participating in puzzle interactions with the receivers as a sender. In addition $\mathcal{A}$ may adaptively corrupt any of the receivers $\mathsf{R}_1, \ldots, \mathsf{R}_m$ at any point during or after the execution. We assume without loss of generality that, after every puzzle-interaction, $\mathcal{A}$ honestly sends TRANS to $\mathcal{Z}$, where TRANS is the puzzle-transcript. Finally, $\mathcal{Z}$ outputs a string in $\{0,1\}^*$. We denote this by $\mathsf{REAL}_{\mathcal{A},\mathcal{Z}}(n)$.
Ideal execution. Consider $\mathcal{A}' \in \mathcal{C}_{\mathsf{sim}}$ in the ideal-model that has a special output-tape (not accessible by $\mathcal{Z}$). In the ideal execution, $\mathcal{A}'$ on input 1^n interacts with puzzle-environment $\mathcal{Z}$. We denote the output of $\mathcal{Z}$ at the end of the execution by $\mathsf{IDEAL}_{\mathcal{A}',\mathcal{Z}}(n)$.

Definition 9. *Adaptive UC-Puzzle. A pair $(\langle \mathsf{S}, \mathsf{R} \rangle, \mathcal{R})$ is a $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure Adaptive UC-puzzle for a polynomial time computable relation $\mathcal{R}$ and model $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$, if the following conditions hold.*

Soundness: *For every malicious PPT receiver $\mathcal{A}$, there exists a negligible function $\epsilon(\cdot)$ such that the probability that $\mathcal{A}$, after an execution with S on common input 1^n, outputs y such that $y \in \mathcal{R}(\mathsf{TRANS})$ where TRANS is the transcript of the messages exchanged in the interaction, is at most $\epsilon()$.*

Statistical Simulatability: *For every adversary $\mathcal{A} \in \mathcal{C}_{\mathsf{env}}$ participating in a **concurrent puzzle execution**, there is a simulator $\mathcal{A}' \in \mathcal{C}_{\mathsf{sim}}$ such that for all puzzle-environments $\mathcal{Z} \in \mathcal{C}_{\mathsf{env}}$, the ensembles $\{\mathsf{REAL}_{\mathcal{A},\mathcal{Z}}(n)\}_{n\in\mathbb{N}}$ and $\{\mathsf{IDEAL}_{\mathcal{A}',\mathcal{Z}}(n)\}_{n\in\mathbb{N}}$ are statistically close over $n \in \mathbb{N}$ and when-*

[8] We mentioned that this is stronger than the notion of QPS security of [42, 48, 3] which only consider indistinguishability w.r.t $\mathcal{PPT}$; we, in analogy with the notion of *strong QPS* of [42] require indistinguishability to hold also w.r.t $\mathcal{PQT}$.

ever $\mathcal{A}'$ sends a message of the form TRANS *to $\mathcal{Z}$, it outputs y in its special output tape such that $y \in \mathcal{R}(\mathsf{TRANS})$.*

The analysis of our equivocal non-malleable commitment scheme will additionally require the puzzle to be **straight-line simulatable**. In fact, for 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 (when there are no communication restrictions on the adversary, such as the timing or partially-isolated adversaries model).[9]

C The Equivocal Non-Malleable Commitment Scheme (EQNMCom)

We note that the construction presented here is the same as the construction of [21, 34] with the following changes: the statistically-binding commitment is replaced with an equivocal commitment and the special-sound WI proofs are replaced with adaptively-secure WIPOK's. Although the constructions are similar, the analysis here differs significantly from the analysis of the previous constructions of [21, 34] where the fact that the first commitment is statistically-binding plays a large part in the proof.

The protocol $\Pi = \langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ proceeds in the following two stages on common input the identity $\mathsf{id} \in \{0,1\}^\ell$ of the committer, common string x, and security parameter n.

1. In Stage 1, the Committer sends $c = \mathsf{EQCom}^x(v)$, where EQCom is a language-based equivocal commitment scheme as in Definition 2 with common input x.

2. In Stage 2, the Committer proves that c is a valid commitment for v. This is proved by 4ℓ invocations of an adaptively-secure (without erasures) WIPOK (See Appendix G) where the messages are scheduled based on the id (as in [21, 34]). 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 3).

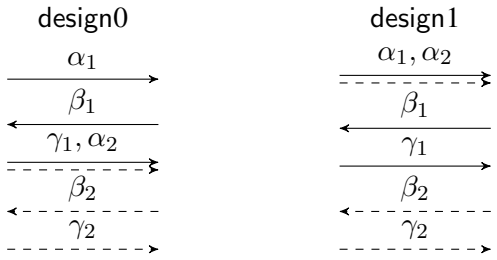

Figure 3: Message schedule in a round in adaptively-secure WIPOK

C.1 Analysis

In this subsection, we prove that $\Pi = \langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ is an equivocal non-malleable commitment scheme when combined with an adaptive UC-puzzle in a preamble phase where the receiver acts the sender and the committer acts as the receiver and the NP-statement x used in $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ is the transcript of the interaction from the preamble phase. More precisely, consider the following protocol: Let $(\langle \mathsf{S}, \mathsf{R} \rangle, \mathcal{R})$ be a $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure adaptive UC-puzzle. The protocol $\overline{\Pi}$ proceeds in the following two phases on common input the identity $\mathsf{id} \in \{0,1\}^\ell$ of the committer, and private-input string r for the committer and security parameter n.

[9] The basic idea in [26], is that the adversary can corrupt a random subset of (dummy) parties between any two messages thereby requiring a "rewinding" simulator to produce their inputs that the simulator does not *a priori* know.

Commitment Protocol $\Pi = \langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$

Common input: An identifier $\mathsf{id} \in \{0,1\}^{\ell}$ and common input x.

Auxiliary Input for Committer: A string $v \in \{0,1\}^{n}$.

Stage 1:

S uniformly chooses $r \in \{0,1\}^{\mathrm{poly}(n)}$.

$S \rightarrow R$: $c = \mathsf{EQCom}^{x}(v; r)$.

Stage 2:

$S \rightarrow R$: 4ℓ adaptively-secure WIPOK of the statement there exist values v, r such that $c = \mathsf{EQCom}^{x}(v; r)$ with verifier query of length $2n$, in the following schedule: For $j = 1$ to ℓ do: Execute $\mathsf{design}_{\mathsf{id}_j}$ followed by Execute $\mathsf{design}_{1-\mathsf{id}_j}$.

Figure 4: Equivocal Non-Malleable Commitment Scheme $\Pi = \langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$

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

Commitment Phase: The parties run protocol $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ with common input x and identifier id. S plays the part of sender with input r.

We now show that the protocol $\overline{\Pi}$ is concurrent non-malleable w.r.t opening in the $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-model.

THEOREM 2. *Commitment scheme $\overline{\Pi}$ described above is concurrent non-malleable w.r.t. opening with independent and identically distributed (i.i.d) commitments*

Before we prove this theorem, we first show that Π is a language-based equivocal commitment scheme:

Lemma 1. *Commitment scheme $\Pi = \langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ shown in Figure 4 is a language-based equivocal commitment scheme.*

Proof. In order to prove the lemma we need to present an equivocator $(\tilde{\mathsf{S}}_{\mathsf{com}}, \mathsf{Adap}_{\mathsf{com}})$ for $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ and prove that $(\tilde{\mathsf{S}}_{\mathsf{com}}, \mathsf{Adap}_{\mathsf{com}})$ has the required properties listed in Definition 2. Intuitively the equivocator, $\tilde{\mathsf{S}}_{\mathsf{com}}$, will run the equivocator for the commitment scheme EQCom as well as the simulator for the WIPOK. Then, $\mathsf{Adap}_{\mathsf{com}}$ will run $\mathsf{Adap}_{\mathsf{eq}}$ for the EQCom scheme and also will adaptively corrupt the prover and run the simulator for the WIPOK, which produces a simulated view for the prover. By taking a closer look at the simulator for the WIPOK presented in Appendix G we see that, in fact, $\tilde{\mathsf{S}}_{\mathsf{com}}$ simply replaces *every* commitment under EQCom in $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ (in both Stage 1 and Stage 2) with an equivocal commitment generated by the equivocator, $\tilde{\mathsf{S}}_{\mathsf{eq}}$ for the commitment scheme EQCom. The fact that $\tilde{\mathsf{S}}_{\mathsf{com}}$ has this form will be crucial for the proof of Lemma 3.

We omit the proof that $(\tilde{\mathsf{S}}_{\mathsf{com}}, \mathsf{Adap}_{\mathsf{com}})$ as described above has the desired properties since it follows straightforwardly from the security properties of EQCom and the adaptive security (without erasures) of the WIPOK. □

Now, we turn towards proving Theorem 2.

Proof of Theorem 2: First we describe the simulator and then prove correctness. Let A be a concurrent man-in-the-middle adversary that on input 1^n participates in at most $m(n)$ left-interactions as a receiver, receiving commitments from an honest committer whose values are chosen uniformly from distribution D and at most $m(n)$ right-interactions as a committer.

As mentioned before, we assume that the puzzle-simulation is **straight-line**. On a high-level, S internally incorporates A and emulates an execution with A as follows:

1. For all puzzle interactions where A^* controls the sender, S follows the puzzle simulator's strategy to simulate the puzzle and obtains a witness which it stores.

2. For all the messages exchanged by A^* in the right interactions, Sim simply forwards the messages to an external receiver.

3. For every left interaction, Sim internally generates the messages using the code of special committer (guaranteed by the scheme), i.e. equivocate in the commitment phase with the witness w obtained from the puzzle interactions. When a decommitment is requested by A, Sim outputs the current partial view of the transcript of messages exchanged by A in a special-output tape. Then, it receives a value v from outside to be decommitted to in the left interaction. Internally, it runs the Adap algorithm guaranteed by the equivocal commitment scheme to generate a decommitment to v and feeds it to A.

4. Whenever the commitment phase with a receiver is completed on the right, Sim temporary stalls the main-execution and tries to extract the value committed to by A in this interaction. For this, Sim selects a random WIPOK from that interaction and *rewinds* A to the point just before which A receives the challenge-message in the WIPOK. Sim supplies a new challenge message and continues simulation. In this simulation, the right interactions are simulated as before (i.e. honestly). However the left interactions are not simulated as before (i.e. equivocating the commitment phase). Instead they are generated using an honest committer committing to a value v, where v is either the decommitment for that left interaction, if one has been obtained by Sim in the main-execution, or a uniformly chosen sample from D.[10] If in the rewinding, A provides a valid response for the selected WIPOK of the right interaction, then using the special-sound property of the WIPOK, Sim extracts the witness used in the WIPOK, which contains the committed value. If the adversary fails to provide a valid response for the particular WIPOK in the right interaction, Sim cancels the current rewinding and starts a new rewinding by supplying a new challenge.

The proof of correctness of the simulator is expressed in the following lemma.

Lemma 2. *The following ensembles are computationally indistinguishable*

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

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

Proof of Lemma 2. We consider a sequence of intermediate hybrid experiments $H_0, \ldots, H_m$. In experiment H_i, we consider a simulator Sim^i that knows the values $(v_1, \ldots, v_i)$ being committed to in the first i left interactions. On input z, Sim^i proceeds as follows: It proceeds exactly as Sim with the exception that only the first i left-interactions are equivocated while the rest are simulated using the honest committer algorithm, committing to values $(v_{i+1}, \ldots)$ both in the main-execution as well as in the rewinding. Let $\mathsf{hyb}^i_A(1^n, v_1, \ldots, v_m, z)$ denote the output of Sim^i in H_i. It follows from description that

$$\mathsf{hyb}^m_A(1^n, v_1, \ldots, v_m, z) \;=\; \mathsf{sim}^S_{\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle}(1^n, v_1, \ldots, v_m, z)$$

[10] Sim can generate such messages for any value v, since by adaptive security, Sim can obtain random coins for an honest committer and any value v that is consistent with any partial transcript generated by the equivocator.

The proof of the Lemma follows from the next two claims using a standard hybrid argument.

Claim 1. *The following ensembles are computationally indistinguishable*

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

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{hyb}^0_A(1^n, v_1, \ldots, v_m, z)\right\}_{n \in N, z \in \{0,1\}^*}$$

Proof of Claim 1. Recall that in hybrid experiment H_0, the simulator simulates all the committers in the left interaction using the honest committer algorithm. The only difference from the MIM experiment is that the puzzles are simulated. Assume for contradiction, there exists an adversary A, distinguisher D, polynomial $p(\cdot)$ such that, for infinitely many n, D distinguishes the ensembles in the claim with probability at least $\frac{1}{p(n)}$. From the definition of the puzzle we have that the distribution of the views in the outputs of $\mathsf{MIM}^A_{\langle C,R \rangle}(v_1, \ldots, v_m, z)$ and $\mathsf{hyb}^0_A(1^n, v_1, \ldots, v_m, z)$ are statistically-close. Furthermore, if the value extracted by the simulator in $\mathsf{hyb}^0_A(1^n, v_1, \ldots, v_m, z)$ in each interaction is consistent with the decommittment made by the adversary in the view output by the simulator, then $\mathsf{MIM}^A_{\langle C,R \rangle}(v_1, \ldots, v_m, z)$ and $\mathsf{hyb}^0_A(1^n, v_1, \ldots, v_m, z)$ are statistically-close. Hence, if D distinguishes the distributions, it must be the case that, the values output in both the experiments differs with probability at least $\frac{1}{p(n)}$. This happens, whenever the value output by the simulator in hyb^0_A is inconsistent with the view output by the simulator. Hence, the Sim^0 obtains two decommitments for the same commitment (one as part of the main-execution and one obtained using the witness extracted) for a commitment made by the adversary in some right-interaction with probability at least $\frac{1}{p(n)}$. With any two valid decommitments, a solution to the puzzle from the preamble phase can be obtained. Consider a slightly altered simulation $\overline{\mathsf{Sim}}^0$ that proceeds exactly like Sim^0 with the exception that all the puzzle interactions in the left interaction are simulated honestly. It follows from the statistical-simulatability of the puzzle that with non-negligible probability, $\overline{\mathsf{Sim}}^0$ extracts a witness for a puzzle in a right interaction where the adversary is a receiver of the puzzle. Hence, if $\overline{\mathsf{Sim}}^0$ runs in $\mathcal{PPT}$, then, $\overline{\mathsf{Sim}}^0$ with adversary A can be used to construct an adversary that violates the soundness of the adaptive UC-puzzle.[11] It only remains to argue that $\overline{\mathsf{Sim}}^0$ run in $\mathcal{PPT}$ and then we arrive at a contradiction. Recall that for every right interaction that completes the commitment phase, Sim^0, and hence $\overline{\mathsf{Sim}}^0$ rewinds repeatedly until it obtains a witness for a random WIPOK. We argue that the expected number of restarts for every right- interaction is $O(1)$ and therefore the expected running time of $\overline{\mathsf{Sim}}^0$ is bounded by some polynomial. Fix a particular right-interaction that completes the commitment phase and select a WIPOK. Given the first message of the WIPOK, let p be the probability that over a random challenge-message that A provides a valid response. Since the rewindings are identically distributed to the main-execution, the expected number of restarts required before $\overline{\mathsf{Sim}}^0$ encounters another execution where A provides a valid response is $\frac{1}{p}$. However, note that $\overline{\mathsf{Sim}}^0$ needs to perform the rewinding only with probability p since otherwise the right-interaction does not complete the commitment phase. Therefore, the expected number of restarts for a particular right interaction is $p \times \frac{1}{p} = 1$. We remark here that, proving the running time of the actual simulator is bounded, essentially follows from the indistinguishability of the hybrids. □

Claim 2. *The following ensembles are computationally indistinguishable*

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{hyb}^0_A(1^n, v_1, \ldots, v_m, z)\right\}_{n \in N, z \in \{0,1\}^*}$$

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{hyb}^m_A(1^n, v_1, \ldots, v_m, z)\right\}_{n \in N, z \in \{0,1\}^*}$$

[11] Simply forward a random puzzle interaction of A during the straight-line simulation to an external sender of a puzzle execution and then internally obtain two decommitments of A and extract a witness whenever A equivocates

Proof of Claim 2. Assume for contradiction, there exists an adversary A, distinguisher D, polynomial $p(\cdot)$ such that, for inifinitely many n, D distinguishes the ensembles in the claim with probability at least $\frac{1}{p(n)}$. Then there exists a function $i : \mathcal{N} \to \mathcal{N}$ such that for infinitely many n, D distinguishes the following two ensembles with probability at least $\frac{1}{mp(n)}$.

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{hyb}_A^{i(n)-1}(v_1, \ldots, v_m, z)\right\}_{n\in N, z\in\{0,1\}^*}$$

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{hyb}_A^{i(n)}(1^n, v_1, \ldots, v_m, z)\right\}_{n\in N, z\in\{0,1\}^*}$$

Let H'_j denote the experiment that proceeds identically to H_j, with the exception that the simulator performs no rewinding. Let hyb'^j_A denote the random variable that represents the view output by the simulator in H'_j. It follows from description that hyb'^j_A is identically distributed to the view in hyb^j_A since the rewindings are conducted independent of the main-execution.

We first claim that the following ensembles are indistinguishable for any function $j : \mathcal{N} \to \mathcal{N}$.

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{hyb}'^{j(n)-1}_A(v_1, \ldots, v_m, z)\right\}_{n\in N, z\in\{0,1\}^*}$$

$$\left\{(v_1 \ldots, v_m) \leftarrow D^n : \mathsf{hyb}'^{j(n)}_A(1^n, v_1, \ldots, v_m, z)\right\}_{n\in N, z\in\{0,1\}^*}$$

This is because the only difference between $H'_{j(n)-1}$ and $H'_{j(n)}$ is that the $j(n)^{th}$ left interaction is equivocated and therefore indistinguishability directly follows from the strong-hiding property of the equivocal commitment.

Recall that if the values extracted by the simulator is always equal to the value decommitted to by the adversary, then the above claim implies that $\mathsf{hyb}_A^{i(n)-1}$ and $\mathsf{hyb}_A^{i(n)}$ are indistinguishable. Therefore, it must be the case that, for infinitely many n, with probability at least $\frac{1}{2mp(n)}$ the value extracted by the simulator is different from the value decommitted to by A. Furthermore, there exists a function $k : \mathcal{N} \to \mathcal{N}$ such that, for infinitely many n, the value extracted by the simulator in the $k(n)^{th}$ right interaction is different from the value decommited to by A in the main-execution with probability at least $\frac{1}{2m^2p(n)}$. Let $\mathsf{hyb}_A^{i,k}$ denote the view output of the simulator in H_i and the value extracted in the k^{th} right interaction. Then there exists a function $k(n)$ such that the probability with which the value output is not the value decommitted in the view jumps by at least $\frac{1}{2m^2p(n)}$ when comparing $\mathsf{hyb}_A^{i(n)-1,k(n)}$ and $\mathsf{hyb}_A^{i(n),k(n)}$ with probability at least $\frac{1}{2m^2p(n)}$ for infinitely many n. Lets say that a $(view, v)$ pair is k- cons if v is the value decommited to by the adversary in k^{th} right-interaction.

We consider the following intermediate hybrid experiments:

Hybrid $\bar{H}_0^k = H_{i-1}$: This experiment proceeds identically to H_{i-1} with the exception that the simulator only extracts the decommitment from the k^{th} right interaction. Define $\overline{\mathsf{hyb}}_A^0$ to be the view output and the value extracted by the simulator, i.e. $\overline{\mathsf{hyb}}_A^0 = \mathsf{hyb}_A^{i-1,k}$.

Hybrid $\bar{H}_1^k$: In the k^{th} right-interaction, we say that a particular WIPOK is a safe WIPOK, if the "safe-point" of this interaction w.r.t i^{th} left interaction corresponds to this WIPOK. The definition of safe-point is analogous and identical to the safe-points defined in [34].[12]The experiment $\bar{H}_1^k$ proceeds identically to $\bar{H}_0^k$ with the exception that it rewinds the adversary to a safe WIPOK in the k^{th} right interaction instead of a choosing a random WIPOK and the i^{th} left interaction. Define $\overline{\mathsf{hyb}}_A^1$ to be the view output and the value extracted by the simulator.

[12]Intuitively, a safe-point ρ of a right interaction, is a point in Δ that lies in between the first two messages α_r and β_r of a WIPOK proof $(\alpha_r, \beta_r, \gamma_r)$ in the right interaction k, such that, when rewinding from ρ to γ_r, if A uses the *same* "scheduling of messages" as in Δ, then the left interaction can be emulated without affecting the hiding property. See [34] for more details.

Hybrid $\bar{H}_2^k$: This experiment proceeds identically to $\bar{H}_1^k$ with the exception that the i^{th} left interaction is simulated using *fresh randomness* in each rewinding. In particular, if the next message in the i^{th} left interaction is the first message of a WIPOK sub-protocol, then fresh randomness is used to generate it.[13] Recall that, in the actual simulator and previous hybrids, this is not the case and in the rewinding phase, the randomness of the all the left interactions are fixed. Furthermore, whenever the adversary tries to corrupt the i^{th} left interaction in a rewinding the simulator cuts off the rewinding and restarts. Define $\overline{\mathsf{hyb}}_A^2$ to be the view output and the value extracted by the simulator.

Hybrid $\bar{H}_3^k$: This experiment proceeds identically to $\overline{\mathsf{hyb}}_A^2$ with the exception that in the i^{th} left interaction, the simulator equivocates the commitment both in the main-execution as well as in the rewindings. Again, as in the previous hybrid, a fixed random tape is used for all the left-interactions in the rewindings except the i^{th} interaction where fresh randomness is used in the rewindings. Every rewinding where the adversary tries to corrupt the committer in the i^{th} left-interactions is cancelled. Define $\overline{\mathsf{hyb}}_A^3$ to be the view output and the value extracted by the simulator.

Hybrid $\bar{H}_4^k$: The experiment proceeds identically to $\overline{\mathsf{hyb}}_A^3$ with the exception that the i^{th} left interaction is also simulated using a fixed random tape for the committer in all the rewindings. Define $\overline{\mathsf{hyb}}_A^4$ to be the view output and the value extracted by the simulator.

Hybrid $\bar{H}_5^k = H_i$: The experiment proceeds identically to H_i with the exception that the simulator only extracts from the k^{th} right interaction . Define $\overline{\mathsf{hyb}}_A^5$ to be the view output and the value extracted by the simulator, i.e. $\overline{\mathsf{hyb}}_A^5 = \mathsf{hyb}_A^{i,k}$.

Since, the difference in probability that $\mathsf{hyb}_A^{i(n)-1,k(n)}$ and $\mathsf{hyb}_A^{i(n),k(n)}$ are k- cons is at least $\frac{1}{p_1(n)} = \frac{1}{2m^2p(n)}$ for infinitely many n, there must exists a $c \in \{1,2,3,4,5\}$ such that the difference in probability that $\overline{\mathsf{hyb}}_A^{c-1}$ and $\overline{\mathsf{hyb}}_A^{c}$ are k- cons is at least $\frac{1}{5p_1(n)}$ for infinitely many n. We argue below for every c that we arrive at a contradiction if the above statement holds for c.

Comparing $\bar{H}_0^k$ and $\bar{H}_1^k$ In this case, we have that, for infinitely many n,

$$|\Pr[\overline{\mathsf{hyb}}_A^0 \text{ is } k\text{-cons }] - \Pr[\overline{\mathsf{hyb}}_A^1 \text{ is } k\text{-cons }]| \geq \frac{1}{5p_1(n)}$$

Proof Sketch: Since the only difference between $\bar{H}_0^k$ and $\bar{H}_1^k$ is the WIPOK chosen to rewind and extract the witness, it must be the case that the probability that the witness extracted from a random WIPOK and the specific WIPOK chosen from the safe point is different must be at least $\frac{1}{5p_1(n)}$. Using this fact, we arrive at a contradiction to the soundness of the puzzle.

First, we note that it is possible for a simulator to check if the value extracted in a random WIPOK and a safe WIPOK are the same. Recall that in the left interactions of the main execution in $\bar{H}_1^k$ and $\bar{H}_2^k$, the simulator is equivocating the first i commitments and honestly committing in the rest of them. This in particular means that the value decommitted to in the first i commitments are chosen after the commitment phase. In the rewinding phase, when the simulator tries to extract the witness from a WIPOK, it simulates the left interactions by using the honest committer strategy with a fixed random tape. Consider an experiment E_0^k where the simulator continues the execution until A completes the commitment phase in the kth right-interaction and then cuts off the simulation. Then it extracts the witness from a random WIPOK and the safe WIPOK. If the values are different, the simulator extracts the solution of the puzzle and outputs it. It follows that the simulator outputs the solution of the puzzle with non-negligible probability.

[13] Jumping ahead, this will allow the i^{th} left-interaction to be forwarded externally to a committer, analogous to [21, 35].

We consider of hybrid experiments and show that in each of them the simulator can output a solution with non-negligible probability and finally arrive at a simulator that violates the soundness of the puzzle.

The first intermediate experiment E_1^k we consider is where the simulator chooses the value to be committed in the first k left interactions before the interaction begins. This modification does not affect the view obtained in the main-execution because all values in the left interactions are chosen independently from distribution D. It also does not affect the rewindings, because the committers strategy is fixed, i.e. its random tape and commitment are fixed. Therefore, E_1^k and E_0^k proceed identically and the simulator outputs the solution to the puzzle with non-negligible probability in E_1^k as well.

In Lemma 3, we show that, it is possible to construct an honest committers algorithm C^* for $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$ that knows the witness of the common input statement x and receives a polynomial sequence of strings $s_1, \ldots, s_m$ such that

- The transcript generated by C^* committing to string v, when the strings received as input $s_1, \ldots, s_m$ are uniformly random, is identically distributed to the transcript of an interaction with an honest committer, committing to a value v, and,
- The transcript generated by C^* committing to string v, when the strings received as input $s_1, \ldots, s_m$ are random commitments to 1 using Com, is identically distributed to the transcript generated by an equivocal commitment using the witness for statement x and decommitted to value v.

If the value to be decommitted to is known at the beginning of an execution, then the commitment phase can be generated using an honest-committer's strategy that additionally receives as input a particular sequence of strings that are either uniformly random or commitments to 0 and 1 under $\langle S, R \rangle$. We now observe that in E_1^k, although the simulator is equivocating the first i commitments in the left, the value to be decommitted to is chosen before the execution begins. Consider the experiment E_2^k that proceeds identically to E_1^k with the exception that the simulator generates the equivocal commitments by using the committer strategy C^* that receives as input a sequence of commitments to 1 using Com. This experiment proceeds identically to E_1^k and the simulator extracts the solution to the puzzle with non-negligible probability.

In the next experiment E_3^k, we consider a simulator that proceeds identically to E_2^k with the exception that the sequence of strings received by the honest-committers in the first k left interactions are chosen uniformly at random (as opposed to commitments to 1). It now follows from the pseudo-randomness of the commitments under Com that the simulator extracts the solution to the puzzle in E_3^k with non-negligible probability. Now, observe that experiment E_3^k is identical to experiment H_0 and this violates the soundness of the puzzle interaction. Thus, we arrive at a contradiction. □

Comparing $\bar{H}_1^k$ and $\bar{H}_2^k$ In this case, we have that, for infinitely many n,

$$|\Pr[\overline{\mathsf{hyb}}_A^1 \text{ is } k\text{-cons }] - \Pr[\overline{\mathsf{hyb}}_A^2 \text{ is } k\text{-cons }]| \geq \frac{1}{5p_1(n)}$$

Proof Sketch: We will again show how to construct an adversary that violates the soundness of the puzzle. The proof of this follows identically as in the previous case. We again consider a simulator that cuts off the adversary after the commitment phase on the k^{th} right interaction is completed and then rewinds to extract a witness from the safe WIPOK in two different ways, i.e. as in $\bar{H}_1^k$ and $\bar{H}_2^k$. Again we have that the simulator extracts the solution to the puzzle with non-negligible probability as the value extracted are different with non-negligible probability. We can follow identically as in the previous hybrid, by considering the sequence of hybrid experiments, E_0^k to E_3^k where the left interactions are all honestly generated. Again we have that the simulator extracts the solution of the puzzle in E_3^k and this violates the solution of the puzzle. □

Comparing $\bar{H}_2^k$ and $\bar{H}_3^k$ In this case, we have that, for infinitely many n,

$$|\Pr[\overline{\mathsf{hyb}}_A^2 \text{ is } k\text{-cons }] - \Pr[\overline{\mathsf{hyb}}_A^3 \text{ is } k\text{-cons }]| \geq \frac{1}{5p_1(n)}$$

In this case, we will show that A can be used to violate the special-hiding property of a variant of the commitment scheme.

Proof Sketch: The idea here (that originates from the work in [21], also used in [34], is that the simulation can be carried out even when the i^{th} left interaction is forwarded externally to a committer participating in $\widetilde{\Pi}$ which is a slightly altered version of the protocol Π. The only difference of $\widetilde{\Pi}$ from Π is that $\widetilde{\Pi}$ does not have a fixed scheduling of WIPOKs in the Stage 2 based on the committers identity. Instead, the receiver can request the committer to provide proofs using WIPOK using designs of its choice. This is analogous to [34]. It was shown in [34], that while rewinding from a safe WIPOK in the k^{th} right-interaction, the messages for the i^{th} left interaction can be received from an external committer, interacting using $\widetilde{\Pi}$. Consider an experiment, where the simulator proceeds identically as in $\bar{H}_3^k$ with the exception that the i^{th} left interaction is forwarded externally to a committer following $\widetilde{\Pi}$ that commits to a value uniformly chosen from D. It now follows that this experiment proceeds identically to $\bar{H}_2^k$ if the external committer is following the honest committer strategy in $\widetilde{\Pi}$, and is identical to $\bar{H}_3^k$ if the external committer is equivocating. Therefore, it is possible to consider an adversary that distinguishes when it receives an honest commitment or a equivocal commitment using $\widetilde{\Pi}$ on the left, by simply extracting the value from the safe WIPOK. This violates the special hiding-property of the $\widetilde{\Pi}$ and thus we arrive at a contradiction. □

Comparing $\bar{H}_3^k$ and $\bar{H}_4^k$ In this case, we have that, for infinitely many n,

$$|\Pr[\overline{\mathsf{hyb}}_A^3 \text{ is } k\text{-cons }] - \Pr[\overline{\mathsf{hyb}}_A^4 \text{ is } k\text{-cons }]| \geq \frac{1}{5p_1(n)}$$

This will follow exactly as with hybrid experiment $\bar{H}_1^k$ and $\bar{H}_2^k$.

Comparing $\bar{H}_4^k$ and $\bar{H}_5^k$ In this case, we have that, for infinitely many n,

$$|\Pr[\overline{\mathsf{hyb}}_A^4 \text{ is } k\text{-cons }] - \Pr[\overline{\mathsf{hyb}}_A^5 \text{ is } k\text{-cons }]| \geq \frac{1}{5p_1(n)}$$

This will follow exactly as with hybrid experiment $\bar{H}_0^k$ and $\bar{H}_1^k$.

□

It only remains to state and prove Lemma 3. First, we need some notation. Consider the following experiments $\mathsf{expt}^{M,R^*}_{\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle}(0, x, v_1, z)$, $\mathsf{expt}^{M,R^*}_{\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle}(1, x, v_1, z)$ where probabilistic polynomial time machines R^*, M interact using the equivocal non-malleable commitment protocol $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$ (defined in Section C) with common input $x \in L$ and private input $w \in \mathcal{R}(x)$:

Experiment $\mathsf{expt}^{M,R^*}_{\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle}(b, x, v_1, z)$**:** R^* plays the part of receiver in the $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$ protocol and initiates a request for a commitment to v_1. Upon this request, a sequence of $2t(n)$ strings is chosen $(s_1^0, \ldots, s_{t(n)}^0, s_1^1, \ldots, s_{t(n)}^1)$ (for some fixed polynomial $t(\cdot)$) in the following way: If $b = 0$, then $(s_1^0, \ldots, s_{t(n)}^0, s_1^1, \ldots, s_{t(n)}^1)$ are chosen uniformly at random. If $b = 1$, $(s_1^0, \ldots, s_{t(n)}^0)$ are chosen to be random commitments to 0 and $(s_1^1, \ldots, s_{t(n)}^1)$ are chosen to be random commitments to 1. A machine running the code of M is initiated with input $(x, w, v_1, (s_1^0, \ldots, s_{t(n)}^0, s_1^1, \ldots, s_{t(n)}^1))$. M interacts with R^* and at any point after M completes the commitment, R^* can request a decommitment from M. Define the output of the experiment $\mathsf{sta}^{M,R^*}_{\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle}(b, x, v_1, z)$ to be the output of R^*.

Lemma 3. *There exists a probabilistic polynomial time machine M^* such that for every probabilistic polynomial time adversary R^*, we have that:*

$$\begin{aligned} & \{\mathsf{sta}^{M,R^*}_{\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle}(0,x,v_1,z)\}_{n\in N,x\in L\cap\{0,1\}^n,w\in\mathcal{R}(x),v_1\in\{0,1\}^n,z\in\{0,1\}^*} \\ \equiv \quad & \{\mathsf{sta}^{R^*}_{\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle}(x,v_1,z)\}_{n\in N,x\in L\cap\{0,1\}^n,w\in\mathcal{R}(x),v_1\in\{0,1\}^n,z\in\{0,1\}^*} \end{aligned}$$

AND

$$\begin{aligned} & \{\mathsf{sta}^{M,R^*}_{\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle}(1,x,v_1,z)\}_{n\in N,x\in L\cap\{0,1\}^n,w\in\mathcal{R}(x),v_1\in\{0,1\}^n,z\in\{0,1\}^*} \\ \equiv \quad & \{\mathsf{sta}^{R^*}_{\langle \tilde{\mathsf{S}}_{\mathsf{com}},R_{\mathsf{com}}\rangle}(x,w,z)\}_{n\in N,x\in L\cap\{0,1\}^n,w\in\mathcal{R}(x),v_1\in\{0,1\}^n,z\in\{0,1\}^*} \end{aligned}$$

Proof. On input $(x,w,v_1,(s^0_1,\ldots,s^0_{t(n)},s^1_1,\ldots,s^1_{t(n)}))$ M^* runs the code of the honest committer $\mathsf{S_{com}}$ in $\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle$ with the following exception: Each time $\mathsf{S_{com}}$ uses the honest sender $\mathsf{S_{eq}}$ in the $\langle \mathsf{S_{eq}},\mathsf{R_{eq}}\rangle$ protocol to construct a commitment $c=\mathsf{EQCom}(\alpha)$ to some bit α, M^* does the following: If $\alpha=0$, M^* runs the code of the honest sender $\mathsf{S_{eq}}$. If $\alpha=1$, M^* does the following:

1. M^* uses the trapdoor w to compute an adjacency matrix I that corresponds to an isomorphism of the graph $G=\Phi(x)$ as well as the corresponding adjacency matrix H for the Hamiltonian cycle in I chooses an adjacency matrix H for a random Hamiltonian cycle.
2. If $H_{k,j}=1$, then M^* sets the bit commitment at position (k,j) in $\overline{\mathsf{Com}}_{k,j}$ to be $Com(1)$.
3. If $H_{k,j}=0$ and $I_{k,j}=0$, then M^* sets the bit commitment at position (k,j) in $\overline{\mathsf{Com}}_{k,j}$ to be an element from the sequence $s^0_1,\ldots,s^0_{t(n)}$ that has not been used yet.
4. If $H_{k,j}=0$ and $I_{k,j}=1$, then M^* sets the bit commitment at position (k,j) in $\overline{\mathsf{Com}}_{k,j}$ to be an element from the sequence $s^1_1,\ldots,s^1_{t(n)}$ that has not been used yet.

To decommit, M^* runs the code of the honest $\mathsf{S_{com}}$ in $\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle$.

Recall that equivocal commitments in the $\langle \mathsf{S_{eq}},\mathsf{R_{eq}}\rangle$ scheme are identically distributed to honest commitments to 0 and that the equivocator $(\tilde{\mathsf{S}}_{\mathsf{com}},\mathsf{Adap}_{\mathsf{com}})$ described in Lemma 1 simply replaces *every* equivocal commitment under EQCom (in both Stage 1 and Stage 2) with an equivocal commitment generated by the equivocator. Therefore, it is clear from inspection that if the $2t(n)$ strings $(s^0_1,\ldots,s^0_{t(n)},s^1_1,\ldots,s^1_{t(n)})$ are chosen uniformly at random, then R^*'s output is identically distributed to the output of $R^*(x,z)$ after receiving a commitment to v_1 using $\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle$ and so the random variables $\mathsf{sta}^{M,R^*}_{\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle}(0,x,v_1,z)$ and $\mathsf{sta}^{R^*}_{\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle}(x,w,z)$ are identically distributed. On the other hand, if the $2t(n)$ strings $(s^0_1,\ldots,s^0_{t(n)},s^1_1,\ldots,s^1_{t(n)})$ are commitments to 0 and 1 respectively, then R^*'s output is identically distributed to the output of $R^*(x,z)$ after receiving a commitment to v_1 using $\langle \tilde{\mathsf{S}}_{\mathsf{com}},R_{\mathsf{com}}\rangle$ and so the random variables $\mathsf{sta}^{M,R^*}_{\langle \mathsf{S_{com}},\mathsf{R_{com}}\rangle}(1,x,v_1,z)$ and $\mathsf{sta}^{R^*}_{\langle \tilde{\mathsf{S}}_{\mathsf{com}},R_{\mathsf{com}}\rangle}(x,v_1,z)$ are identically distributed. Thus, the lemma is proved. □

This concludes the proof of and Lemma 3 Theorem 2.

□

Functionality $\mathcal{F}_{\mathsf{mcom}}$.

$\mathcal{F}_{\mathsf{mcom}}$ proceeds as follows, running with parties $P_1, \ldots, P_n$ and an adversary S:

- Upon receiving input $(\mathsf{commit}, sid, ssid, P_i, P_j, \beta)$ from P_i, where $\beta \in \{0,1\}$, record the tuple $(ssid, P_i, P_j, \beta)$ and send the message $(\mathsf{receipt}, sid, ssid, P_i, P_j)$ to P_j and S. Ignore any future commit messages with the same $ssid$ from P_i to P_j.
- Upon receiving a value $(\mathsf{reveal}, sid, ssid)$ from P_i: If a tuple $(ssid, P_i, P_j, \beta)$ was previously recorded, then send the message $(\mathsf{reveal}, sid, ssid, P_i, P_j, \beta)$ to P_j and S. Otherwise, ignore.
- Upon receiving a message $(\mathsf{corrupt}-P_i, sid)$ from the adversary, send $(ssid, P_i, P_j, \beta)$ to the adversary for each recorded tuple where P_i is the committer. Furthermore, if the adversary now provides a value β', and the receipt output was not yet written to P_j's tape, then change the recorded value to β'.

Figure 5: $\mathcal{F}_{\mathsf{mcom}}$

D Proof of Main Theorem

We restate our main theorem and provide the proof below.

THEOREM 3 (**Main Theorem** (restatement)). *Assume the existence of a $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure UC-puzzle in a $\mathcal{G}$-hybrid model, an* EQNMCom *protocol secure w.r.t $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$ and the existence of a simulatable PKE scheme. Then, for every "well-formed" functionality $\mathcal{F}$, there exists a protocol Π in the $\mathcal{G}$-hybrid model that realizes $\hat{\mathcal{F}}$ with $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-adaptive UC-security.*

On a high-level, the compilation proceeds in two steps:

- First, every functionality is compiled into a protocol in the $\mathcal{F}_{\mathsf{mcom}}$-hybrid model. In the $\mathcal{F}_{\mathsf{mcom}}$-hybrid, all parties have access to the ideal commitment functionality called $\mathcal{F}_{\mathsf{mcom}}$ functionality. This step is formalized in the $\mathcal{F}_{\mathsf{mcom}}$-lemma (Lemma 4) and essentially follows as corollary from previous works [11, 17, 30, 14, 13].
- In the second step, assuming the existence of a UC-*puzzle* and a EQNMCom protocol, we show that the $\mathcal{F}_{\mathsf{mcom}}$ functionality can be securely realized in the real-model. This step is formalized in the Puzzle-lemma (Lemma 7).

We use the standard definition of the $\mathcal{F}_{\mathsf{mcom}}$ functionality [11], the multi-session extension of $\mathcal{F}_{\mathsf{com}}$-functionality. See Figure 5 for the definition.

Next, we provide the $\mathcal{F}_{\mathsf{mcom}}$-Lemma and the Puzzle Lemma. The proof of the main theorem follows using a standard hybrid argument combining the two lemmas.

Lemma 4 ($\mathcal{F}_{\mathsf{mcom}}$-Lemma). *Assume the existence of simulatable PKE secure w.r.t $\mathcal{C}_{\mathsf{sim}}$. For every well-formed functionality $\mathcal{F}$, there exists a protocol Π in the $\mathcal{F}_{\mathsf{mcom}}$-hybrid model, such that, for every adversary $\mathcal{A} \in \mathcal{C}_{\mathsf{sim}}$ in the $\mathcal{F}_{\mathsf{mcom}}$-hybrid model, there exists an adversary simulator $\mathcal{A}' \in \mathcal{C}_{\mathsf{sim}}$, such that for every environment $\mathcal{Z} \in \mathcal{C}_{\mathsf{env}}$, the following two ensembles are indistinguishable w.r.t $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$.*

- $\left\{\mathsf{Exec}^{\mathcal{F}_{\mathsf{mcom}}}_{\Pi,\mathcal{A},\mathcal{Z}}(\mathsf{n})\right\}_{\mathsf{n}\in\mathbb{N}}$
- $\left\{\mathsf{Exec}^{\hat{\mathcal{F}}}_{\pi_{\mathsf{ideal}},\mathcal{A}',\mathcal{Z}}(\mathsf{n})\right\}_{\mathsf{n}\in\mathbb{N}}$

The main technical contribution of our work is the following lemma:

Lemma 5 (**Adaptive-Puzzle-Lemma**). *Let Π' be a protocol in the $\mathcal{F}_{\mathsf{mcom}}$-hybrid model. Assume the existence of a $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure adaptive puzzle $\langle \mathsf{S}, \mathsf{R} \rangle$ in a $\mathcal{G}$-hybrid model, a stand-alone* EQNMCom *$\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ secure w.r.t $cl(\mathcal{C}_{\mathsf{sim}}, \mathcal{C}_{\mathsf{env}})$ and simulatable PKE scheme secure w.r.t $\mathcal{C}_{\mathsf{sim}}$. Then, there exists a protocol Π in the $\mathcal{G}$-hybrid such that, for every uniform $\mathcal{PPT}$ adversary $\mathcal{A}$, there exists a simulator $\mathcal{A}' \in \mathcal{C}_{\mathsf{sim}}$, such that, for every environment $\mathcal{Z} \in \mathcal{C}_{\mathsf{env}}$, the following two ensembles are indistinguishable over N w.r.t $\mathcal{C}_{\mathsf{sim}}$.*

- $\left\{ \mathsf{Exec}^{\mathcal{G}}_{\Pi,\mathcal{A},\mathcal{Z}}(\mathsf{n}) \right\}_{\mathsf{n} \in \mathbb{N}}$
- $\left\{ \mathsf{Exec}^{\mathcal{F}_{\mathsf{mcom}}}_{\Pi',\mathcal{A}',\mathcal{Z}}(\mathsf{n}) \right\}_{\mathsf{n} \in \mathbb{N}}$

Proof of the Adaptive Puzzle Lemma: First, in Figure 2 we construct a protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ that implements the $\mathcal{F}_{\mathsf{mcom}}$-functionality. Next, given any protocol Π' in $\mathcal{F}_{\mathsf{mcom}}$-hybrid model, the protocol Π in the real model is constructed from Π' by instantiating the $\mathcal{F}_{\mathsf{mcom}}$ functionality using our protocol $\langle \mathsf{S}, \mathsf{R} \rangle$. More precisely, all invocations of the $\mathcal{F}_{\mathsf{mcom}}$ functionality with input (sender,$sid, ssid, P_j, \beta$) from an honest party P_i is replaced with an instance of $\langle \mathsf{S}, \mathsf{R} \rangle$ between P_i and P_j on identity $\mathsf{id} = (\mathsf{P_i}, \mathsf{sid}, \mathsf{ssid})$. We provide the construction of $\langle \mathsf{S}, \mathsf{R} \rangle$ and then prove correctness.

D.1 The Adaptive Commitment Protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ and the Adaptive UC Simulator

Let $(\langle \mathsf{S}, \mathsf{R} \rangle, \mathcal{R})$ be a $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure puzzle in the $\mathcal{G}$-hybrid, $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ be a EQNMCom protocol secure w.r.t $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$, $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ be a non-interactive EQCom protocol secure w.r.t. $cl(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$. $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{oGen}, \mathsf{oRndEnc}, \mathsf{rGen}, \mathsf{rRndEnc})$ be a simulatable PKE scheme (as defined by [17]). Let L be a language in NP with witness relation R_L and let G be a pseudo-random generator (which exists based on one-way function which in turn can be based on simulatable PKE). See Figure 2 for a formal description of the protocol $\langle \mathsf{S}, \mathsf{R} \rangle$. An overview of the protocol is given in Section 4.2.

We show that for every adversary $\mathcal{A} \in \mathcal{PPT}$ in the real-model, there exists a simulator $\mathcal{A}' \in \mathcal{C}_{\mathsf{sim}}$ such that no environment $\mathcal{Z} \in \mathcal{C}_{\mathsf{env}}$ can distinguish if it is interacting with $\mathcal{A}$ in the real-model or $\mathcal{A}'$ in the $\mathcal{F}_{\mathsf{mcom}}$-hybrid.

Consider $\mathcal{A}'$ that internally incorporates $\mathcal{A}$ and emulates an execution with $\mathcal{A}$. $\mathcal{A}'$ forwards all messages from $\mathcal{A}$ externally to its intended recipients except messages that are part of any execution using $\langle \mathsf{S}, \mathsf{R} \rangle$, which are instead, dealt with internally. Recall that, since $\mathcal{A}' \in \mathcal{C}_{\mathsf{sim}}$ we have that at the end of every puzzle interaction where $\mathcal{A}$ controls the sender, $\mathcal{A}'$ can obtain a witness to the puzzle transcript. For messages that are part of an execution of $\langle \mathsf{S}, \mathsf{R} \rangle$, $\mathcal{A}'$ does the following:

Simulating the Communication with $\mathcal{Z}$: Every message that $\mathcal{A}'$ receives from the environment $\mathcal{Z}$ is written to $\mathcal{A}$'s input tape. Similarly, every output value that $\mathcal{A}$ writes to its output tape is copied to $\mathcal{A}'$'s own output tape (to be read later by $\mathcal{Z}$).

The Sender is Corrupted and the Receiver is Honest: $\mathcal{A}'$ does the following:

Preamble:

1. $\mathcal{A}'$ simulates the Adaptive UC-Puzzle while playing the part of the receiver, producing transcript TRANS_1 while extracting the trapdoor y
2. $\mathcal{A}'$ honestly plays the part of the sender in the Adaptive UC-Puzzle producing transcript TRANS_2.

Commit Phase:

1. $\mathcal{A}'$ chooses $r_{Gen} \leftarrow \{0,1\}^k$ and computes $(\text{PK}, \text{SK}) = \mathsf{Gen}(r_{Gen})$, $r = \mathsf{rGen}(r_{Gen})$.
2. $\mathcal{A}'$ uses the simulator for generating equivocal commitments for $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$ and knowledge of the trapdoor y to send an equivocal commitment in Step 1 of Stage 1.
3. Upon receiving the string r^0_{S}, $\mathcal{A}'$ equivocally decommits to $r^0_{\mathsf{R}} = r \oplus r^0_{\mathsf{S}}$. $\mathcal{A}'$ chooses r^1_{R} at random and sends to $\mathcal{A}$ on behalf of R.
4. Upon receiving (C, S_0, S_1) from $\mathcal{A}$ in Step 3, $\mathcal{A}'$ computes $m_0 = \mathsf{Dec}_{\text{SK}}(S_0)$ and $m_1 = \mathsf{Dec}_{\text{SK}}(S_1)$. If m_0 is the valid decommitment of C to bit b, $\mathcal{A}'$ sends the message $(\mathsf{commit}, sid, ssid, \mathsf{S}, \mathsf{R}, \beta)$ to the ideal functionality $\mathcal{F}_{\mathsf{mcom}}$ on behalf of S. Otherwise, if m_1 is the valid decommitment of C to bit β, $\mathcal{A}'$ sends the message $(\mathsf{commit}, sid, ssid, \mathsf{S}, \mathsf{R}, \beta)$ to the ideal functionality $\mathcal{F}_{\mathsf{mcom}}$. If both are invalid, $\mathcal{A}'$ chooses a random bit β and sends $(\mathsf{commit}, sid, ssid, \mathsf{S}, \mathsf{R}, \beta)$ to the ideal functionality $\mathcal{F}_{\mathsf{mcom}}$. Additionally, $\mathcal{A}'$ aborts the simulation of R upon $\mathcal{A}$'s decommitment.

Reveal Phase:

1. Upon receiving a valid decommitment from $\mathcal{A}$, $\mathcal{A}'$ sends the message $(\mathsf{reveal}, sid, ssid)$ to $\mathcal{F}_{\mathsf{mcom}}$.
2. Upon receiving an invalid decommitment from $\mathcal{A}$, $\mathcal{A}'$ aborts the simulation of R.

The Sender is Honest and the Receiver is Corrupted: $\mathcal{A}'$ does the following:

Preamble:

1. $\mathcal{A}'$ honestly plays the part of the sender in the Adaptive UC-Puzzle producing transcript TRANS_1.
2. $\mathcal{A}'$ simulates the Adaptive UC-Puzzle while playing the part of the receiver, producing transcript TRANS_2 while extracting the trapdoor y

Commit Phase:

1. $\mathcal{A}'$ chooses $s \leftarrow \{0,1\}^n$ and computes $r = G(s)$.
2. $\mathcal{A}'$ uses the simulator for generating equivocal commitments for $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$ and knowledge of the trapdoor y to send an equivocal commitment in Step 1 of Stage 2. $\mathcal{A}'$ chooses r^0_{S} at random and sends to $\mathcal{A}$ on behalf of S.
3. Upon receiving the string r^1_{R}, $\mathcal{A}'$ equivocally decommits to $r^1_{\mathsf{S}} = r \oplus r^1_{\mathsf{R}}$. Note that $x = r^1_{\mathsf{S}} \oplus r^1_{\mathsf{R}} = r$.
4. $\mathcal{A}'$ generates a random bit β. Using its knowledge of the trapdoor for common input x the simulator generates an equivocal commitment C for protocol $\langle \mathsf{S_{eq}}, \mathsf{R_{eq}} \rangle$.
5. $\mathcal{A}'$ sets S_β to be an encryption under PK of a decommitment of C to 0 and sets $S_{1-\beta}$ to be an encryption under PK of a decommitment of C to 0
6. $\mathcal{A}'$ forwards (C, S_0, S_1) to $\mathcal{A}$.

Reveal Phase: Upon receiving a message $(\mathsf{reveal}, sid, ssid, \mathsf{S}, \mathsf{R}, \beta)$ from the ideal functionality $\mathcal{F}_{\mathsf{mcom}}$, $\mathcal{A}'$ does the following:

1. $\mathcal{A}'$ reveals the decommitment information for C corresponding to bit b and the randomness used to generate the encryption S_v where $v = 1 - \beta$ if $b = 1$ and $v = \beta$ if $b = 0$.
2. The simulator uses $\mathsf{rRndEnc}$ to produce randomness r_{Enc} such that $\mathsf{oRndEnc}(\mathrm{PK}, r_{\mathsf{Enc}}) = S_{1-v}$, and sends r to the adversary.

The Sender and the Receiver are Honest: $\mathcal{A}'$ must produce a simulated transcript on behalf of the honest parties. $\mathcal{A}'$ does the following: When sending messages on behalf of the honest sender, $\mathcal{A}'$ acts as in the case (see above) where the sender is honest and the receiver is corrupted. When sending messages on behalf of the honest receiver, $\mathcal{A}'$ acts as in the case where the receiver is honest and the sender is corrupted (see above).

Dealing with Corruptions: When the adversary corrupts the sender S, $\mathcal{A}'$ sends the message $(\mathsf{corrupt}-\mathsf{S}, sid)$ to the ideal functionality $\mathcal{F}_{\mathsf{mcom}}$ and receives the value of the bit β. Now, $\mathcal{A}'$ needs to provide $\mathcal{A}$ with the randomness consistent with the (C, S_0, S_1) messages sent on behalf of S as well as the input bit b. $\mathcal{A}'$ does this in the same way as when simulating commitment $(\mathsf{reveal}, sid, ssid, \mathsf{S}, \mathsf{R}, \beta)$ messages in the case of corrupted receiver above.

D.2 Correctness of Simulation

We now proceed to prove correctness of simulation. Recall that the simulator manipulates coin-tosses so that it can equivocate commitments made to the adversary and extract the ones committed to by the adversary. More precisely, for the left-interactions, where the adversary receives commitments, the simulator manipulates the coin-toss to generate the CRS and for the right interactions, where the adversary sends commitments, the simulator manipulates the coin-toss to generate the public-key for the simulatable encryption scheme. In order for the simulation to work successfully, we will require that the adversary not be able to manipulate the other coin-tosses—the coin-toss for generating the public- keys in the left interactions and the coin-toss for generating the CRS in the right interactions. We ensure this property by relying on the non-malleability of the equivocal commitment scheme. In other words, we show that the adversary can never equivocate commitments made using $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ in those coin-toss interactions.

Towards proving correctness, we consider a series of intermediate hybrid experiments from the real-world to the $\mathcal{F}_{\mathsf{com}}$-hybrid world with the adversary $\mathcal{A}$. Additionally, we define the following property that we maintain as invariant across all hybrids and intuitively, will hold true only if the adversary does not equivocate any of the commitments made using $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$: We say that the adversary $\mathcal{A}$ is *non-abusing* if the following two distributions are indistinguishable

$\mathsf{Expr1}_n(z)$: Emulate a complete execution with adversary $\mathcal{A}(1^n)$, environment $\mathcal{Z}$ with auxiliary input z and all honest parties. In the emulated view, choose at random a $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction where $\mathcal{A}$ controls the initiator $\mathsf{I}_{\mathsf{coin}}$. If $\mathcal{A}$ corrupts the corresponding responder $\mathsf{R}_{\mathsf{coin}}$ before Step 3 of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol or fails to complete the interaction, output $(\mathsf{View}_{\mathcal{A}}, \perp)$, where $\mathsf{View}_{\mathcal{A}}$ is the view of the adversary in the simulation. Otherwise, if the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction completes successfully with outcome r, then $(\mathsf{View}_{\mathcal{A}}, r)$.

$\mathsf{Expr2}_n(z)$: As before, emulate an execution with adversary $\mathcal{A}(1^n)$ environment $\mathcal{Z}$ with auxiliary input z and all honest parties. Choose at random a $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction where $\mathcal{A}$ controls the initiator $\mathsf{I}_{\mathsf{coin}}$ and continue the emulation until the completion of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction. If $\mathcal{A}$ corrupts the

corresponding responder $\mathsf{R}_{\mathsf{coin}}$ before Step 3 of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol, or fails to complete the interaction, output $(\mathsf{View}_{\mathcal{A}}, \perp)$, where $\mathsf{View}_{\mathcal{A}}$ is the view of the adversary in the simulation. Otherwise, if the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction completes successfully with outcome r, let r_{I} be the string sent by $\mathcal{A}$ in Step 3 of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol. Repeat the following:

- Choose string r^* uniformly at random.
- Rewind $\mathcal{A}$ to the point right before Step 2 of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol.
- Send string $\mathsf{r}^* \oplus \mathsf{r}_{\mathsf{I}}$ to $\mathcal{A}$ on behalf of $\mathsf{R}_{\mathsf{coin}}$ in Step 2 of the of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol.
- Continue simulation until $\mathcal{A}$ decommits. If the adversary fails to decommit or tries to adaptively corrupt the responder, cancel the simulation and start over. Otherwise, let the value decommitted to be $\tilde{\mathsf{r}}_{\mathsf{I}}$

until the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction completes successfully with outcome $\tilde{\mathsf{r}}$. If $\tilde{\mathsf{r}} \neq \mathsf{r}^*$ then output a special symbol $\perp_{\text{FAIL}}$. Otherwise, output the $(\mathsf{View}_{\mathcal{A}}, \tilde{\mathsf{r}})$.

Remark 2. *If the distributions of* $\mathsf{Expr1}_n(z)$ *and* $\mathsf{Expr2}_n(z)$ *are indistinguishable then it implies that the adversary decommits to the same string* r_{I} *with high-probability, i.e. does not equivocate.*

Remark 3. $\mathsf{View}_{\mathcal{A}}$ *outputted in* $\mathsf{Expr1}_n(z)$ *and* $\mathsf{Expr2}_n(z)$ *are identically distributed.*

Remark 4. *The experiment* $\mathsf{Expr2}_n(z)$*, in expectation, takes polynomial time to simulate. This is because, even though the simulator rewinds the adversary repeatedly, each rewinding is simulated identically as the main simulation with independent randomness. More formally, if* p *is the probability with which the adversary decommits successfully from Step 2 of the coin-toss without corrupting the responder, then* p *is the probability with which the simulator starts rewinding and in expectation rewinds* $1/p$ *times before it obtains another simulation where the adversary decommits without corrupting the responder. Therefore, in expectation, the simulator performs simulation* $p \times 1/p = O(1)$ *times. Since each simulation takes at most* $poly(n)$ *time, in expectation,* $\mathsf{Expr2}_n(z)$ *takes polynomial time to simulate. However, to make* $\mathsf{Expr2}_n(z)$ *useful in our analysis, we introduce the experiment* $\mathsf{EffExpr2}_n(z, q(\cdot))$ *an efficient version of* $\mathsf{Expr2}_n$*, which runs in* strict *polynomial time.*

Before continuing to the Hybrids, we introduce some useful modified experiments. Let $m(\cdot)$ be a function that describes a bound on the maximum number of interactions. Consider the experiment $\mathsf{Expr1}^k_n(z)$ (resp. $\mathsf{Expr2}^k_n(z)$) that proceeds exactly like $\mathsf{Expr1}^k_n(z)$ (resp. $\mathsf{Expr2}^k_n(z)$) with the exception that it chooses the k^{th} coin-toss, where the order of the coin-tosses is determined by the order in which $\mathcal{A}$ decommits in Step 3 and where the adversary controls the initiator. Moreover, in $\mathsf{Expr2}^k_n$ the rewinding is repeated until *both* of the following hold:

- The $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol successfully completes.
- The decommitment corresponding to the rewound coin-toss is again the k-th decommitment of the experiment.

We note that if $\mathsf{Expr1}_n(z)$ and $\mathsf{Expr2}_n(z)$ are indistinguishable then, for every $1 \leq k \leq m(n)$, $\mathsf{Expr1}^k_n(z)$ and $\mathsf{Expr2}^k_n(z)$ are also indistinguishable.

Note that although $\mathsf{Expr2}^k_n(z)$ runs in expected polynomial time, it may not run in strict polynomial time. This is because the number of rewindings in a given execution may be unbounded. Thus, we define an analogous experiment to $\mathsf{Expr2}^k_n(x)$, called $\mathsf{EffExpr2}^k_n(z, q(\cdot))$, which has a bounded run time. Formally, for any polynomial $q(\cdot)$, we define the following experiment:

$\mathsf{EffExpr2}_n^k(z, q(\cdot))$: The experiment proceeds identically to $\mathsf{Expr2}_n^k(z)$ except that there are at most $q(n)$ rewinding attempts. If after $q(n)$ rewinds, the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol has not successfully completed or the corresponding decommitment is not the k^{th} decommitment, abort the experiment and output $\perp$. Otherwise, output whatever $\mathsf{Expr2}_n^k(z)$ outputs.

The next claim quantifies (as a function of $q(\cdot)$) the statistical distance between the distribution over the output of $\mathsf{EffExpr2}^k$ and the distribution over the output of $\mathsf{Expr2}^k$:

Claim 3. *For every polynomial $q(\cdot)$ and for every $n \in \mathbb{N}$, the statistical distance between the following two probability ensembles is at most $\frac{m(n)}{q(n)}$:*

- $\{\overline{\mathsf{Expr2}}_n^k(z)\}_{z \in \{0,1\}^*}$
- $\{\overline{\mathsf{EffExpr2}}_n^k(z, q(\cdot))\}_{z \in \{0,1\}^*}$

where $\overline{\mathsf{Expr2}}_n^k(z)$ and $\overline{\mathsf{EffExpr2}}_n^k(z, q(\cdot))$ are the outputs of $\mathsf{Expr2}_n^k(z)$ and $\mathsf{EffExpr2}_n^k(z, q(\cdot))$, respectively.

Proof. We note that unless an abort occurs in experiment $\mathsf{EffExpr2}$, the random variables $\overline{\mathsf{Expr2}}_n(z)$ and $\overline{\mathsf{EffExpr2}}_n(z, q(\cdot))$ are identically distributed. Thus, the statistical distance can be upperbounded by the probability that $\mathsf{EffExpr2}$ aborts without successful completion of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol in the rewinding stage.

By a standard argument we have that the expected number of rewindings before a successful completion of $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ in experiment $\mathsf{Expr2}$ is $m(n)$. Therefore, by Markov's inequality, the probability that more than $q(n)$ number of rewindings are necessary for successful completion of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol in $\mathsf{Expr2}$ is at most $m(n)/q(n)$. So the probability that $\mathsf{EffExpr2}_n(z, q(\cdot))$ aborts without successful completion of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol in the rewinding stage is at most $m(n)/q(n)$ and the claim is proved. □

The hybrid experiments are as follows:

Hybrid H_0 or the real-world experiment: Since this is the real-world experiment there is no indistinguishability requirement. However, we need to show that $\mathcal{A}$ is non-abusing in H_0. Intuitively, this holds since from the binding property of the commitment scheme $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ we have that if the adversary equivocates, then we can extract the solution of the adaptive UC-puzzle and this violates the soundness of the puzzle. More formally, we prove the following claim:

Claim 4. *$\mathcal{A}$ is non-abusing in H_0*

Assume for contradiction there exists a distinguisher D and polynomial $p(\cdot)$ such that for infinitely many n, D distinguishes $\mathsf{Expr1}_n(z)$ and $\mathsf{Expr2}_n(z)$ with probability at least $\frac{1}{p(n)}$. Since r^* is chosen uniformly at random in each rewound execution (and thus $\mathsf{r}^* \oplus \mathsf{r}_{\mathsf{I}}$ is also uniformly distributed), if $\mathsf{r}^* = \tilde{\mathsf{r}}$ always, then $\mathsf{Expr1}_n(z)$ and $\mathsf{Expr2}_n(z)$ are identically distributed. Hence if D distinguishes the two experiments with probability $\frac{1}{p(n)}$, it must be the case that $\mathsf{Expr2}_n(z)$ outputs $\perp_{\text{FAIL}}$ with probability at least $\frac{1}{p(n)}$. However, we now show that the existence of such an $\mathcal{A}$ implies that there exists a probabilistic polynomial-time adversary $\overline{\mathcal{A}}$ violating the soundness of the adaptive UC-puzzle.

On a high-level, this follows from the fact that whenever $\mathsf{Expr2}_n(z)$ outputs $\perp_{\text{FAIL}}$, the adversary is equivocating, which in turn means a solution to the adaptive UC-puzzle can be extracted and this violates the soundness condition of the puzzle. More formally, consider $\mathcal{A}$ for which $\mathsf{Expr2}_n(z)$ outputs $\perp_{\text{FAIL}}$ with probability $\frac{1}{p(n)}$ for infinitely many n. Fix an n for which this happens.

On input 1^n and auxiliary input z, $\overline{\mathcal{A}}$ internally incorporates $\mathcal{A}(1^n)$, $\mathcal{Z}(z)$ and all honest parties and begins emulating an execution of hybrid experiment H_0 with the following exceptions: $\overline{\mathcal{A}}$ chooses a random

$\langle \mathsf{S}, \mathsf{R} \rangle$ interaction where the adversary controls one of the parties and forwards externally the puzzle interaction where $\mathcal{A}$ controls the receiver. On completion, $\overline{\mathcal{A}}$ chooses the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction from the same $\langle \mathsf{S}, \mathsf{R} \rangle$ interaction where $\mathcal{A}$ controls the initiator $\mathsf{I}_{\mathsf{coin}}$. After completion of Stage 3 of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction, if $\mathcal{A}$ fails to decommit, $\overline{\mathcal{A}}$ outputs $\perp$. Otherwise, it stores the decommitment as r_l. Next, it rewinds to the end of Stage 1 and starts a new emulation (just as in $\mathsf{Expr2}_n$). If the adversary fails to decommit, $\overline{\mathcal{A}}$ rewinds again. Otherwise, it stores the second decommitment as $\tilde{r}_l$. Finally, if $r_l \neq \tilde{r}_l$, it extracts the witness for the puzzle transcript corresponding to this interaction and outputs the witness. Otherwise it outputs $\perp$. Also, if at any point $\mathcal{A}$ tries to adaptive corrupt the other party, $\overline{\mathcal{A}}$ aborts the current execution and rewinds again.

We claim that with non-negligible probability $\overline{\mathcal{A}}$ outputs a witness of the puzzle, thus violating the soundness of the puzzle. Towards this, consider a modified experiment $\mathsf{Expr2}^k_n(z)$ that proceeds exactly like $\mathsf{Expr2}_n$ with the exception that it chooses the k^{th} coin-toss where the adversary controls the initiator instead of a random interaction. By an averaging argument, it holds that there exists a particular k for which $\mathsf{Expr2}^k_n(z)$ outputs $\perp_{\text{FAIL}}$ with some non-negligible probability $1/p(n)$. Now, $\overline{\mathcal{A}}$ begins its internal emulation. If the k^{th} $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction fails to complete successfully, $\overline{\mathcal{A}}$ halts. Otherwise, assume $\mathcal{A}$ successfully decommits to some value r_l in the chosen execution of $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction. Then $\overline{\mathcal{A}}$ rewinds $\mathcal{A}$ to the point right before $\overline{\mathcal{A}}$ sends r^1_{R} on behalf of the responder and instead sends a new random value $r^* \oplus r_1$ $\overline{\mathcal{A}}$ continues to rewind $\mathcal{A}$ for at most $4p(n)$ times or until the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol successfully completes. Note that the execution of $\overline{\mathcal{A}}$ is distributed identically to an execution of $\mathsf{EffExpr2}^k_n(z, 4m(n) \cdot p(n))$. Thus, since the statistical distance between $\mathsf{EffExpr2}^k_n(z, 4m(n) \cdot p(n))$ and $\mathsf{Expr2}^k_n(z)$ is at most $1/4p(n)$, it follows that with non-negligible probability of at least $1/p(n) - 1/4p(n)$ $\mathcal{A}$ outputs a value $\tilde{r}_l \neq r_l$ in the rewound execution. But in the case that $\tilde{r}_l \neq r_l$, $\overline{\mathcal{A}}$ extracts the witness for the puzzle transcript corresponding to this interaction and outputs the witness. Thus, $\mathcal{A}$ violates the soundness of the adaptive UC-puzzle and so the claim is proved.

Hybrid H_1: This hybrid proceeds identically to H_0 with the exception that all puzzle-interactions where the honest party plays the part of the receiver are simulated. For every adversary $\mathcal{A}$, we construct another adversary $\overline{\mathcal{A}} \in \mathcal{C}_{\mathsf{sim}}$ that internally emulates $\mathcal{A}$ and simulates puzzles while extracting trapdoors for all puzzles where $\mathcal{A}$ plays the role of sender.

In more detail, an execution in H_1 proceeds identically to the real-execution, with the exception that all parties running $\langle \mathsf{S}, \mathsf{R} \rangle$, instead of participating in the preamble phase of $\langle \mathsf{S}, \mathsf{R} \rangle$, receive a simulated puzzle-transcript from $\overline{\mathcal{A}}$. Furthermore, for every puzzle interaction where the party controlled by the adversary is the sender and the receiver is honest, $\overline{\mathcal{A}}$ outputs a witness w corresponding to the simulated puzzle-transcript (in a special-output tape). Additionally, upon adaptive corruption of the receiver in a puzzle interaction, where the sender is controlled by the adversary, $\overline{\mathcal{A}}$ produces random coins for an honest receiver that are consistent with the simulated puzzle-transcript. To construct such an $\overline{\mathcal{A}}$ given $\mathcal{A}$, we rely on the adaptive simulatability of the puzzle in a concurrent puzzle execution. We consider an adversary $\mathcal{A}_{\mathsf{puz}}$ that incorporates $\mathcal{A}$ internally and forwards all puzzle interactions with $\mathcal{A}$ as the sender to external receivers. This $\mathcal{A}_{\mathsf{puz}}$ also simulates all other puzzle interactions internally. All other interactions of $\mathcal{A}$ are forwarded by $\mathcal{A}_{\mathsf{puz}}$ to the puzzle environment that incorporates $\mathcal{A}$ and the other honest parties. Since this can be viewed as a concurrent puzzle execution, there must exist a simulator $\mathcal{A}'_{\mathsf{puz}}$ that simulates all puzzle interactions, outputs a witness w, and successfully simulates adaptive corruptions. Finally, to construct $\overline{\mathcal{A}}$ we incorporate $\mathcal{A}'_{\mathsf{puz}}$ and emulate an execution by forwarding the messages between $\mathcal{A}'_{\mathsf{puz}}$ and the actual parties instead of sending to $\mathcal{Z}_{\mathsf{puz}}$.

The proof of indisintguishability follows identically as in [35] and we omit it. The non-abusing property follows from the statistical indistinguishability of $\mathcal{A}$'s view[14] in H_0 and H_1. Hence we have the following

[14] If $\overline{\mathcal{A}}$ is non-abusing, then just as in proof of Claim 4, we can conclude that $\overline{\mathcal{A}}$ is equivocating in H_1. Then with non-negligible probability over the random-tapes for $\overline{\mathcal{A}}$ and partial transcripts where $\overline{\mathcal{A}}$ completes a commitment using $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$, it holds that

claims.

Claim 5. *The output of $\mathcal{Z}$ in H_0 and H_1 is indistinguishable.*

Claim 6. *$\mathcal{A}$ is non-abusing in H_1*

In subsequent hybrids, the adversary we consider is $\overline{\mathcal{A}}$. However, to avoid confusion in notation, we denote the adversary by $\mathcal{A}$ only.

Hybrid H_2: This hybrid proceeds identically to H_1 with the exception that in all interactions with an honest receiver, the commitments received in Stage 1 are switched to simulated equivocal commitments. More specifically, the protocol $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ occurring in Step 1 of the two coin-tosses in Stage 1, is modified for interactions where the adversary plays the part of receiver in $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ in the following ways:

- The first commitment sent by S in the $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ protocol is replaced by a simulated (equivocal) commitment which can be opened to any value.
- The WIPOK's are replaced with simulated adaptively-secure WIPOK's which can be opened consistently with any valid witness.
- When a decommitment is requested, a value r^0_{R} or r^1_{S} (as appropriate) is chosen uniformly at random and a decommitment to the chosen value is produced.

Note that, in particular, this means that in $\mathsf{Expr2}_n(z)$, commitments produced by $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ (where the adversary played the role of receiver) will be decommitted to different values in the initial and rewound views.

Claim 7. *The output of $\mathcal{Z}$ in H_1 and H_2 is indistinguishable.*

Claim 8. *$\mathcal{A}$ is non-abusing in H_2*

Proof: We prove both the above claims simultaneously. They follow essentially from the simulation-extracaibility property of $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$. Recall from proof of Claim 4 that if the outputs of the two experiments are distinguishable, then it implies that $\mathsf{Expr2}_n(z)$ outputs $\perp_{\text{FAIL}}$. Consider the following adversary $\overline{\mathcal{A}}$ that violates the simulation-extractability property of $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$.

On input 1^n and auxiliary input z, $\overline{\mathcal{A}}$ internally incorporates $\mathcal{A}(1^n)$, $\mathcal{Z}(z)$ and all honest parties and begins emulating an execution of hybrid experiment H_1 with the following exception: $\overline{\mathcal{A}}$ forwards all $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ interactions that are part of $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interactions where the adversary controls the receiver are forwarded externally to honest committers on the left and all $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ interactions that are part of $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interactions where the adversary controls the sender are forwarded to external receivers on the right. Whenever $\mathcal{A}$ requests a decommitment for the coin-toss interactions on the left, $\overline{\mathcal{A}}$ externally requests a decommitment. For the decommitment phase on the right, $\overline{\mathcal{A}}$ simply forwards the decommitment made by $\mathcal{A}$ in the internal $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interactions. At the end $\overline{\mathcal{A}}$ outputs $\mathcal{A}$'s view and all the value decommitted to in the right interactions. Observe that when the left commitments are sent by honest committers the view output is identical to view output in H_1 and when the commitments are equivocated, the view is identical to one output in H_2. Furthermore, the simulation proceeds identically to the simulator for the $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ protocol. Since $\mathsf{Expr2}_n(z)$ outputs $\perp_{\text{FAIL}}$ with non-negligible probability, following the proof of Claim 4, it holds that, when the commitment in the left are equivocated, there exists a particular k for

$\overline{\mathcal{A}}$ equivocates with non-negligible probability. This violates the statistical-indistinguishability as for the fixed random tape, $\mathcal{A}$ never equivocates and an unbounded prover, given a partial transcript and random tape, can find the unique value $\mathcal{A}$ decommits to and distinguish from the value decommitted to by $\overline{\mathcal{A}}$.

which $\overline{\mathcal{A}}$ equivocates in k^{th} right-interaction with non-negligible probability. This means that we can simulate $\mathsf{EffExpr2}_n^k(4m(n) \cdot p(n), z)$ and with non-negligible probability $\mathcal{A}$ decommits to a different value in the rewound execution. However, from the simulation-extractability property of $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$, it holds that, whenever the left-commitments are equivocated, there is a unique value that any adversary can decommit to after the commitment stage is completed. Thus, we arrive at a contradiction.

Hybrid H_3: This hybrid proceeds identically to H_2 with the exception that the protocol $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$ occurring in Step 1 of the first coin-toss in Stage 1 is modified for interactions where the adversary plays the part of receiver in $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$ so that instead of sampling a uniformly random r_{R}^0 and decommitting to this value, we sample r_{R}^0 as follows:

- Using $\mathsf{Gen}(r_{\mathsf{Gen}})$ sample a public-key, secret-key pair (pk, sk).
- Run $\mathsf{rGen}(r_{\mathsf{Gen}})$ to obtain the string s such that $\mathsf{oGen}(s) = \text{PK}$.
- Decommit to $r_{\mathsf{R}}^0 = r_{\mathsf{S}}^0 \oplus s$.

We show that both the indistinguishability and the non-abusing property reduce to the indistinguishability of random strings s to strings s sampled by running $\mathsf{rGen}(r_{\mathsf{Gen}})$ where r_{Gen} is sampled uniformly at random. Note that a simulator $\overline{\mathcal{A}}$ emulating an execution in Hybrid H_3 onward can extract the adversary's committed values by decommitting to $r = r^{\mathsf{S}} \oplus s$ such that $\overline{\mathcal{A}}$ knows the corresponding SK for $\mathsf{oGen}(s) = \text{PK}$ and then decrypting the decommitment information contained in S_0 and/or S_1.

Claim 9. *The output of $\mathcal{Z}$ in H_2 and H_3 is indistinguishable.*

Proof. Assume that there exists a PPT algorithm D that distinguishes the output of $\mathcal{Z}$ in H_2 and H_3 with probability $\frac{1}{p(n)}$ for some polynomial $p(\cdot)$ and infinitely many n. We construct an adversary $\overline{\mathcal{A}}$ that will be able to distinguish strings s chosen uniformly at random from string $s = \mathsf{rGen}(r_{\mathsf{Gen}})$ where r_{Gen} is chosen uniformly at random (and thus breaks the oblivious generation property of the simulatable PKE).

Consider a machine $\overline{\mathcal{A}}$ that on input 1^n and auxiliary input z, participates in an execution with a challenger C and internally incorporates $\mathcal{A}$, $\mathcal{Z}$, and all the honest parties and emulates an interaction in H_2. $\overline{\mathcal{A}}$ receives from C a sequence of values $\{s_1, \ldots, s_{m(n)}\}$ chosen either uniformly at random or chosen such that $s_i = \mathsf{rGen}(r_{\mathsf{Gen}}^i)$. $\overline{\mathcal{A}}$ continues the emulation of $\mathcal{A}$ as in H_2 with the difference that in the i-th the commitment protocol $\langle \mathsf{S_{com}}, \mathsf{R_{com}} \rangle$, $\overline{\mathcal{A}}$ decommits to the value $r_{\mathsf{R}}^0 = r_{\mathsf{S}}^0 \oplus s_i$. At the end of the execution, $\overline{\mathcal{A}}$ runs D on the output of $\mathcal{Z}$ and outputs whatever D outputs.

Note that when the strings $\{s_1, \ldots, s_{m(n)}\}$ are generated via $\mathsf{rGen}(r_{\mathsf{Gen}})$ then the emulation produces a view for $\mathcal{Z}$ that is identical to its view in H_3. On the other hand, when the strings $\{s_1, \ldots, s_{m(n)}\}$ are chosen uniformly at random then the emulation produces a view for $\mathcal{Z}$ that is identical to its view in H_2. Thus, $\overline{\mathcal{A}}$ distinguishes random strings s from strings s sampled by running $\mathsf{rGen}(r_{\mathsf{Gen}})$ where r_{Gen} is sampled uniformly at random with the same probability that D distinguishes the ouput of $\mathcal{Z}$ in H_2 and H_3. This implies that $\overline{\mathcal{A}}$ distinguishes with non-negligible probability, which is a contradiction to the security of the simulatable PKE scheme $\mathcal{E}$ and so the claim is proved. □

Claim 10. *$\mathcal{A}$ is non-abusing in H_3*

Proof. The proof for $\mathcal{A}$ being non-abusing essentially follows from the proof of Claim 9 above. Details follow.

Assume towards contradiction that $\mathcal{A}$ is abusing in H_3. Recall that this implies that for some polynomial $p(\cdot)$, some fixed k and for infinitely many n, $\mathsf{Expr2}_n^k(z)$ outputs $\perp_{\text{FAIL}}$ with probability $1/p(n)$. Using $\mathcal{A}$, we will construct an adversary $\overline{\mathcal{A}}$ that breaks the security of the simulatable PKE scheme $\mathcal{E}$. Consider the following adversary $\overline{\mathcal{A}}$: On input 1^n and auxiliary input z, $\overline{\mathcal{A}}$ internally incorporates $\mathcal{A}(1^n)$, $\mathcal{Z}(z)$ and all honest

parties. Additionally, $\overline{\mathcal{A}}$ receives externally a sequence of $4m^2(n) \cdot p(n)$ values, $\{s_1, \ldots, s_{4m^2(n)\cdot p(n)}\}$. $\overline{\mathcal{A}}$ begins emulating an execution of hybrid experiment H_3 and chooses a random $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction where $\mathcal{A}$ controls the initiator $\mathsf{I}_{\mathsf{coin}}$. $\overline{\mathcal{A}}$ continues the emulation of $\mathcal{A}$ with the difference that in the k-th commitment protocol $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$, $\overline{\mathcal{A}}$ decommits to the value $r^0_{\mathsf{R}} = r^0_{\mathsf{S}} \oplus s_k$. After completion of Stage 3 of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction, if $\mathcal{A}$ fails to decommit, $\overline{\mathcal{A}}$ outputs $\perp$. Otherwise, it stores the decommitment as r. Next, it rewinds to the end of Stage 1 and starts a new emulation (just as in $\mathsf{EffExpr2}^k_n$). Again, in the k-th the commitment protocol $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ of the rewound execution, $\overline{\mathcal{A}}$ decommits to the value $r^0_{\mathsf{R}} = r^0_{\mathsf{S}} \oplus s_{m(n)+k}$. If the adversary fails to decommit, $\overline{\mathcal{A}}$ rewinds again, continuing the experiment for at most $4m(n) \cdot p(n)$ number of rewindings. If at the end of the rewinding attempts, $\mathcal{A}$ decommits to $\tilde{\mathsf{r}}$ where $\tilde{\mathsf{r}} \neq \mathsf{r}$, $\overline{\mathcal{A}}$ outputs 1; if $\tilde{\mathsf{r}} = \mathsf{r}$, $\overline{\mathcal{A}}$ outputs 0.

Note that when the strings $\{s_1, \ldots, s_{4m^2(n)\cdot p(n)}\}$ are generated via $\mathsf{rGen}(r_{\mathsf{Gen}})$ then the emulation produces a view for $\mathcal{A}$ that is identical to its view in $\mathsf{EffExpr2}^k_n(z)$ of H_3. Since the distribution of outputs of $\mathsf{EffExpr2}^k_n(z)$ and $\mathsf{Expr2}^k_n(z)$ have statistical distance at most $1/4p(n)$, this implies that in H_3, $\overline{\mathcal{A}}$ outputs 1 with non-negligible probability of at least $1/p(n) - 1/4p(n)$. On the other hand, when the strings $\{s_1, \ldots, s_{4m^2(n)\cdot p(n)}\}$ are chosen uniformly at random then the emulation produces a view for $\mathcal{A}$ that is identical to its view in $\mathsf{EffExpr2}^k_n(z)$ of H_2. Since $\mathcal{A}$ is non-abusing in H_2, we have in this case that $\overline{\mathcal{A}}$ outputs 1 with negligible probability.

So we have that when $\{s_1, \ldots, s_{4m^2(n)p(n)}\}$ are chosen via $\mathsf{rGen}(r_{\mathsf{Gen}})$ $\overline{\mathcal{A}}$ outputs 1 with non-negligible probability and when $\{s_1, \ldots, s_{4m^2(n)p(n)}\}$ are chosen uniformly at random $\overline{\mathcal{A}}$ outputs 1 with negligible probability. Thus, $\overline{\mathcal{A}}$ distinguishes random strings s and strings s sampled by running $\mathsf{rGen}(r_{\mathsf{Gen}})$ where r_{Gen} is sampled uniformly at random. This is a contradiction to the security of $\mathcal{E}$ and so the claim is proved. □

Hybrid H_4: This hybrid proceeds identically to H_3 with the exception that the protocol $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ occurring in Step 1 of the second coin-toss in Stage 1 is modified for interactions where the adversary plays the part of receiver in $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ so that instead of sampling a uniformly random r^1_{S} and decommitting to this value, we sample r^1_{S} as follows:

- Sample s uniformly at random and set $r = G(s)$.
- Decommit to $r^1_{\mathsf{S}} = r^1_{\mathsf{R}} \oplus r$.

Claim 11. *The output of $\mathcal{Z}$ in H_3 and H_4 is indistinguishable. Moreover, $\mathcal{A}$ is non-abusing in H_4*

The proof of Claim 11 proceeds analogously to the proofs of Claims 9 and 10. Here we consider an adversary $\overline{\mathcal{A}}$ that receives externally a sequence of strings $\{s_1, \ldots, s_{4m^2(n)\cdot p(n)}\}$ which are either uniformly random or generated via the pseudorandom generator G. We show that $\overline{\mathcal{A}}$ perfectly emulates an execution in H_3 (or emulates $\mathsf{Expr2}_n(z)$ in H_3) when the received strings are uniformly random and that $\overline{\mathcal{A}}$ perfectly emulates an execution in H_4 (or emulates $\mathsf{Expr2}_n(z)$ in H_4) when the received strings are pseudorandom. Thus, if the output of $\mathcal{Z}$ in H_3 and H_4 is distinguishable or if $\mathcal{A}$ is abusing in H_4 (and not abusing in H_3), then $\overline{\mathcal{A}}$ distinguishes random and pseudorandom strings. This is a contradiction to the security of the pseudorandom generator G, and so the claim is proved.

Hybrid H_5: This hybrid proceeds identically to H_4 with the exception that the protocol $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ occurring in Step 1 of Stage 2 is modified for interactions in which the adversary plays the part of receiver in the following way: The commitment C is replaced by a simulated (equivocal) commitment which can be opened to both 0 and 1.

We show that both the indistinguishability and the non-abusing property reduce to the special-hiding property of $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$.

Claim 12. *The output of $\mathcal{Z}$ in H_4 and H_5 is indistinguishable. Moreover, $\mathcal{A}$ is non-abusing in H_5.*

Proof. The proof of Claim 12 proceeds analogously to the proofs of Claims 9 and 10. Assume for contradiction there exists an environment $\mathcal{Z}$ that distinguishes the experiments H_4 and H_5. More precisely, there exists D and polynomial $p(\cdot)$ such that D distinguishes the output of $\mathcal{Z}$ in both the experiments. We show given $D, \mathcal{Z}$ and $\mathcal{A}$ how to violate the special-hiding property of the commitment (See Definition 2).

Consider a machine $\overline{\mathcal{A}}$ that on input 1^n and auxiliary input z, internally incorporates $\mathcal{A}$, $\mathcal{Z}$, and all the honest parties and emulates an interaction in H_4. Whenever $\mathcal{A}$ wishes to receive a commitment from an honest receiver to a bit β in Stage 2 of $\langle \mathsf{S}, \mathsf{R} \rangle$, instead of constructing C by emulating the $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ interaction internally, $\overline{\mathcal{A}}$ makes a request externally for a commitment C to bit β. When $\mathcal{A}$ expects a decommitment in the internal emulation $\overline{\mathcal{A}}$ again requests the external committer for a decommitment of C to bit β. Finally, $\overline{\mathcal{A}}$ runs D on the output of $\mathcal{Z}$ and outputs what D outputs.

Observe that when the external committer runs the code of the honest committer S in $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$, then the output of $\overline{\mathcal{A}}$ is identically distributed to the output of D in H_4. Similarly, whenever the external committer runs the code of the equivocator in $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$, then the output of $\overline{\mathcal{A}}$ is identically distributed to the output of D in H_5. Therefore, D distinguishes honest and simulated commitments, which is a contradiction to the special-hiding property of $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$.

The proof for $\mathcal{A}$ being non-abusing essentially follows from above. Assume towards contradiction that $\mathcal{A}$ is abusing and thus for some polynomial $p(\cdot)$, some fixed k and for infinitely many n, $\mathsf{Expr2}_n^k(z)$ outputs $\perp_{\text{FAIL}}$ with probability $1/p(n)$. In this case, $\overline{\mathcal{A}}$ will need to request additional external commitments C so that it can simulate $\mathsf{EffExpr2}_n^k(4m(n) \cdot p(n), z)$. Specifically, $\overline{\mathcal{A}}$ begins emulating an execution of hybrid experiment H_5 and chooses a random $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction where $\mathcal{A}$ controls the initiator $\mathsf{I}_{\mathsf{coin}}$. $\overline{\mathcal{A}}$ continues the emulation of $\mathcal{A}$ with the difference that whenever an equivocal commitment is required in Stage 2 of a $\langle \mathsf{S}, \mathsf{R} \rangle$ protocol where $\mathcal{A}$ plays the receiver, $\overline{\mathcal{A}}$ requests an external commitment C and embeds it in the transcript. After completion of Stage 3 of the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction, if $\mathcal{A}$ fails to decommit, $\overline{\mathcal{A}}$ outputs $\perp$. Otherwise, it stores the decommitment as r. Next, it rewinds to the end of Stage 1 and starts a new emulation (just as in $\mathsf{EffExpr2}_n^k$). Again, replacing the equivocal commitment in Stage 2 with an externally supplied commitment. If the adversary fails to decommit, $\overline{\mathcal{A}}$ continues rewinding attempts for at most $4m(n) \cdot p(n)$ number of times. At the end of the rewinding attempts, if $\mathcal{A}$ decommits to $\tilde{\mathsf{r}}$ where $\tilde{\mathsf{r}} \neq \mathsf{r}$, $\overline{\mathcal{A}}$ outputs 1; if $\tilde{\mathsf{r}} = \mathsf{r}$, $\overline{\mathcal{A}}$ outputs 0.

Note that when the external commitments are generated via $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ then the emulation produces a view for $\mathcal{A}$ that is identical to its view in $\mathsf{EffExpr2}_n^k(4m(n) \cdot p(n), z)$ of H_4. Since $\mathcal{A}$ is non-abusing in H_4, we have in this case that $\overline{\mathcal{A}}$ outputs 1 with negligible probability. On the other hand, when the external commitments are generated via $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ then the emulation produces a view for $\mathcal{A}$ that is identical to its view in $\mathsf{Expr2}_n(z)$ of H_5. Since the distribution of outputs of $\mathsf{EffExpr2}_n^k(z)$ and $\mathsf{Expr2}_n^k(z)$ have statistical distance at most $1/4p(n)$, this implies that in H_4, $\overline{\mathcal{A}}$ outputs 1 with non-negligible probability of at least $1/p(n) - 1/4p(n)$.

So we have that when the external commitments are generated via $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$, $\overline{\mathcal{A}}$ outputs 1 with non-negligible probability and when the external commitments are generated via $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$, $\overline{\mathcal{A}}$ outputs 1 with negligible probability. Thus, $\overline{\mathcal{A}}$ distinguishes commitments generated by $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ and commitments generated by $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$. This is a contradiction to the special-hiding property of $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ and so the claim is proved. □

We are now ready to prove our correctness claim:

Claim 13. *For every adversary $\mathcal{A}$ in H_5 that successfully commits to a bit β in Stage 2, it is the case that $\mathcal{A}$ encrypts decommitment information for both 0 and 1 in S_0, S_1 with negligible probability.*

Proof. Assume for contradiction there exists an adversary $\mathcal{A}$ and a value j (where $1 \leq j \leq m(n)$), such that in the j^{th} $\langle \mathsf{S}, \mathsf{R} \rangle$ interaction where $\mathcal{A}$ plays the part of sender, it is the case that $\mathcal{A}$ encrypts decommitment information for both 0 and 1 in S_0, S_1 probability $1/p(n)$ for some polynomial $p(\cdot)$.

Consider a machine $\overline{\mathcal{A}}$ that on input 1^n, auxiliary input z, and non-uniform advice $p(\cdot)$, participates in a security experiment with adversary $\mathcal{A}$. $\overline{\mathcal{A}}$ chooses a sequence of uniformly random strings $\rho_1, \ldots, \rho_{4m(n)\cdot p(n)}$, internally incorporates $\mathcal{A}$, $\mathcal{Z}$ and all the honest parties and emulates an interaction in H_5. Note that with all but negligible probability, none of $\rho_1, \ldots, \rho_{4m(n)\cdot p(n)}$ are in the range of the pseudorandom generator G. $\overline{\mathcal{A}}$ choose s the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction corresponding to the j^{th} $\langle \mathsf{S}, \mathsf{R} \rangle$ interaction where $\mathcal{A}$ controls the initiator $\mathsf{I}_{\mathsf{coin}}$ of the coin-toss. Call this $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction the k^{th} $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction. $\overline{\mathcal{A}}$ will then attempt to fix the outcome of the k^{th} coin toss to some ρ_i for $1 \leq i \leq 4m(n) \cdot p(n)$ so that $x = \rho_i$ in the Stage 2 $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ interaction.

To this end, $\overline{\mathcal{A}}$ begins its internal emulation. If the k^{th} $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction fails to complete successfully, $\overline{\mathcal{A}}$ halts. Otherwise, assume $\mathcal{A}$ successfully decommits to some value r^1_{S} in Step 3 of the chosen execution of $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ (within the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction) and $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ completes with outcome r. Then $\overline{\mathcal{A}}$ rewinds $\mathcal{A}$ to the point right before $\overline{\mathcal{A}}$ sends r^1_{R} on behalf of the responder and instead sends the value $(r^1)'_{\mathsf{R}} = \rho_1 \oplus r^1_{\mathsf{S}}$. $\overline{\mathcal{A}}$ continues to rewind $\mathcal{A}$ for at most $4m(n) \cdot p(n)$ times or until the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol successfully completes. Note that the execution of $\overline{\mathcal{A}}$ is distributed identically to an execution of $\mathsf{EffExpr2}^k_n(z, 4m(n) \cdot p(n))$. Moreover, note that $\overline{\mathcal{A}}$ in H_5 can decrypt and extract the committed values of the adversary, if the adversary starts to construct commitments in Stage 2 with decommitment information to both 0 and 1 encrypted in S_0, S_1. Thus, due to the non-abusing property of H_5, the fact that the outputs of $\mathsf{EffExpr2}^k_n(z, 4m(n) \cdot p(n))$ and $\mathsf{Expr2}^k_n(z)$ are $1/4p(n)$-close, and the fact that $\mathsf{View}_{\mathcal{A}}$ is identically distributed in $\mathsf{Expr1}^k_n(z)$ and $\mathsf{Expr2}^k_n(z)$, we have that the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol completes successfully in the rewound execution with an outcome ρ_i and $\mathcal{A}$ encrypts decommitment information for both 0 and 1 in S_0, S_1 is at least $1/p(n) - 1/4p(n) - \mathrm{neg}(n)$. But this is impossible, since with all but negligible probability, none of $\rho_1, \ldots, \rho_{4m(n)\cdot p(n)}$ are in the range of G. Thus, we have reached contradiction. □

Hybrid H_6: In this hybrid, Step 2 of Stage 2 of $\langle \mathsf{S}, \mathsf{R} \rangle$ is modified for interactions in which the adversary plays the part of receiver in the following way: Decommitment information for both bits 0 and 1 is encrypted in S_0, S_1.

Indistinguishability and the statistical binding property will be reduced to the security properties of the simulatable PKE scheme. Note that to reduce to the indistinguishability of encryptions $\mathsf{Enc}(\mathrm{PK}^*, m, r)$ of a specified message m and strings generated at random via $\mathsf{oRndEnc}(\mathrm{PK}^*, r_{\mathsf{Enc}})$ (where PK^* is generated via oGen with uniform randomness r' which is also given to the adversary), we need to ensure that the outcome of the first coin-toss in Stage 1 of $\langle \mathsf{S}, \mathsf{R} \rangle$ yields the target public key PK^*. Fixing the outcome of the coin-toss will require rewinding the adversary and in order to guarantee that the rewinding strategy is successful, we will rely on the fact that the adversary is non-abusing. More specifically, we consider the intermediate hybrids $H^0_6, H^1_6, \ldots, H^{m(n)}_6$ where $H^0_6 = H_5$ and H^i_6 is the hybrid where the $m(n) - i + 1$-th through $m(n)$-th commitments in interactions $\langle \mathsf{S}, \mathsf{R} \rangle$ where the adversary plays the part of the receiver, are constructed such that decommitment information to both 0 and 1 is encrypted in strings S_0, S_1 of Stage 2.

We are now ready to prove indistinguishability of the Hybrid experiments:

Claim 14. *For $1 \leq k \leq m(n)$ the output of $\mathcal{Z}$ in H^{k-1}_6 and H^k_6 is indistinguishable.*

Note that Claim 14 immediately implies that the output of $\mathcal{Z}$ in H_5 and H_6 is indistinguishable. We now proceed to prove Claim 14.

Proof. Assume for contradiction there exists an adversary $\mathcal{A}$, an environment $\mathcal{Z}$, a value k (where $1 \leq k \leq m(n)$), a distinguisher D and a polynomial $p(\cdot)$ such that for infinitely many n, D distinguishes the output of $\mathcal{Z}$ in H^{k-1}_6 and H^k_6 with probability $1/p(n)$ for some polynomial $p(\cdot)$.

Consider a machine $\overline{\mathcal{A}}_k$ that on input 1^n, auxiliary input z, and non-uniform advice $p(\cdot)$, participates in a security experiment for the simulatable PKE scheme $\mathcal{E}$: $\overline{\mathcal{A}}_k$ receives externally a sequence of uniformly

random values $\{r_1^*, \ldots, r_{4m(n)\cdot p(n)}^*\}$ such that for each $1 \leq i \leq 4m(n) \cdot p(n)$, $\mathsf{oGen}(r_i^*) = \text{PK}_i^*$, internally incorporates $\mathcal{A}(1^n)$, $\mathcal{Z}(z)$ and all the honest parties and emulates an interaction in H_6. Intuitively, $\overline{\mathcal{A}}_k$ will embed one of the challenge obliviously generated public keys and ciphertexts from the external security experiment in Stage 2 of the $m(n) - i + 1$-th execution of the coin tossing protocol $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ (where coin-tosses are ordered according to the order of the decommitments in Step 3). To this end, on input bit β, $\overline{\mathcal{A}}_k$ will play the role of sender and interact with $\mathcal{A}$ in $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ using common input x. $\overline{\mathcal{A}}_k$ runs the equivocator for $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ to construct an equivocal commitment C which it can decommitment to both 0 and 1. $\overline{\mathcal{A}}_k$ chooses $b \in \{0, 1\}$ at random and sets S_b to be an encryption of a decommitment to β. Next, $\overline{\mathcal{A}}_k$ sets the message m in the external experiment to be a correct decommitment to bit $1 - \beta$ and receives challenge ciphertexts $\{S_{1-b}^1, \ldots, S_{1-b}^{4m(n)\cdot p(n)}\}$ (one for each challenge public key). $\overline{\mathcal{A}}_k$ must distinguish whether the ciphertexts $\{S_{1-b}^1, \ldots, S_{1-b}^{4m(n)\cdot p(n)}\}$ are all encryptions of a decommitment to $1 - \beta$ or whether the ciphertexts $\{S_{1-b}^1, \ldots, S_{1-b}^{4m(n)\cdot p(n)}\}$ are all outputted by $\mathsf{oRndEnc}(\text{PK}_i^*, r_{\mathsf{Enc}}^i)$ where r_{Enc}^i is chosen uniformly at random.

Next, $\overline{\mathcal{A}}_i$ begins to run its internal emulation. If $\mathcal{A}$ successfully decommits in the $m(n) - i + 1$-th $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ interaction in the first coin toss of Stage 1 to some value r_{R}^0 and the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ interaction completes, then $\overline{\mathcal{A}}_k$ rewinds $\mathcal{A}$ to the point right before $\overline{\mathcal{A}}_i$ sends r_{S}^0 and instead sends the value $\overline{r}_{\mathsf{S}}^0 = r_1^* \oplus r_{\mathsf{R}}^0$. $\overline{\mathcal{A}}_k$ continues to rewind $\mathcal{A}$ for at most $4m(n) \cdot p(n)$ times or until the $\langle \mathsf{I}_{\mathsf{coin}}, \mathsf{R}_{\mathsf{coin}} \rangle$ protocol successfully completes and the decommitment corresponding to the rewound coin-toss is the k-th decommitment of the rewound execution. Note that up to the $m(n) - i + 1$-th $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ interaction, the execution of $\overline{\mathcal{A}}_k$ is distributed identically to an execution of $\mathsf{EffExpr2}_n^k(z, 4m(n) \cdot p(n))$ in H_5.

Thus, due to the non-abusing property of $\mathcal{A}$ in H_5, and the fact that the output of $\mathsf{EffExpr2}_n^k(z, 4m(n) \cdot p(n))$ and $\mathsf{Expr2}_n^k(z)$ are statistically $1/4p(n)$-close, we have that with probability at least $1 - 1/4p(n) - \text{neg}(n)$, the outcome of the coin toss in the rewound execution is $r_i^* = r_i^* \oplus r_{\mathsf{R}}^0 \oplus r_{\mathsf{R}}^0 = \overline{r}_{\mathsf{S}}^0 \oplus r_{\mathsf{R}}^0$ (for some i). Thus, since $\mathsf{oGen}(r_i^*) = \text{PK}_i^*$, $\overline{\mathcal{A}}_k$ can embed its challenge ciphertext S_{1-b}^i (which is encrypted under public key PK_i^*) in Stage 2 of $\langle \mathsf{S}, \mathsf{R} \rangle$ for the $m(n) - i + 1$-th commitment.

For the $m(n) - i + 2$-th through $m(n)$-th commitments, $\overline{\mathcal{A}}_i$ chooses $b \in \{0, 1\}$ uniformly at random, sets S_b to be an encryption of the decommitment information corresponding to β and sets S_{1-b} to be an encryption of the decommitment information corresponding to $1 - \beta$.

Finally, $\overline{\mathcal{A}}_k$ runs D on the output of $\mathcal{Z}$ and outputs whatever D outputs. Note that if $\{S_{1-b}^1, \ldots, S_{1-b}^{4m(n)\cdot p(n)}\}$ are all outputted by $\mathsf{oRndEnc}(\text{PK}_i^*, r_{\mathsf{Enc}}^i)$, then the output of $\mathcal{Z}$ is $1/4p(n)$-close to the output of $\mathcal{Z}$ in H_6^{k-1}. On the other hand, if $\{S_{1-b}^1, \ldots, S_{1-b}^{4m(n)\cdot p(n)}\}$ are encryptions of a decommitment to $1 - \beta$ under $\text{PK}_1^*, \ldots, \text{PK}_{4m(n)\cdot p(n)}^*$, then the output of $\mathcal{Z}$ is $1/4p(n)$-close to the output of $\mathcal{Z}$ in H_6^k. Thus, the difference between the probability that D outputs 1 in the first case and D outputs 1 in the second case is at least $1/p(n) - 1/4p(n) - 1/4p(n) = 1/2p(n)$. So $\overline{\mathcal{A}}_k$ distinguishes between encryptions of decommitment to $1 - \beta$ and obliviously generated ciphertexts with non-negligible probability. This yields a contradiction to the security of $\mathcal{E}$ and so the claim is proved. □

Claim 15. *For every adversary $\mathcal{A}$ in H_5 that successfully commits to a bit β in Stage 2, it is the case that $\mathcal{A}$ encrypts decommitment information for both 0 and 1 in S_0, S_1 with negligible probability.*

Proof. Since $\mathcal{A}$'s commitments can be extracted in H_5 and H_6, this follows immediately from the indistinguishability of the output of $\mathcal{Z}$ in H_5 and H_6. □

E Puzzle Instantiations

E.1 Adaptive UC in the Common Reference String (CRS) Model

In the common reference string model [8] the parties have access to a CRS choses from a specified trusted distribution D, which is captured via the following ideal functionality $\mathcal{F}_{CRS}^{D}$ (Figure 6) that samples a string r from the distribution D and sets it as a CRS.

Functionality $\mathcal{F}_{CRS}^{D}$

1. Upon activation with an id sid run the sampling algorithm D with uniformly distributed random input $\rho \in \{0,1\}^n$ to obtain a reference string $r = D(\rho)$. Store D, ρ, r and send (sid, r) to the adversary.
2. On input (CRS, sid) from a party P with session id sid', return (sid, r) to that party only if $sid = sid'$, and ignore the message, otherwise.

Figure 6: Common Reference String Functionality

We construct a puzzle in the $\mathcal{F}_{CRS}^{G}$-hybrid, where G is a pseudorandom generator.

Protocol $\langle S, R\rangle$: On input sid, S and R request a common reference string from ideal functionality $\mathcal{F}_{CRS}^{G}$ by sending sid.

Relation: $\mathcal{R} = \{(x, y) | y = G(x)\}$

Figure 7: Common Reference String Puzzle

THEOREM 4. *Assume the existence of a simulatable PKE scheme and the existence of an* EQNMCom *scheme. Let G be a pseudorandom generator. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ with adaptive UC security in the $\mathcal{F}_{CRS}^{G}$-hybrid.*

E.2 Adaptive UC in the Uniform Reference String (URS) Model

When the distribution D in the CRS model is fixed as the uniform distribution, we obtain the uniform reference string model [11]. Let the URS-functionality be $\mathcal{F}_{URS} = \mathcal{F}_{CRS}^{I}$, where I is the identity function. Since the $\mathcal{F}_{CRS}^{G}$-functionality implements $\mathcal{F}_{URS}$-functionality when G is pseudo-random generator, any protocol that realizes f in the $\mathcal{F}_{CRS}^{G}$-hybrid also realized the same functionality in the $\mathcal{F}_{URS}$-hybrid.

THEOREM 5. *Assume the existence of a simulatable PKE scheme and the existence of an* EQNMCom *scheme. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ with adaptive UC security in the $\mathcal{F}_{URS}^{G}$-hybrid.*

E.3 Adaptive UC in the Key Registration Model

In the key registration model [2] includes a service that allows all parties to obtain a public key derived from a seed, which is kept secret by the service. The service is modeled as an ideal functionality $\mathcal{F}_{KR}^{f}$ parameterized by a function $f : \{0,1\}^* \to \{0,1\}^*$, which is presented in Figure 8.

Using the KR-functionality we construct a puzzle as follows:

Functionality $\mathcal{F}_{KR}^{f}$

Upon activation with input sid and security parameter n initialize a set R of empty strings.

Registration: On input message $(register, sid)$ from party P_1 send to the adversary $\mathcal{A}$ $(register, sid, P_i)$ and receive a value p'. If $p' \in R$, then set $p \leftarrow p'$. Otherwise, choose $r \leftarrow \{0,1\}^n$ and set $p \leftarrow f(r)$ and add p to R. Finally, record (P_i, p) and return (sid, p) to both P_i and $\mathcal{A}$.

Registration by corrupted party: On input message $(register, sid, r)$ from a corrupted party P_i, add $P_i, f(r)$ but does not add $f(r)$ to R.

Retrieval: On input $(retrieve, sid, P_i, P_j)$ from party P_j, send $(retrieve, sid, P_i, P_j)$ to $\mathcal{A}$ and get back a value p. If (P_i, p) is recorder, return (sid, P_i, p) to P_j. Otherwise, return $(sid, P_i, \perp)$ to P_j

Figure 8: Key Registration functionality

Protocol $\langle S, R \rangle$: On input sid, R sends $(retrieve, sid, S, R)$ to the ideal functionality $\mathcal{F}_{KR}^{f}$ to obtain a public key.

Relation: $\mathcal{R} = \{(x, y) | y = f(x)\}$

Figure 9: Key Registration Model Puzzle

THEOREM 6. *Assume the existence of a simulatable PKE scheme and the existence of an* EQNMCom *scheme. Let f be a one-way function. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ with adaptive UC security in the $\mathcal{F}_{KR}^{f}$-hybrid.*

E.4 Non-Uniform Adaptive UC

In this model, we consider environments that are $\mathcal{PPT}$ machines and ideal-model adversaries that are n.u.$\mathcal{PPT}$ machines. First, we construct an adaptive puzzle in this model and then state our main theorem. To construct an adaptive puzzle in this model, we make the same complexity theoretic assumptions as those made in [35]; namely, we assume the existence of an *evasive set* $\mathcal{L}$ in P.

Recall the definition of an evasive set [35]:

Definition 10. *A set $\mathcal{L}$ is said to be evasive, if for all n, $S \cap \{0,1\}^n \neq \emptyset$ and for any $\mathcal{PPT}$ machine M, there is a negligible function $v(\cdot)$, such that,* $\Pr[M(1^n) \in S \cap \{0,1\}^n] \leq v(n)$

In [35], several other assumptions sufficient for constructing puzzles in this model. We note that in the adaptive case we can also construct puzzles under each of the assumptions used by [35]. However, for simplicity, we focus only on the assumption that there exists an an evasive set in P.

Lemma 6. *Assume the existence of an evasive set $\mathcal{L}$ in* P. *Then there exists an adaptive puzzle in $(\mathcal{PPT}, n.u.\mathcal{PPT})$ with an empty protocol.*

Proof. Let λ denote the empty string. Define the puzzle $\mathsf{P}_{nu} = (\langle \mathsf{S}, \mathsf{R} \rangle, \mathcal{R})$ as follows (see Figure 10):

We prove soundness and adaptive, statistical simulatability of the puzzle.

Protocol $\langle \mathsf{S}, \mathsf{R} \rangle$: S and R on input 1^n run the empty protocol.

Relation: $\mathcal{R} = \{(x, \lambda) | x \in \mathcal{L}\}$

Figure 10: Non-uniform Puzzle

Soundness: Since $\mathcal{L}$ is evasive, no cheating $\mathcal{PPT}$ receiver can output x such that $(x, \lambda) \in \mathcal{R}$, i.e. $x \in \mathcal{L}$ with more than negligible probability.
Adaptive Simulatability: Consider an adversary $\mathcal{A}$ that participates in a concurrent adaptive puzzle execution with environment $\mathcal{Z}$. We construct a n.u.$\mathcal{PPT}$ adversary $\mathcal{A}'$ that receives $\delta \in \mathcal{L}$ as non-uniform advice and proceeds as follows: It incorporates $\mathcal{A}$ internally and emulates an execution with A. It forwards all messages from $\mathcal{A}$ to $\mathcal{Z}$, except the messages involved in the puzzle interactions with $\mathcal{A}$. However, since the protocol is empty, there are no messages exchanged in the puzzle interaction. Clearly, dealing with adaptive corruptions is trivial since no messages are exchanged in the puzzle interaction. To outputs a witness, $\mathcal{A}'$ simply outputs δ on its special output tape whenever $\mathcal{A}$ sends $(\mathsf{TRANS} = \lambda, C)$ to $\mathcal{Z}$ for a puzzle interaction. Finally, since the interaction between $\mathcal{A}'$ with $\mathcal{Z}$ is identical to the interaction between $\mathcal{A}$ with $\mathcal{Z}$, the real and ideal executions are perfectly indistinguishable to $\mathcal{Z}$. □

THEOREM 7. *Assume the existence of simulatable PKE secure against n.u.$\mathcal{PPT}$, the existence of an* EQNMCom *scheme secure against n.u.$\mathcal{PPT}$, and the existence of an evasive set $\mathcal{L}$. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ with adaptive Non-Uniform UC security.*

E.5 Quasi-Polynomial Adaptive UC

Recall that the Quasi-Polynomial Simulation model is a relaxation of the standard simulation-based definition of security, allowing for a super polynomial-time or Quasi-polynomial simulation (QPS).

THEOREM 8. *Assume the existence of simulatable PKE secure against $\mathcal{PQT}$, the existence of an* EQNMCom *scheme secure against $\mathcal{PQT}$, and the existence of one-way functions that can be inverted with probability* 1 *in $\mathcal{PQT}$. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ with adaptive QPS-UC security.*

We remark that one-way functions that are invertible by $\mathcal{PQT}$ machines as well as EQNMCom schemes can be constructed based on one-way functions with sub-exponential hardness. Thus, assuming simulatable PKE secure against $\mathcal{PQT}$ and one-way function with subexponential hardness, we obtain as a corollary a protocol that securely realizes any functionality with adaptive QPS-UC security.

The notion of security we achieve is analagous to the one in [35] which guarantees that the output of the simulator is indistinguishable also for $\mathcal{PQT}$. This means that anything an attacker can learn "on-line" (in poly-time) can be simulated "off-line" (in qpoly-time) in a way that is indistinguishable also "off-line".

We present the following Adaptive UC-Puzzle in the QPS model (See Figure 11). Let f be a one-way function that can be inverted with probability 1 in $\mathcal{PQT}$.
Soundness: This follows directly from the one-wayness of f and the witness-hiding property of the proof given by the sender.
Adaptive Simulatability: The simulator $\mathcal{A}'$ simply plays the part of the honest receiver. Upon adaptive corruption, $\mathcal{A}'$ reveals the randomness of the honest receiver. Clearly, this simulation is identically distributed to a real execution. To output a witness, we require $\mathcal{A}'$ to compute the inverse of $y = f(x)$ for a random x. While emulating $\mathcal{A}$, if $\mathcal{A}$ completes a puzzle-interaction by convincing the receiver in the WHPOK, then $\mathcal{A}'$

Protocol $\langle S, R\rangle$:

$S \to R$: Pick $x \leftarrow \{0,1\}^n$ and send $y = f(x)$ to R.

$S \leftrightarrow R$: a witness-hiding argument of knowledge of the statement that there exists x' such that $y = f(x')$.

Relation: $\mathcal{R} = \{(x, y)|y = f(x)\}$

Figure 11: QPS Puzzle

Functionality $\mathcal{F}_{\mathsf{sun}}$.

1. Upon activation with session id sid proceed as follows. Send the message $(\mathsf{Activated}, sid)$ to the adversary, and wait to receive bad a message (n, sid, D). Run the sampling algorithm d on a uniformly distributed random input ρ from $\{0,1\}^n$ to obtain a reference string $r = D(\rho)$. Store D, ρ, r and send $(\mathsf{CRS}, sid, r, \rho)$ to the adversary.
2. When receiving input (CRS, sid) from some party P with session id sid', send (CRS, sid, r) to that party if $sid = sid'$; otherwise ignore the message.

Figure 12: $\mathcal{F}_{\mathsf{sun}}$

inverts x to obtain a witness y such that $y = f(x)$. If an inverse exists, it finds one since f is invertible by $\mathcal{PQT}$ machines. From the soundness property of the WHPOK, it follows that, if $\mathcal{A}$ convinces the receiver, then except with negligible probability, x has an inverse w.r.t. f.

E.6 Adaptive UC in the Sunspots model

Below we describe the functionality $\mathcal{F}_{\mathsf{sun}}$ (See Figure 12).

We construct an adaptive UC-puzzle in the sunspots model which relies on a statistically hiding commitments $\langle C, R\rangle$ with additional algorithms (C^*, Adap) that have the following properties:

Invertibility: For every (expected) PPT machine R^*, let τ be the transcript of the interaction between R^* and C on input bit β and random tape $r \in \{0,1\}^*$ for C. Then $\mathsf{Adap}(r, \tau)$ produces a random tape r' such that $\langle C^*, R^*\rangle$ yields transcript τ when C^* uses random tape r'.

Strong Oblivious Simulation: For every (expected) PPT machine R^*, it holds that, the following ensembles are statistically indistinguishable over $n \in N$.

- $\{(\mathsf{sta}^{R^*, r_1}_{\langle C^*, R\rangle, r_1}(z), r_1)\}_{n\in N, r_1, r_2\in\{0,1\}^n, z\in\{0,1\}^*, \beta\in\{0,1\}}$
- $\{(\mathsf{sta}^{R^*}_{\langle C, R\rangle, r_2}(\beta, z), \mathsf{Adap}(r_2, \tau))\}_{n\in N, r_1, r_2\in\{0,1\}^n, z\in\{0,1\}^*, \beta\in\{0,1\}}$

where $\mathsf{sta}^{R^*, r_1}_{\langle C^*, R\rangle}(\beta, z)$ denotes the random variable describing the output of R^* after receiving a commitment from C^* using random tape r_1, $\mathsf{sta}^{R^*}_{\langle C, R\rangle, r_2}(\beta, z)$ denotes the random variable describing the output of R^* after receiving a commitment from C to bit β using random tape r_2 and τ denotes the transcript produced by $\langle C, R\rangle$.

We note that the standard construction of statistically-hiding commitment scheme from collision-resistant hash function (CRHF) fulfills the above definition when the CRHF has "random outputs" (i.e. for randomly chosen input x, the output of the CRHF is statistically indistinguishable from random). Such a CRHF was constructed by [28] from lattice-based assumptions. Additionally, the construction of statistically-hiding commitment from one-way permutation (OWP) of [40] has the desired properties, since C^* can simply choose a random image $y = \pi(x)$ of the OWP π, without knowing the corresponding x and run the interactive hashing protocol obliviously. We note that the construction of [40] relies on a general hardness assumption but requires $\mathrm{poly}(n)$ rounds while the construction of [28] relies on a concrete hardness assumption but is constant-round. For concreteness, we state the theorem below for the case of CRHF with random output. Our proof is written for the general case, assuming any commitment scheme that satisfies the properties above.

THEOREM 9. *Assume the existence of simulatable PKE, collision-resistant hash-functions with random output, and an* $\mathsf{EQNMCom}$ *scheme. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π in the $\mathcal{F}_{\mathsf{sun}}$-hybrid that realizes $\hat{\mathcal{F}}$ with adaptive UC-security w.r.t. (μ, d, t)-conforming adversaries where $\mu(n) - d(n) > n^{\epsilon}$ for some $\epsilon > 0$.*

We first consider a $\mathcal{F}_{\mathsf{sun}}^{G}$-hybrid model, where $\mathcal{F}_{\mathsf{sun}}^{G}$ is the ideal functionality identical fo $\mathcal{F}_{\mathsf{sun}}$, with the exception that, instead of running the sampling algorithm D on a uniformly distributed ρ, it runs D on input $G(x)$ for a uniformly random x, where G is a pseudo-random generator. We conclude that the protocol constructed in the $\mathcal{F}_{\mathsf{sun}}^{G}$-hybrid also securely realizes the functionality in the $\mathcal{F}_{\mathsf{sun}}$-hybrid.

We proceed towards constructing a puzzle in the $\mathcal{F}_{\mathsf{sun}}^{G}$-hybrid. Let $G : \{0,1\}^{n^{\delta}} \rightarrow \{0,1\}^{*}$ be a pseudo-random generator that expands a seed of length n^{δ} (for $\delta > 0$) to a stream of bits such that $d(n) + n^{\delta} + |G| < \mu(n)$. Such a $\delta(n)$ is guaranteed to exist since $\mu(n) - d(n) > n^{\epsilon}$. Such a generator can be constructed from any one-way function, which exists under the assumption of simulatable PKE.

Our construction of the puzzle is similar to the construction used in [35] (which is based on [12]), the only difference is the type of commitment we use in the construction.

Let $(V_1, P_1, V_2, P_2, V_3)$ be the respective verifier and prover algorithms for a public-coin universal argument for the language

$$\mathcal{L}_{\mathsf{KOL}} = \{r \mid r \in \{0,1\}^{n} \text{ and } \mathsf{KOL}(r) < n^{\frac{\epsilon+\delta}{2}}\},$$

where $\mathsf{KOL}(x)$ is the *Kolmogorov complexity* of a string x. Such a system can be constructed based on collision-resistant hash functions. We describe a language of transcripts of universal arguments in which the prover's messages are committed instead of sent to the verifier. In order to commit, we use a special statistically hiding commitment scheme $\langle C, R \rangle$, which satisfies the properties listed above. We describe the puzzle construction below (See Figure 13).

Soundness: Suppose a $\mathcal{PPT}$ receiver R^* is able to break the soundness by outputting the witness for a puzzle with probability p. We use R^* to construct another efficient algorithm P which breaks the soundness property of the universal argument system with probability $\mathrm{poly}(p)$. The soundness of the universal argument system therefore implies that p must be negligible which implies the soundness of the puzzle. We show that P breaks the soundness of the universal argument w.p. $\mathrm{poly}(p)$ on the statement that the reference string r sampled from $\mathcal{F}_{\mathsf{sun}}^{G}$-functionality has a "short" description. Since G is pseudo-random, if p is non-negligible, then P breaks the soundness with non-negligible probability in the hybrid experiment when r is sampled from the $\mathcal{F}_{\mathsf{sun}}$ functionality. Since, D has min-entropy $\mu(n)$, w.p. at most $2^{-n^{\frac{\epsilon-\delta}{2}}} = 2^{-O(n^{\epsilon})}$, r has a short description and therefore no computationally bounded prover can succeed in the universal argument with non-negligible probability. Thus, p is negligible.

More precisely, P upon receiving the verifier message v_1, feeds v_1 to R and then internally simulates the rest of the puzzle until R outputs the witness. By hypothesis, this succeeds with probability p. Let p_1 be a decommitment to the first message sent by R. P forwards p_1 externally to the verifier and receives the next

Protocol $\langle S, R\rangle$: S and R obtain the reference string r from the $\mathcal{F}_{\mathsf{sun}}^{G}$-functionality.

$S \to R$: Pick $v_1 \leftarrow V_1(r, n)$ and send to R.

$R \leftrightarrow S$: R and S interact using $\langle C^*, R\rangle$, where R plays the role of C^*. We denote by c_1 the resulting transcript.

$S \to R$: Pick $v_2 \leftarrow V_2(r, n)$ and send to R.

$R \leftrightarrow S$: R and S interact using $\langle C^*, R\rangle$, where R plays the role of C^*. We denote by c_2 the resulting transcript.

Relation:

$$\mathcal{R} = \left\{ (\mathsf{TRANS}, w) \mid \begin{array}{l} \mathsf{TRANS} = (r, v_1, c_1, v_2, c_2), w = ((p_1, r_1), (p_2, r_2)) \\ c_1 \leftarrow \langle S, R\rangle(p_1, r_1), c_2 \leftarrow \langle S, R\rangle(p_2, r_2) \text{ and} \\ V_3(s, v_1, p_1, v_2, p_2) = 1 \end{array} \right\}$$

Figure 13: Sun Spots Puzzle

message v_2. At this point, P rewinds R and feeds v_2 instead of the second message (simulated before) from the verifier and continues to simulate the rest of the puzzle. If R outputs a witness $((p_1', r_1'), (p_2, r_2))$ then we argue that the p_1' outputted must, with all but negligible probability, be the same as p_1 outputted during the first simulation. Otherwise, R breaks the binding of the equivocal commitment and we obtain a witness M to $r \in \mathcal{L}_{\mathsf{KOL}}$. In particular, this means that R distinguishes the output of G from a truly random string. Now, we argue that with probability at least p^2, the transcript (v_1, p_1, v_2, p_2) is an accepting transcript for the universal argument.

Adaptive Simulatability: We achieve statistical simulation by allowing the simulator $\mathcal{A}'$ to set the reference string and obtain the witness, which is the description of D, G and x, whose combined size by construction is $n^{\delta} + O(1) + d(n) < n^{\frac{\epsilon+\delta}{2}}$. Furthermore, while emulating a receiver in a puzzle with adversary $\mathcal{A}$, instead of following the honest receiver's code, $\mathcal{A}'$ runs the protocol $\langle S, R\rangle$ with the sender S in the second and fourth step of the puzzle interaction. The simulator runs the code of an honest prover (P_1, P_2) in the universal argument with witness (D, G, x) obtaining transcript (v_1, p_1, v_2, p_2) and sends commitments to p_1 and p_2 using $\langle S, R\rangle$. Thus, the values committed to by $\mathcal{A}'$ and the randomness used to commit amount to a trapdoor for the puzzle. Upon adaptive corruption, $\mathcal{A}'$ uses Adap to produce randomness r' to show that the transcript "could have" been produced using C^*. Notice that, due to the properties of $\langle S, R\rangle$, even after the randomness r' has been produced, the puzzle sender's view is statistically indistinguishable in the real and simulated interaction.

E.7 Adaptive Bounded Concurrent MPC in the plain model

As in the sunspots model, we construct an adaptive UC-puzzle in the bounded concurrent setting which relies on a statistically hiding commitments $\langle C, R\rangle$ with additional algorithms (C^*, Adap). For concreteness, we again state the theorem below for the case of CRHF with random output. Our proof is written for the general case, assuming any commitment scheme that satisfies the required properties.

THEOREM 10. *Assume the existence of simulatable PKE, collision-resistant hash-functions with random output, and an* EQNMCom *scheme. Then, for every well-formed functionality $\mathcal{F}$, there exists a protocol π in the standard model that realizes $\hat{\mathcal{F}}$ with adaptive security under m-bounded concurrent composition.*

Our construction of the puzzle leverages the non-black box techniques of Barak [1] (which were sub-

Common input: 1^n

Length paramter: $\ell(n)$

Protocol $\langle S, R\rangle$:

Trapdoor Generation:

$S \to R$: Choose $h \leftarrow \{0,1\}^n$ (where h defines a collision resistant hash function with range $\{0,1\}^n$) and send to R.

$R \leftrightarrow S$: R and S interact using $\langle C^*, R\rangle$, where R plays the role of C^*. We denote by c the resulting transcript.

$S \to R$: In the i-th concurrent execution, Choose $r \leftarrow PRF_s(i)$, where PRF is a pseudorandom function and s is a secret key used by S in all concurrent executions and send r to R.

Universal Argument

$S \to R$: Pick $v_1 \leftarrow V_1(r, n)$ and send to R.

$R \leftrightarrow S$: R and S interact using $\langle C^*, R\rangle$, where R plays the role of C^*. We denote by c_1 the resulting transcript.

$S \to R$: Pick $v_2 \leftarrow V_2(r, n)$ and send to R.

$R \leftrightarrow S$: R and S interact using $\langle C^*, R\rangle$, where R plays the role of C^*. We denote by c_2 the resulting transcript.

Relation:

$$\mathcal{R} = \left\{ (\mathsf{TRANS}, w) \;\middle|\; \begin{array}{l} \mathsf{TRANS} = (h, c, r, v_1, c_1, v_2, c_2), w = (M, s, s_1, s_2, y, p_1, p_2) \\ c \leftarrow \langle S, R\rangle(h(M), s), c_1 \leftarrow \langle S, R\rangle(p_1, r_1), c_2 \leftarrow \langle S, R\rangle(p_2, r_2) \text{ and} \\ V_3(s, v_1, p_1, v_2, p_2) = 1 \text{ and} \\ \mathcal{U}(M, c, y) \text{ outputs } r \text{ within } n^{\log\log n} \text{ steps, where } \mathcal{U} \text{ is a universal Turing} \\ \text{machine and} |y| \leq |r| - n. \end{array} \right\}$$

Figure 14: Bounded Concurrent Puzzle

sequently extended to the setting of bounded concurrent secure computation of general functionalities [36, 43]). The main difference is the type of commitment we use in the construction. We remark that our construction is inspired by [26], where they show how to construct stand-alone adaptively secure multi-party computation. While they need Barak's protocol to be non-malleable and rely on techniques from [42], we only need the original construction by Barak.

We define the following relation $R_{\mathcal{U}}^{T(n)}$. We say that $(\langle M, x, t\rangle, w) \in R_{\mathcal{U}}^{T(n)}$, where M is a description of a Turing machine, x, w are strings and t is a number, if M accepts (x, w) within t steps and $t \leq T(|\langle M, s, t\rangle|)$. We define the language $L_{\mathcal{U}}^{T(n)} = L(R_{\mathcal{U}}^{T(n)})$.

Let $(V_1, P_1, V_2, P_2, V_3)$ be the respective verifier and prover algorithms for the public-coin universal argument presented in [1] for the language $L_{\mathcal{U}}^{n^{\log\log n}}$.

We describe the puzzle construction in Figure 14.

In the puzzle protocol defined in Figure 14, we require that $\ell(n)$ fulfills the following requirement: All messages sent to S in concurrent executions between the time that R sends the commitment c and S replies

with $r \in \{0,1\}^{\ell(n)}$ can be described in less than $\ell(n)$ bits. Assume that the length of all party's messages (except the string r sent when the party plays the part of the puzzle Sender in the Trapdoor Generation phase) in a single execution of the protocol computing some functionality f is bounded by a polynomial $q(n)$. We will show below that for m concurrent executions, taking $\ell(n) = m \cdot q(n) + 2n$ is sufficient.

Soundness: Suppose a $\mathcal{PPT}$ receiver R^* is able to break the soundness by outputting the witness for a puzzle with probability p. We use R^* to construct another efficient algorithm $\mathcal{A}$ which either breaks the security of the CRHF h, breaks the binding property of the commitment scheme $\langle S, R\rangle$ or breaks the security of the PRF.

More precisely, $\mathcal{A}$ plays the role of the honest sender S with R^* and simulates the entire puzzle protocol until R outputs the witness. However, $\mathcal{A}$ sends random strings r instead of pseudorandom strings. By hypothesis, this succeeds with probability p (otherwise we can distinguish between pseudorandom and random strings). Let $((p_1, r_1), (p_2, r_2))$ be the witness outputted by R^*, where the corresponding transcript (v_1, p_1, v_2, p_2) is an accepting transcript for the universal argument. Now, we may rewind again and submit a different message v_2'. With probability p, R^* will output a witness $((p_1', r_1'), (p_2', r_2'))$ where the transcript (v_1, p_1, v_2', p_2') is an accepting transcript. Moreover, the p_1' outputted must, with all but negligible probability, be the same as p_1 outputted during the first simulation. Otherwise, R^* breaks the binding of the commitment $\langle S, R\rangle$. We repeat this process a polynomial (expected) number of times until we are able to extract the witness M, s.

Next, we rewind to the Trapdoor Generation phase of the protocol and send a different message r in the second message from S to R^*. Now, by repeating the process described above, we ultimately extract a different witness M', s' such that $c = \langle S, R\rangle(h(M), s) = \langle S, R\rangle(h(M'), s')$. Thus either finding a collision in the CRHF h or breaking the binding property of the commitment scheme $\langle S, R\rangle$.

Adaptive Simulatability: We achieve statistical simulation by having the simulator $\mathcal{A}'$ do the following:

- If the simulator $\mathcal{A}'$ plays the part of the puzzle Receiver in the i-th execution, $\mathcal{A}'$ commits to the code of the adversary, the simulated incoming messages (other than the incoming strings r) and to the PRF key s, as in [1, 45]. Note that the length of this commitment is at most $m \cdot q(n) + n$ and that it correctly describes the next message function of the adversary.

- The simulator runs the code of an honest prover (P_1, P_2) in the universal argument obtaining transcript (v_1, p_1, v_2, p_2) and sends commitments to p_1 and p_2 using $\langle S, R\rangle$. Thus, the values committed to by $\mathcal{A}'$ and the randomness used to commit amount to a trapdoor for the puzzle.

- Upon adaptive corruption, $\mathcal{A}'$ uses Adap to produce randomness r' to show that the transcript "could have" been produced using C^*. Notice that, due to the properties of $\langle S, R\rangle$, even after the randomness r' has been produced, the puzzle sender's view is statistically indistinguishable in the real and simulated interaction.

E.8 Adaptive UC in the Timing model

We prove feasibility of our result in the timing model, which is the same as presented in [35], in the following theorem.

THEOREM 11. *Let $\epsilon > 1$ and $\Delta > 0$ be constants. Assume the existence of simulatable PKE and a $2\epsilon^2\Delta$-delayed* EQNMCom *scheme. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ with $(\Delta, \epsilon, 2\epsilon^2\Delta)$-timed adaptive UC-security.*

For the proof of the above theorem we need to show that Lemma 5 holds the timing model and also adapt the definition of a puzzle to handle entities with clock tapes. To achieve the first task we require that the puzzle environment is δ-delaying and soundness and simulatability hold with respect to ϵ-drift preserving adversaries. Thus we obtain the following claim for the lemma:

Lemma 7 (**Adaptive-Puzzle-Lemma in the Timing Model**). *Let $\epsilon > 1$ and $\Delta > 0$ be constants. Let Π' be a $\epsilon^2\Delta$-delayed protocol in the $\mathcal{F}_{\mathsf{mcom}}$-hybrid model. Assume the existence of a $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-secure adaptive puzzle $\langle \mathsf{S}, \mathsf{R} \rangle$ in a $\mathcal{G}$-hybrid model, $\epsilon^2\Delta$-delayed stand-alone $\mathsf{EQNMCom}$ $\langle \mathsf{S}_{\mathsf{com}}, \mathsf{R}_{\mathsf{com}} \rangle$ secure w.r.t $cl(\mathcal{C}_{\mathsf{sim}}, \mathcal{C}_{\mathsf{env}})$ and simulatable PKE scheme secure w.r.t $\mathcal{C}_{\mathsf{sim}}$. Then, there exists a protocol Π in the $\mathcal{G}$-hybrid such that, for every uniform $\mathcal{PPT}$ adversary $\mathcal{A}$ that is ϵ-drift preserving, there exists a simulator $\mathcal{A}' \in \mathcal{C}_{\mathsf{sim}}$, such that, for every $\epsilon^2\Delta$-delaying environment $\mathcal{Z} \in \mathcal{C}_{\mathsf{env}}$, the following two ensembles are indistinguishable over N w.r.t $\mathcal{C}_{\mathsf{sim}}$.*

- $\left\{ \mathsf{Exec}^{\mathcal{G}}_{\Pi,\mathcal{A},\mathcal{Z}}(\mathsf{n}) \right\}_{\mathsf{n}\in\mathbb{N}}$
- $\left\{ \mathsf{Exec}^{\mathcal{F}_{\mathsf{mcom}}}_{\Pi',\mathcal{A}',\mathcal{Z}}(\mathsf{n}) \right\}_{\mathsf{n}\in\mathbb{N}}$

We adapt the proof of Lemma 5 to the timing model. There we considered a sequence of hybrid experiments starting with the execution of the adversary $\mathcal{A}$ in the real world to the execution with the simulator $\mathcal{A}'$ in the $\mathcal{F}_{\mathsf{com}}$-hybrid world. We constructed non-abusing adversaries in each of the hybrids and showed that the executions in the hybrids are indistinguishable for the environment $\mathcal{Z}$. The first step was to construct an adversary relying on the simulatability of the puzzles. In hybrid H_1 we construct an adversary $\mathcal{A}'$ that incorporates $\mathcal{A}$ and simulates all puzzles interactions. In order to show that hybrid H_0 (the real world) and hybrid H_1 are indistinguishable we constructed an adversary $\mathcal{A}_{puz}$ in a concurrent puzzle execution, which incorporates $\mathcal{A}$ and emulates the interaction of $\mathcal{A}$'s environment. Thus the indistinguishability of H_0 and H_1 is reduced to indistinguishability of $\mathcal{Z}_{puz}$ in concurrent puzzle execution with $\mathcal{A}_{puz}$ and its simulator $\mathcal{A}'_{puz}$. To ensure that this holds in the timing model we require that (1) $\mathcal{Z}_{puz}$ is $\epsilon^2\delta$-delaying environment and (2) the internal emulation of the execution by $\mathcal{A}_{puz}$ is identical to H_0. The first conditions holds since $\mathcal{Z}_{puz}$ incorporates $\mathcal{Z}$ and the honest parties and emulates only the interactions of these parties that are not part of the puzzle-interactions. Thus all messages sent from $\mathcal{Z}$ or the honest parties to the adversary that are forwarded from $\mathcal{Z}_{puz}$ to $\mathcal{A}_{puz}$ are $\epsilon^2\delta$-delayed since $\mathcal{Z}$ is $\epsilon^2\delta$-delaying, and messages from the honest parties which are not part of the puzzles interactions are $\epsilon^2\delta$-delayed.

In order to satisfy the second condition we have to account for the special messages $(time, *, *)$ and $(reset - time, *, *)$ that the adversary $\mathcal{A}$ can send to alter the parties' clock-tapes. We introduce two modifications of $\mathcal{A}_{puz}$ and $\mathcal{Z}_{puz}$ to achieve this. First, we require that $\mathcal{A}_{puz}$ forward all special messages from $\mathcal{A}$ to $\mathcal{Z}_{puz}$ and also adjust appropriately the local clock-tapes of the parties in the internal emulation. Since $\mathcal{A}_{puz}$ forwards to the external receiver the messages between $\mathcal{A}$ and the honest parties where $\mathcal{A}$ acts as a sender, we need to synchronize the clocks of those external receivers for the puzzle interactions. For this we require that $\mathcal{Z}_{puz}$ forward the appropriate message for the clock-tapes to the external receives. The above modifications of $\mathcal{A}_{puz}$ and $\mathcal{Z}_{puz}$ suffice for the proof of the non-abusing property as well (the only difference in the puzzle environment is the final output). The rest of the hybrids in the proof of the lemma are the same as before since they use the simulated puzzles and rely only on the EQNMCOM properties.

We turn towards constructing an adaptive puzzle in the timing model. Define the puzzle $\mathcal{P}_{tim}(\langle S, R \rangle, \mathcal{R})$ as follows (see Figure 15).

Soundness: The soundness of the puzzle follows directly from the one-wayness of f and the witness-hiding property of the protocols.

Adaptive Simulatability: To simulate a concurrent puzzle-execution with $\mathcal{A}$ and its environment $\mathcal{Z}$, $\mathcal{A}'$, as before, internally emulates an execution with $\mathcal{A}$ while playing the role of the honest receiver. Upon adaptive corruption, $\mathcal{A}'$ simply reveals the inputs and randomness used while running the code of the honest receiver during puzzle interactions (note that the inputs and randomness used in puzzle interactions are independent of the inputs of the honest receiver to the commitment functionality). To extract the witness in a puzzle challenged by $\mathcal{A}$, $\mathcal{A}'$ essentially rewinds $\mathcal{A}$ in the witness-hiding proof-of-knowledge sub-protocol to obtain another accepting transcript. Using the special-sound property of the proof-of-knowledge protocol, the

Protocol $\langle S, R \rangle$:

$S \rightarrow R$: Pick $x \leftarrow \{0,1\}^n$ and send $y = f(x)$ to R.

$S \leftrightarrow R$: a witness-hiding special-sound argument of knowledge of the statement that there exists x' such that $y = f(x')$. R issues a $\mathsf{time-out}$ if more than $2\epsilon\Delta$ local time units elapsed since the challenge in the WHPOK was issued and the response was received from S.

Relation: $\mathcal{R} = \{(x, y) | y = f(x)\}$

Figure 15: Timing Model Puzzle

adversary $\mathcal{A}'$ can then extract the witness used in the proof and outputs that as the witness for the puzzle transcript.

More formally, whenever $\mathcal{A}$ completes a puzzle-interaction with a receiver, $\mathcal{A}'$ temporarily stalls the emulation and rewinds $\mathcal{A}$ to the state where it receives a challenge in the WHPOK sub-protocol. It feeds a new challenge and continues the emulation to obtain a response. While performing emulation from a given challenge, $\mathcal{A}$ expects to exchange messages with $\mathcal{Z}$ and other receivers. Since, the receivers are internally emulated, messages exchanged between $\mathcal{A}$ and the receivers can be emulated internally. Messages exchanged with $\mathcal{Z}$ are delicate, since we cannot rewind the external $\mathcal{Z}$. Note, however, that in a rewinding, $\mathcal{A}$ receives two kinds of messages from $\mathcal{Z}$: (1) messages that were sent before the new challenge was fed to $\mathcal{A}$ in a rewinding, and (2) messages that were sent after. The former messages were received by $\mathcal{A}$ in the main execution can be replayed by $\mathcal{A}'$ to $\mathcal{A}$. For the latter kind of messages, we claim that $\mathcal{A}'$ does not have to emulate them. As $\mathcal{A}$ is ϵ-drift-preserving, the receivers clock-tape advances at least $2\epsilon\Delta\frac{1}{\epsilon}$ time units before the puzzle-environment's clock-tape advances $2\epsilon\Delta$ time units. Since, the receiver issues a time-out when its clock-tape advances $2\epsilon\Delta$ steps since it sent the challenge, $\mathcal{A}$ needs to respond to the challenge before the message from $\mathcal{Z}$ reaches $\mathcal{A}$. Finally, messages to $\mathcal{Z}$ from $\mathcal{A}$ in a rewinding are ignored by $\mathcal{A}'$. Finally we need to argue that $\mathcal{A}'$ runs in polynomial time. Let $q(n)$ be the expected time that $\mathcal{A}'$ spends to extract the witness. Let p be the probability that the receiver is not corrupted during the rewinding and $\mathcal{A}$ responds to a challenge in the WHPOK of the puzzle before the receiver times out. Then the expected number of times that $\mathcal{A}'$ has to rewind before $\mathcal{A}$ responds to the challenge before the receiver times out (conditioned that the receiver is not corrupted) is $\frac{1}{p}$. Therefore, the total time spent is $p \cdot \frac{1}{p} \cdot q(n)$, which is polynomial.

E.9 Adaptive UC in the Tamper-Proof Hardware Model

The tamper-proof hardware model introduces a physical assumption that enables protocols to be executed in an isolated environment. This assumption is instantiated through the existence of tamper-proof hardware tokens, which allows a party P_i to create a hardware token that implements a functionality F and give this token to any party P_j. Now the party P_j can interact with the token and access the embedded functionality in a black-box manner. The tamper-proof property means that an adversary that has a token can do nothing more than observe the input and output from the interaction with it, i.e. he cannot alter in anyway the functionality that the token implements. The notion of a tamper-proof hardware token in formalized though the ideal functionality $\mathcal{F}_{wrap}$ in Figure 16 introduced by Katz [33].

The following theorem states our result in the tamper-proof model.

THEOREM 12. *Assume the existence of simulateable PKE and an* EQNMCom *scheme. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ with adaptive UC-security in the $\mathcal{F}_{wrap}$-hybrid model.*

Functionality $\mathcal{F}_{wrap}$.

Let p be a polynomial and n be a security parameter for $\mathcal{F}_{wrap}$.

Create: On input $(create, sid, P_i, P_j, M)$ from P_i, where P_j is another user of the system and M is an interactive Turing machine, do:

1. Send $(create, sid, P_i, P_j, M)$ to P_j.
2. If there is no tuple of the form $(P_i, P_j, *, *, *)$ stored, then store $(P_i, P_j, M, 0,)$.

Execute: On input (run, sid, P, msg) from P', find the unique stored tuple $(P, P', M, i, state)$ (if no such tuple exists, then do nothing). Then do:

Case 1 $(i = 0)$: Choose random $w \leftarrow \{0,1\}^{p(k)}$. Run $M(msg; w)$ for at most $p(k)$ steps, and let out be the response (set $out = \perp$ if M does not respond in the allotted time). Send (sid, P, out) to P'. Store $(P, P', M, 1, (msg, w))$ and erase $(P, P', M, i, state)$.

Case 1 $(i = 1)$: Parse state as (msg_1, w). Run $M(msg_1||msg; w)$ for at most $p(k)$ steps, and let out be the response (set $out = \perp$ if M does not respond in the allotted time). Send (sid, P, out) to P'. Store $(P, P', M, 0,)$ and erase $(P, P', M, i, state)$.

Figure 16: $\mathcal{F}_{wrap}$

In order to prove the theorem it suffices to construct an adaptive UC puzzle in the $\mathcal{F}_{wrap}$-hybrid model. Unlike the other puzzles this will be a "stateful" puzzle in the sense that a party is required to spawn a subroutine of S at the beginning of the execution and use this subroutine to generate any consecutive puzzle. This routine can keep state across multiple executions and thus the generated puzzle instances are not independent. Figure 17 presents the resulting puzzle in the tamper-proof model.

We argue the soundness and simulatability properties of the puzzle in Figure 17 as follows:

Soundness: It follows from the one-wayness the function f and the witness-hiding property of the protocol.

Simulation: To simulated concurrent puzzle execution with the adversary $\mathcal{A}$ and the environment $\mathcal{Z}$, $\mathcal{A}'$ emulates internally an execution with $\mathcal{A}$ where it acts as $\mathcal{F}_{wrap}$. $\mathcal{A}'$ obtains the message $(create, sid, P_i, P_j, M^*)$ sent by $\mathcal{A}$. Later in a challenge protocol by $\mathcal{A}$ to P_j, $\mathcal{A}'$ extract the witness to a puzzle y by rewinding M^* in the witness-hiding argument-of-knowledge sub-protocol. Since M^* does not receive messages from any other parties other than P_j during the execution (and the rewinding), the extraction can finish in isolation without intervening the adversary $\mathcal{A}$ and the environment $\mathcal{Z}$. If party P_j is corrupted during rewinding, $\mathcal{A}'$ does not have to execute the simulation.

E.10 Adaptive, Partially Isolated Adversaries Model

In this section, we consider a model that incorporates the physical assumption that protocols can be run in a (partially) isolated environment. In particular, we assume that a player P_j can ensure that another player P_i is *partially isolated* for a short portion of the computation. During this time, P_i can only exchange a limited number of bits with the environment but P_j's communication is unrestricted. More specifically, we assume

Protocol $\langle S, R \rangle$:

S proceeds in two phases:

- When it is first spawned and invoked on inputs the identity of the sender P_i and the session id sid, it uniformly picks a string $x \in \{0,1\}^n$, computes its image y through the one-way function f, and stores (y, P_i, sid) as an internal state.
- Later when S is invoked on inputs the identity of the puzzle receiver P_j to challenge P_j, S checks whether this is the first time interacting with party P_j, if so, it "creates" and "gives" P_j a token, which encapsulates the functionality M that gives a witness-hiding argument-of-knowledge of the statement that y is in the image set of f, by sending the message $(create, sid, P_i, P_j, M)$ to $\mathcal{F}_{wrap}$. To actually challenge P_j, S simply sends y as the puzzle to the receiver.

Upon receiving y from the sender, R accesses M via $\mathcal{F}_{wrap}$ as follows: it sends (run, sid, S, ϵ) to $\mathcal{F}_{wrap}$ (ϵ is an empty string), and then receives from M a WHAOK of the statement that y is in the image set of f (forwarded by $\mathcal{F}_{wrap}$).

Relation: $\mathcal{R} = \{(x, y) | y = f(x)\}$

Figure 17: Tamper-Proof Model Puzzle

the existence of some threshold ℓ, such that P_j can prevent P_i from exchanging more than ℓ bits with the environment.

The partially isolated adversaries model was introduced by [19, 20], and formalized as the isolate ideal functionality $\mathcal{F}_{\mathsf{isolate}}$. We recall the formal description of $\mathcal{F}_{\mathsf{isolate}}$ as in [20] in Figure 18.

We obtain an analogue of the result of [20], using our puzzle framework:

THEOREM 13. *Assume the existence of simulatable PKE scheme, and the existence of an* EQNMCom *scheme. Then, for every well-formed ideal functionality $\mathcal{F}$, there exists a protocol π that realizes $\hat{\mathcal{F}}$ in the Adaptive, Partially Isolated Adversaries model.*

To prove the theorem, it suffices to construct a puzzle in the $\mathcal{F}_{\mathsf{isolate}}$-hybrid model. In all the previous models, the puzzle protocols $\langle \mathsf{S}, \mathsf{R} \rangle$ are executed in a "stateless" way, that is, whenever a party intends to challenge (acting as the sender of the puzzle) another, it spawns *independently* a new subroutine of S to generate the puzzle. In this model, we consider a "stateful" puzzle, which requires a party to spawn a subroutine of S at the beginning of its execution, and use this subroutine to generate all the puzzles it needs throughout its lifetime. (Note that the receiver part of the puzzle protocol is still "stateless".) It is stateful in the sense that the subroutine can keep states across multiple invocations, and hence the puzzle instances generated are not independent to each other, but correlated. More precisely, we define the puzzle $\mathsf{P}_{\mathsf{isolate}} = (\langle \mathsf{S}, \mathsf{R} \rangle, \mathcal{R})$ for the $\mathcal{F}_{\mathsf{isolate}}$-hybrid model as follows. The interactive Turing machine S, proceeds in two phases:

- When it is first spawned and invoked on inputs the identity of the sender P_i and the session id sid– called the initialization phase–it uniformly picks a string $x \in \{0,1\}^n$, computes its image y through the one-way function f, and stores (y, P_i, sid) as an internal state.
- Let Π be an ℓ-Isolated Proof of Knowledge Protocol as defined by [19], where parties P_i, P_j interact and P_i proves that it knows a witness w to an NP-statement z. We note that by definition, such a protocol is standalone zero-knowledge and hence, is also *witness-hiding*.

The $\mathcal{F}_{\mathsf{isolate}}$ ideal functionality is parameterized by an isolation paramter ℓ, a security paramter κ and a polynomial p.

Isolation of P_i: Wait until receiving messages $(\mathsf{isolate}, sid, P_i, P_j)$ from P_j and $(\mathsf{isolate}, sid, P_i, P_j, M)$ from P_i. If there is already a stored tuple of the form $(P_i, P_j, \cdot, \cdot, \cdot, \cdot)$ then ignore the command. Otherwise:

1. Parse the string M as the description of an ITM with four communication tapes; two tapes ("in" and "out") for regular protocol communication with P_j and two tapes for secret communication with P_i. Let the value state encode the initial state of M (including the value of a work tape and an initialized random tape). Define new values $\mathsf{inCom} = 0, \mathsf{outCom} = 0$ and store the tuple $(P_i, P_j, M, \mathsf{state}, \mathsf{inCom}, \mathsf{outCom})$.
2. Send $(\mathsf{isolate}, sid, P_i)$ to P_j.

Interaction with P_j: On input $(\mathsf{run}, sid, P_i, P_j, \mathsf{msg})$ from P_j, retrieve the tuple $(P_i, P_j, M, \mathsf{state}, \mathsf{inCom}, \mathsf{outCom})$. If there is no such tuple then ignore the command.

1. Place the string msg on the "in" tape designated for P_i and run M for $p(\kappa)$ steps.
2. If there is any value msg' on the output tape for P_j then send the message $(\mathsf{reply}, sid, P_i, \mathsf{msg}')$ to P_j.
3. If there is any value msg' on the output tape for P_i and $\mathsf{outCom} + |\mathsf{msg}'| < \ell$ then send the message $(\mathsf{secretCom}, sid, P_j, P_i, \mathsf{msg}')$ to P_i and update $\mathsf{outCom} = \mathsf{outCom} + |\mathsf{msg}'|$.
4. Update the value of state in the stored tuple to encode the updated state of M and the values of its tapes.

Communication: On input $(\mathsf{secretCom}, sid, P_i, P_j, \mathsf{msg})$ from P_i, if there is no tuple of the form $(P_i, P_j, M, \mathsf{state}, \mathsf{inCom}, \mathsf{outCom})$ then ignore. Also if the tuple has $\mathsf{inCom} + |\mathsf{msg}| > \ell$ then ignore the command. Otherwise:

1. Update $\mathsf{inCom} = \mathsf{inCom} + |\mathsf{msg}|$, place msg on the "in" tape for P_i and run M for $p(\kappa)$ steps.
2. Proceed with steps $2, 3, 4$ of the above command.

Release of P_i: On input $(\mathsf{release}, sid, P_i, P_j)$ from P_j, retrieve the tuple $(P_i, P_j, M, \mathsf{state}, \mathsf{inCom}, \mathsf{outCom})$ and send $(\mathsf{release}, sid, P_i, P_j, \mathsf{state})$ to P_i.

Figure 18: The $\mathcal{F}_{\mathsf{isolate}}$ Ideal Functionality

P_j, playing the part of receiver, initializes a puzzle interaction with S by sending the message $(\mathsf{isolate}, sid, P_i, P_j)$ to the Ideal Functionality. S replies with the message $(\mathsf{isolate}, sid, P_i, P_j, M)$, where M is a description of an ITM playing the part of the Prover in protocol Π, interacting via protocol communication with verifier P_j and via secret communication with P_i. The NP-statement being proved is simply that y is in the range of f, and by the end of the protocol, P_j should be convinced that P_i knows x such that $f(x) = y$.

S and P_j interact with M via the Ideal Functionality messages run and $\mathsf{secretCom}$. When the protocol completes, P_j sends a message $(\mathsf{release}, sid, P_i, P_j)$ to the Ideal Functionality.

To actually challenge P_j, S simply sends y as the puzzle to the receiver. The puzzle relation $\mathcal{R}$ is simply $\{(x, y) \mid y = f(x)\}$.

The soundness of the puzzle follows directly from the one-wayness of the function f and the witness-hiding property of the protocol. Furthermore, to adaptively simulate a concurrent puzzle interaction with adversary $\mathcal{A}$ and environment $\mathcal{Z}$, $\mathcal{A}'$ internally emulates an execution with $\mathcal{A}$ and acts as the $\mathcal{F}_{\mathsf{isolate}}$ functionality for $\mathcal{A}$. Whenever $\mathcal{A}$ sends a message $(\mathsf{isolate}, sid, P_i, P_j, M)$ to $\mathcal{F}_{\mathsf{isolate}}$, $\mathcal{A}'$ obtains the message. Later to extract the witness of a puzzle y challenged by $\mathcal{A}$ (controlling P_i) to P_j, $\mathcal{A}'$ simply runs the knowledge extractor of the ℓ-Isolated Proof of Knowledge to extract the witness. Using the [19] construction of ℓ-Isolated Proofs of Knowledge, we have that the simulation of $\mathcal{A}'$ is perfect; addtionally, we note that since the [19] verifier is public-coin, dealing with adaptive corruptions is trivial. Thus, we achieve perfect, adaptive simulation.

F Constructing Non-Interactive, Language-Based, Equivocal Commitments

Let Com be a non-interactive commitment scheme with a pseudorandom range. Such a commitment scheme can be constructed from OWF.

Let $\mathcal{L}$ be an NP-Language and $\mathcal{R}$, the associated NP-relation. Since the language $\mathcal{L} \in \mathsf{NP}$, we can reduce $\mathcal{L}$ to the NP-complete problem Hamiltonian Cycle. Thus, given the public input x (which may or may not be in $\mathcal{L}$), we can use a (deterministic) Karp reduction to a graph G which contains a Hamiltonian cycle. Moreover, finding a Hamiltonian cycle H in the graph G, implies finding a trapdoor w such that $\mathcal{R}(x, w) = 1$. Let Φ denote the deterministic mapping from strings x to a graphs G induced by the Karp reduction.

The protocol is specified in Figures 19, 20 and has appeared before in [11]. For completeness, we present it again here and show that it satisfies the properties of an equivocal commitment scheme as specified in Definition 2

We omit the security analysis of the non-interactive, language-based equivocal commitment scheme $\langle \mathsf{S_{eq}}, \mathsf{R_{eq}} \rangle$ presented in Figures 19 and 20, since it is standard.

G Constructing Adaptively-secure WIPOK

The adaptively-secure (without erasures) WIPOK construction given here is similar to the one given in [38]. As in [38], it is based on Blum's Σ-protocol for graph Hamiltonicity [5]. Let Com be any commitment scheme. The Σ-protocol proceeds as follows (see figure 21):

We construct adaptively-secure WIPOK by replacing each commitment Com in the Σ-protocol with a non-interactive equivocal commitment $Com^*(\pi(G')_{i,j})$, as constructed above.

Lemma 8. *When commitments Com are replaced with equivocal commitments Com^* generated by running the protocol $\langle \mathsf{S_{eq}}, \mathsf{R_{eq}} \rangle$ presented in Figures 19 and 20 then we have that the protocol in Figure 21 is a WIPOK (with soundness $1/2$) and is secure under adaptive corruptions.*

$\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ **on common input** x **and private input** β**: Commitment phase**

To commit to $\beta = 1$**:**

1. S_{eq} chooses an $n \times n$ adjacency matrix H of a random n-node Hamiltonian cycle.
2. S_{eq} sends a matrix $\overline{\mathsf{Com}}$ of $n \times n$ strings where the following holds:
   - $\overline{\mathsf{Com}}_{i,j}$ contains a random commitment to 1 under Com iff $H_{i,j} = 1$.
   - $\overline{\mathsf{Com}}_{i,j}$ contains a random string iff $H_{i,j} = 0$.

To commit to $\beta = 0$**:**

1. S_{eq} chooses an $n \times n$ adjacency matrix I which corresponds to a random isomorphism of $G = \Phi(x)$.
2. S_{eq} sends a matrix $\overline{\mathsf{Com}}$ of $n \times n$ strings where the following holds:
   - $\overline{\mathsf{Com}}_{i,j}$ contains a random commitment to 1 under Com iff $I_{i,j} = 1$.
   - $\overline{\mathsf{Com}}_{i,j}$ contains a random commitment to 0 under Com iff $I_{i,j} = 0$.

Let $C = \mathsf{EQCom}^x(\beta; r)$ denote the transcript of the commit phase when S_{eq} uses randomness r.

$\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ **on common input** x**: Decommitment phase**

To decommit to a 0**:**

1. S_{eq} opens the commitments in $\overline{\mathsf{Com}}$ where $\overline{\mathsf{Com}}_{i,j}$ is a commitment to 1 and shows that these correspond to a random Hamiltonian cycle.
2. S_{eq} produces the randomness it used to sample the remaining random strings in the matrix $\overline{\mathsf{Com}}$.

To decommit to a 1**:**

1. S_{eq} opens the commitments in $\overline{\mathsf{Com}}$ to obtain adjacency matrix I and shows an isomorphism from $G = \Phi(x)$ to this graph.

Figure 19: Non-interactive, language-based equivocal commitment scheme $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$

sketch. The analysis of the soundness of the protocol follows from the analysis of the underlying Σ-protocol, which we omit since it is by now a standard argument.

Next we need to prove the witness-indistinguishability and proof of knowledge properties as well as the fact that the protocol is secure under adaptive corruptions. In fact, we show that the above construction is not only a WIPOK, but is a Zero Knowledge Proof of Knowledge. We now present a simulator which satisfies the zero-knowledge property and can also handle *adaptive* corruptions (for simplicity, we consider here only post-execution corruptions). This implies that the scheme above is zero-knowledge as well as secure under *adaptive corruptions*.

On input graph G', the Simulator does the following:

Simulation of Prover's first message: Use the simulator for $\langle \mathsf{S}_{\mathsf{eq}}, \mathsf{R}_{\mathsf{eq}} \rangle$ to compute a commitment for each position in an $n \times n$ matrix (each position in the matrix can now be opened to either 0 or 1).

Simulation of Prover's second message

$\langle \tilde{\mathsf{S}}_{\mathsf{eq}}, R_{\mathsf{eq}} \rangle$ **on common input** $x \in L$ **and private input** w **where** $w \in \mathcal{R}(x)$**: Equivocal Commitment**

1. $\tilde{\mathsf{S}}_{\mathsf{eq}}$ chooses an $n \times n$ adjacency matrix I which corresponds to a random isomorphism of $G = \Phi(x)$.
2. $\tilde{\mathsf{S}}_{\mathsf{eq}}$ sends a matrix $\overline{\mathsf{Com}}$ of $n \times n$ strings where the following holds:
   - $\overline{\mathsf{Com}}_{i,j}$ contains a random commitment to 1 under Com iff $I_{i,j} = 1$.
   - $\overline{\mathsf{Com}}_{i,j}$ contains a random commitment to 0 under Com iff $I_{i,j} = 0$.

Let $C = \mathsf{EQCom}^{*x}(r)$ denote the transcript of the commit phase when $\tilde{\mathsf{S}}_{\mathsf{eq}}$ uses randomness r.

$\mathsf{Adap}_{\mathsf{eq}}(x, w, r, \tau, v)$**, where** τ **is the transcript generated by** $\langle \tilde{\mathsf{S}}_{\mathsf{eq}}, R_{\mathsf{eq}} \rangle$ **on common input** $x \in L$**: Equivocal Decommitment**

$\mathsf{Adap}_{\mathsf{eq}}$ **decommits to** $v = 0$ **as follows:**

1. $\mathsf{Adap}_{\mathsf{eq}}$ opens all the commitments in $\overline{\mathsf{Com}}$ to reveal adjacency matrix I and shows an isomorphism from $G = \Phi(x)$ to this graph.

$\mathsf{Adap}_{\mathsf{eq}}$ **decommits to** $v = 1$ **as follows:**

1. $\mathsf{Adap}_{\mathsf{eq}}$ uses w to open the commitments in $\overline{\mathsf{Com}}$ that correspond to the Hamiltonian cycle in $G = \Phi(x)$ and shows that these correspond to a random Hamiltonian cycle.
2. $\mathsf{Adap}_{\mathsf{eq}}$ produces random coins for sampling the remaining strings in $\overline{\mathsf{Com}}$ at random.

Figure 20: Non-interactive, language-based equivocal commitment scheme–Equivocator $(\tilde{\mathsf{S}}_{\mathsf{eq}}, \mathsf{Adap}_{\mathsf{eq}})$

- If $b = 0$, choose a random permutation π and equivocally open the commitments of the $n \times n$ matrix to be consistent with $\pi(G')$.
- If $b = 1$, choose a random cycle C and equivocally open the commitments that correspond to the Hamiltonian cycle to be consistent with the cycle.

Upon post-execution corruption of Prover: Upon corruption, the simulator learns the witness, the cycle H of graph G'.

- If $b = 0$, all the commitments have already been opened, the permutation π has been revealed and there is no additional information revealed to the adversary upon corruption.
- If $b = 1$, find some permutation π' of the vertices of G' such that $\pi'(H) = C$. Note that since H and C are simply n-node cycles, finding such a π' takes linear time. Equivocally open the commitments of the remaining entries of the $n \times n$ matrix to be consistent with $\pi'(G')$.

We omit the analysis of the above simulator. It is straightforward to check that the simulator simultaneously satisfies the zero-knowledge property and also simulates adaptive corruptions successfully.

We additionally omit the proof that the protocol is a proof of knowledge, which is also straightforward. □

Σ Protocol

Prover's input: Graph G' (we also use the notation G' to represent the adjacency matrix of G') with Hamiltonian cycle H.

Prover's first message:

- Choose a permutation π of the vertices of G'.
- Commit to the adjacency matrix of $\pi(G')$ by sending $[Com(\pi(G')_{i,j})]_{1 \leq i \leq n, 1 \leq j \leq n}$ to the Verifier.

Verifier's message: Verifier chooses $b \in \{0, 1\}$ at random and sends to Prover.

Prover's second message:

- If $b = 0$, reveal π and open the commitments of the entire adjacency matrix.
- If $b = 1$, reveal only the cycle $\pi(H)$ in $\pi(G')$ by opening the commitments that correspond to the Hamiltonian cycle.

Verifier checks the following:

- If $b = 0$, do the following: Given π, check that the opened adjacency matrix is equal to $\pi(G')$. Check that each of the commitments was opened correctly.
- If $b = 1$, check that the opened commitments correspond to a Hamiltonian cycle. Check that each of the commitments was opened correctly.

Figure 21: Σ Protocol