Environmentally Friendly Composable Multi-party Computation in the Plain Model from Standard (Timed) Assumptions

Brandon Broadnax[1], Jeremias Mechler[2](✉), and Jörn Müller-Quade[2]

[1] Robert Bosch GmbH, Stuttgart, Germany
broadnax@ira.uka.de
[2] KASTEL, Karlsruhe Institute of Technology, Karlsruhe, Germany
{mechler,mueller-quade}@kit.edu

Abstract. Starting with the work of Rivest et al. in 1996, timed assumptions have found many applications in cryptography, building e.g. the foundation of the blockchain technology. They also have been used in the context of classical MPC, e.g. to enable fairness. We follow this line of research to obtain composable general MPC in the plain model.

This approach comes with a major advantage regarding *environmental friendliness*, a property coined by Canetti et al. (FOCS 2013). Informally, this means that our constructions do not "hurt" game-based security properties of protocols that hold against polynomial-time adversaries when executed alone.

As an additional property, our constructions can be plugged into any UC-secure protocol without loss of security.

Towards proving the security of our constructions, we introduce a variant of the UC security notion that captures timed cryptographic assumptions. Combining standard timed commitment schemes and standard polynomial-time hardness assumptions, we construct a composable commitment scheme in the plain model. As this construction is constant-round and black-box, we obtain the *first fully* environmentally friendly composable constant-round black-box general MPC protocol in the plain model from standard (timed) assumptions.

1 Introduction

In order to achieve the very strong notion of universally composable (UC) security [Can01], trusted setups are required [CF01]. However, in practice, trusted setups are often hard to come by. Therefore, a long line of research (e.g. [Pas03,BS05,LPV09,Gar+12,GKP18,Dac+13,PS04,CLP10, CLP13,Bro+17]) has investigated how composable multi-party computation (MPC) can be achieved in the plain model, i.e. only assuming authenticated communication.

For the full version [BMM21], see https://eprint.iacr.org/2021/843.

K. Nissim and B. Waters (Eds.): TCC 2021, LNCS 13042, pp. 750–781, 2021.
https://doi.org/10.1007/978-3-030-90459-3_25

Common to their techniques is that the simulation is *environmentally unfriendly*, i.e. "hurts" the security of protocols that run along-side and that rely on polynomial-time hardness assumptions.

Formally, this is captured by the notion of *environmental friendliness* as defined by Canetti, Lin, and Pass [CLP13], which considers all game-based security properties of a protocol against polynomial-time adversaries.

The typical reason for limited environmental friendliness is a super-polynomial simulation, which can break polynomial-time assumptions used in other protocols, therefore impacting their security properties. This holds even if the super-polynomial resources are restricted by e.g. an angel.

However, super-polynomial simulation techniques are not the only danger to the security of other protocols: Non-uniform advice given to the simulator (e.g. as in [LPV09]) may impact the security of previously started protocols—even if they are concurrently composable and secure against non-uniform adversaries. This additional property is not considered by the definition of environmental friendliness.

Ever since composable MPC in the plain model has been investigated, the following question has been left unanswered:

Can we achieve composable MPC in the plain model that is friendly to protocols that are executed along-side and may have started previously?

Previous results suggest that a simulation technique that runs in polynomial-time and does not rely on non-uniform advice is needed. Such a simulation cannot be achieved, in principle, even by previous advanced approaches like Angel-based security or shielded oracles. Therefore, new techniques to overcome the impossibility results of UC security are needed.

With the advent of the blockchain era, timed cryptographic assumptions have seen widespread use in the real world. A very popular example is the *proof of work* protocol of the Bitcoin blockchain. Even though its hardness is not based on some well-understood cryptographic assumption, it has proven to work nevertheless for many years.

Timed variants of classic cryptographic primitives such as commitment schemes can be constructed from timed assumptions that are inspired by well-understood standard assumptions. Rivest, Shamir, and Wagner [RSW96] have initiated this study and proposed a time-lock puzzle based on the hardness of factoring and the time required to square modulo a composite. Based on such assumptions, timed cryptographic primitives such as time-lock puzzles and timed-release encryption [RSW96] or timed signatures and timed commitment schemes [BN00] can be constructed in the plain model. More recently, stronger primitives such as non-malleable time-lock puzzles and commitment schemes have been constructed [Eph+20,KLX20] using a setup.

As timed assumptions and primitives can be broken in polynomial-time by definition, they seem destined to solve the problem of limited friendliness exhibited by previous approaches for composable MPC in the plain model. In the following, we thus investigate the following questions:

Can we use timed assumptions to achieve composable MPC in the plain model? What are the advantages and disadvantages of such an approach?

We answer the first question affirmatively and propose a new approach for general MPC in the plain model based on asymmetries that are only temporary and much smaller compared to previous approaches. Namely, these asymmetries consist of only a polynomial number of computation steps sufficient to leverage timed cryptographic assumptions. The very feasibility of this approach may seem surprising as timed cryptographic primitives eventually lose their security. For example, timed commitments will eventually leak their secret by definition. Previous constructions crucially rely on this not to happen, i.e. the complexity asymmetry and the ensuing security to hold throughout the whole execution. We side-step this problem by using timed assumptions to merely set up short-lived trapdoors that can only be used while the assumptions still hold. After their security has expired, the (now possibly leaked) trapdoor is useless for the adversary. Yet, a simulator can use it to establish a long-lived trapdoor based on some classical polynomial-time assumption.

We introduce the notion of TLUC ("time-lock UC") security, which is based on UC security and cast in the unmodified UC framework. With TLUC, honest parties may set up timers with some timeout $\ell \in \mathbb{N}$ that expire when all entities have spent more than ℓ steps in total. This allows to capture the security of (stand-alone) timed primitives such as time-lock puzzles or timed commitment schemes. While computations performed by protocol parties, environment and adversary are counted against timers, computations performed by the simulator are not. This allows simulators to break timed assumptions "at no cost" in terms of time accounting, while remaining polynomially bounded. Such a simulator can then, for example, extract a timed commitment while it is still hiding for environment and adversary.

With respect to the question of environmental friendliness, it suffices to see that the notion of TLUC security is a meaningful special case of UC security, which is fully environmentally friendly. This already implies that our notion also features full environmental friendliness as defined by [CLP13].

In order to be friendly to previously started protocols, a uniform simulation, i.e. one that does not rely on non-uniform advice, is needed. Looking ahead, this is indeed the case for our composable commitment scheme.

To the best of our knowledge, we are the first to achieve both of these properties simultaneously.

Leveraging timed assumptions for composability comes with a number of additional advantages. Namely, our notion is UC-compatible in the sense that if π UC-emulates ϕ for arbitrary protocols π and ϕ, then π also TLUC-emulates ϕ. TLUC security allows the reuse of UC protocols in the sense that one can take a UC-secure protocol ρ making one subroutine call to $\mathcal{F}$ that UC-realizes some ideal functionality $\mathcal{G}$ and replace $\mathcal{F}$ with its TLUC realization π. The composite protocol ρ^{π} is then guaranteed to TLUC-realize $\mathcal{G}$. These properties are not generally offered in full by other notions that allow composable general MPC in the plain model and are not implied by (limited) environmental friendliness. What is more, TLUC security is meaningful for ideal functionalities that rely on (even uniform) polynomial-time assumptions. This is in contrast to e.g. SPS security, where such functionalities are affected by the super-polynomial simulator.

Unfortunately, TLUC security is not closed under composition. Thus, one has to manually prove that multiple instances of π TLUC-realize multiple instances of $\mathcal{F}$ (i.e. $\hat{\pi}$ TLUC-realizes $\hat{\mathcal{F}}$).

Like previous approaches for general MPC in the plain model and even UC security, TLUC security is not friendly to *timed* game-based properties of other protocols, e.g. the timed hiding property of a timed commitment scheme. This property is neither captured by the definition of environmental friendliness nor fulfilled by any previous notion that allows composable MPC—not even UC security.

Towards realizing composable general MPC, we first construct a commitment scheme that TLUC-realizes the ideal functionality for multiple commitments $\mathcal{F}_{\text{MCOM}}$. In more detail, we combine a (possibly malleable) timed commitment with a non-malleable commitment to construct a commitment that is equivocal and concurrently simulation-sound, i.e. retains its binding property even if the adversary sees equivocated commitments. We show that this suffices to replace the CRS of the UC-secure commitment scheme of Canetti and Fischlin [CF01] with coin-tosses, assuming that trapdoor one-way permutations with dense public description [DP92] exist. The resulting composable commitment scheme is constant-round, black-box, in the plain model and makes use of standard polynomial-time and standard timed assumptions only. We note that our approach is conceptually different from recent results [Eph+20, KLX20, Bau+21, Bau+20] which define non-malleable or composable timed primitives and realize them using a trusted setup.

Due to the reusability of UC protocols, we can plug our construction into any UC protocol in the $\mathcal{F}_{\text{MCOM}}$-hybrid model while maintaining TLUC security. Using e.g. a variant of the MPC protocol of Hazay and Venkitasubramaniam [HV15], we are the first to obtain a composable constant-round, black-box and environmentally friendly general MPC protocol from standard polynomial-time and timed assumptions that does not impact the security of other protocols relying on (non-timed) polynomial-time hardness assumptions.

1.1 Related Work

Towards achieving composable MPC in the plain model, a number of approaches have been proposed.

SPS Security, introduced by [Pas03], considers simulators that may have a super-polynomial run-time, giving them an advantage over the polynomially-bounded environment at the expense of environmental friendliness and UC reusability.

While earlier approaches such as [Pas03,BS05] require (non-standard) super-polynomial hardness assumptions, newer approaches such as [LPV09,Gar+12, GKP18] require only standard polynomial-time hardness assumptions.

Due to the complexity asymmetry between environment and simulator, these constructions do not offer general composition. The transitivity of SPS security holds only with respect to protocols whose security is not "hurt" by the stronger simulator, e.g. protocols that are information-theoretically secure such as [IPS08]. Thus, (general) reusability of UC protocols is lost.

[LPV09] have generalized the notion of UC security to $(\mathcal{C}_{\mathsf{env}}, \mathcal{C}_{\mathsf{sim}})$-security, where $\mathcal{C}_{\mathsf{env}}$ and $\mathcal{C}_{\mathsf{sim}}$ denote the complexity classes of environment resp. simulator. They present a construction for non-malleable zero-knowledge from UC puzzles that can be plugged into an appropriate general MPC protocol. For their construction in the plain model, [LPV09] assume simulators that run in non-uniform polynomial-time while the environment runs in uniform polynomial-time. However, the non-uniform simulation may impact the security of protocols that have started in the past. Also, if $\mathcal{C}_{\mathsf{sim}}$ is non-uniform polynomial-time, then the security notion is not meaningful for ideal functionalities that rely on uniform polynomial-time hardness assumptions.

[Dac+13] have extended the work of [LPV09] by considering adaptive security. Starting with a UC puzzle, they construct a commitment scheme satisfying their new and strong notion of non-malleability from simulatable public-key encryption. This non-black-box and non-constant-round construction can then be plugged into an appropriate protocol, yielding adaptively secure composable general MPC.

Recently, [GKP18] have presented a SPS-secure black-box OT protocol from constant-round semi-honest OT and collision-resistant hash functions, i.e. standard polynomial-time hardness assumptions only. Their construction is secure against static corruptions and has a lower round complexity than other constant-round constructions such as [Bro+17].

Angel-Based Security and Environmental Friendliness. The weak composition properties of SPS security have subsequently been improved upon by notions where the simulator itself remains polynomially bounded, but is aided by some super-polynomial entity that is also available to the environment. Such frameworks include *Angel-based security* [PS04], or *UC with super-polynomial helpers* [CLP10]. [CLP10] construct a non-constant-round CCA commitment scheme from one-way functions and use it to realize the ideal functionality for commitments. Their construction can be plugged into any constant-round UC protocol ρ in the $\mathcal{F}_{\mathrm{COM}}$-hybrid model without losing security. This property, called *round robustness*, has been generalized by [CLP13] to the property of *environmental friendliness*. The helper of [CLP13] is environmentally friendly for protocols whose security is proven via black-box reductions to game-based cryptographic hardness assumptions with bounded polynomial round complexity.

Shielded Oracles. [Bro+17] have introduced the notion of UC security with *shielded oracles* that strictly lies between SPS security and Angel-based security.

Their construction for a composable commitment scheme makes use of standard polynomial-time hardness assumptions only, is constant-round and black-box. While their notion is not environmentally friendly, they showed that the constructions can be plugged into a special class of UC-secure protocols without loss of security.

Other Models and Notions. There have been proposed a number of different models which enable (composable) MPC in the plain model. The *timing model* introduced by [KLP05] considers a communication network with time bounds and parties that have access to a local clock with little drift. There, non-constant-round non-black-box MPC secure under general composition is possible. This is done by *delaying* other protocols that are executed concurrently and incomparable to our approach.

The notion of *input indistinguishability*, first defined by [MPR06] and generalized and strengthened by [Gar+12], is another security notion capturing concurrent self-composition that can be achieved in the plain model. However, the constructions of [MPR06,Gar+12] are non-black-box. Also, input indistinguishability is weaker than UC security.

Non-malleable Time-Lock Puzzles and Commitments. [Eph+20] have introduced the notion of *non-malleable time-lock puzzles* and timed commitments and present constructions in the random oracle model. Similar results have been obtained by [KLX20] in the algebraic group model. While both results can possibly be used as building blocks in our constructions, they are not in the plain model.

TARDIS and CRAFT. TARDIS [Bau+21] extends the GUC framework [Can+07] to include a notion of *abstract time* and *ticked functionalities* whose behavior can depend on the elapsed time. In this setting, universally composable abstractions of time-lock puzzles can be defined and realized in the random oracle model. We note that the goal of [Bau+21] is different than ours. We use stand-alone-secure and possibly *malleable* timed primitives such as (malleable) timed commitments in order to achieve composability in the plain model. In contrast to TARDIS, we do not aim to define composable security notions for timed primitives. CRAFT [Bau+20] realizes composable MPC in the TARDIS framework with additional guarantees such as output-independent abort, also relying on a random oracle.

1.2 Our Results

New Security Notion for Composable Security. The notion of UC security considers entities that are polynomially bounded and inherently unaware of other computations going on. Thus, timed assumptions cannot be properly used in UC protocols. With TLUC security, we consider a variant of UC security that allows a party P to set up *timers* associated with a number of steps ℓ. At any point, P may query if the execution experiment in total (including the environment, adversary and other protocol parties) has performed ℓ or more steps. This allows the use of timed cryptographic primitives such as timed commitments.

Similar to SPS security, our security notion is not closed under composition and features the single-instance composition theorem only (Theorem 2).

Environmental Friendliness. Very informally, *environmental friendliness*, introduced by Canetti, Lin, and Pass [CLP13], deals with the problem of negative "side-effects" a protocol π may have on game-based properties of another protocol π' that runs *along-side* (where neither protocol is a subroutine of the other) and relies on polynomial-time hardness assumptions. Formally, this is captured in a stand-alone model for game-based security properties. Previous notions that feature general MPC in the plain model suffer from limited environmental friendliness because super-polynomial simulation, e.g. due to use of a super-polynomial helper, may break polynomial-time hardness assumptions of other protocols that run along-side, resulting in limited environmental friendliness. While not considered by the definition of environmental friendliness, giving the simulator non-uniform advice may hurt the security of (even non-uniformly) secure protocols or protocols that have been previously executed. Being a special case of UC security, TLUC security is fully environmentally friendly (Proposition 5).

We note that the established notion does not consider *timed* game-based properties such as the timed hiding property of a timed commitment scheme. As such, our notion as well as *all* previous notions such as e.g. SPS security, Angel-based security and even UC security are not fully friendly in this respect.

UC Compatibility and Reusability. As *all* UC protocols retain their security under our notion (UC compatibility, Proposition 3) and TLUC simulators run in strict polynomial-time, we can realize a UC-complete functionality $\mathcal{F}$ in TLUC and plug it into *any* existing UC-secure protocol making one subroutine call to $\mathcal{F}$ without loss of security (UC reusability, Corollary 3). This is not implied by environmental friendliness *per se*. As the simulation is always polynomial-time, (even uniformly only) computationally secure ideal functionalities are meaningful in our framework.

Composable Commitment Scheme in the Plain Model. Combining a timed commitment scheme and a pCCA-secure commitment scheme, we construct a non-malleable and partially simulatable coin-toss that is sufficient to "bootstrap" the CRS of a UC-secure commitment scheme such as the $\mathsf{UCC}_{\mathsf{OneTime}}$ scheme of Canetti and Fischlin [CF01] in the plain model. The resulting commitment scheme is concurrently composable and TLUC-realizes the ideal functionality for multiple commitments $\mathcal{F}_{\mathrm{MCOM}}$ (Theorem 4). As the simulation is uniform, π_{MCOM} does not hurt the security of any protocol making use of polynomial-time assumptions, including uniform ones.

Composable Constant-Round General MPC in the Plain Model. Plugging our construction for $\mathcal{F}_{\mathrm{MCOM}}$ into a variant of the general MPC protocol due to

[HV15], we obtain a constant-round black-box and environmentally friendly general MPC protocol from standard polynomial and standard timed assumptions in the plain model (Theorem 5). We remark that our results are in the static corruption setting.

1.3 Outline

We first cover important definition and technical aspects in the preliminaries (Sect. 2). In Sect. 3, we introduce the notion of *timed simulation-soundness* for commitment schemes and present a construction. We continue with a short introduction into TLUC security (Sect. 4), which is a variant of UC security that captures timed assumptions and fulfilled by our composable commitment scheme in the plain model (Sect. 5). Finally, we show how we can use this commitment scheme to achieve composable general MPC in Sect. 6. For details, we refer the reader to the full version [BMM21].

2 Preliminaries

2.1 Notation

Let $n \in \mathbb{N}$. Then, $[n]$ denotes the set $\{1, \ldots, n\}$. Let H_i be some hybrid. Then out_i denotes the output of H_i. $\mathsf{negl}(\kappa)$ denotes an unspecified negligible function in the security parameter $\kappa \in \mathbb{N}$. $x \overset{\$}{\leftarrow} Y$ denotes that x is drawn uniformly at random from the set Y. $x \leftarrow Y$ denotes that x is either the output of the probabilistic algorithm Y or sampled according to the probability distribution Y. Let π_1, π_2 be protocols. Then, $\pi_1 \geq_{\mathrm{UC}} \pi_2$ denotes that π_1 UC-emulates π_2 and $\pi_1^{\pi_2}$ denotes that π_1 makes at least one subroutine call to π_2.

2.2 Machine Model, Notion of Time

When considering polynomial-time hardness assumptions, the particularities of machine models rarely matter. This is because different (classical) machine models can be usually emulated by each other with polynomial run-time overhead or speedup. With polynomial-time being closed under addition and multiplication, polynomial-time hardness assumptions do not become insecure if there is a machine model where some problem can be solved (polynomially) more efficient.

In this paper, we consider timed primitives such as timed commitment schemes. For timed primitives, security often is only guaranteed against adversaries adhering to some kind of (concrete) run-time bound in a fixed machine model. For such assumptions, changing the machine model can make the difference between security and insecurity. This is obvious for stark differences, e.g. when going from a sequential to a parallel machine model when considering timed assumptions that hold only against sequential adversaries. However, this problem also manifests with more subtle changes like allowing a larger alphabet for Turing machines, which may result in a linear speedup.

More problems arise during security reductions that require the emulation of Turing machines. Suppose that we want to show the security of some protocol π by using a ℓ-bounded timed assumption. We call ℓ the *timed security parameter*. In the security proof, the adversary $\mathcal{A}'$ against the timed assumption has to internally emulate the ℓ-bounded adversary $\mathcal{A}$ as well as (parts of) the protocol π. Just internally emulating the ℓ-bounded adversary may incur an overhead that does not allow the reduction to go through, because $\mathcal{A}'$ may always require more than ℓ steps due to its emulation overhead, even when just running the code of $\mathcal{A}$ and relaying messages. Additional overhead may occur e.g. for extracting the correct answer based on the internally emulated adversary's output. These caveats have to be accounted for.

Later on, we use timed primitives in the UC framework (cf. Sect. 4). While UC security can be stated using various machine models [Can01], we adhere to the standard model of interactive Turing machines. However, as e.g. the particular alphabet or the number of work tapes is left unspecified[1], so is the exact notion of run-time in that particular model. In order to argue about the security of timed assumptions in our security notion, we thus have to map the underspecified notion of run-time of interactive Turing machines as defined in the UC framework to the (possibly also underspecified) notion of run-time for the timed assumption. Following the Cobham-Edmonds thesis (see e.g. [Gol08]) or the extended Church-Turing thesis, we assume that this is always possible with a polynomial overhead or speedup in a classical setting, i.e. when not considering quantum computations.

For common machine models such as Turing machines, Boolean circuits or (parallel) random access machines, explicit emulation constructions and bounds for the overhead resp. speedup are known.

When constructing a protocol with security against $\ell(\kappa)$-bounded adversaries, we thus require the timed building blocks to be secure against adversaries with timed security parameter $\ell'(\ell(\kappa), \kappa)$[2] where ℓ' is a sufficiently large polynomial that accounts for possible run-time mismatches due to emulation overhead, reduction overhead or (polynomial) efficiency changes between machine models. As we do not want to make assumptions about the machine models being used, we do not explicitly specify ℓ'. However, as soon as all machine models and reductions are fixed, ℓ' is well-defined. Also, for our constructions, we show that ℓ' is sufficiently generic and e.g. is independent of the TLUC environment under consideration.

Note that the timed security parameter generally grows with increasing protocol nesting depth, similar to the tightness loss in standard reductions.

In our protocols, we use `timer` messages parameterized by an ID *id* to allow protocol parties later check if more steps than allowed by the timed security parameter ℓ have been elapsed by sending a message (`notify`, *id*). If the answer

[1] Newer versions of the UC framework such as UC2020 explicitly allow multiple work tapes, allowing the emulation of other Turing machines with only additive overhead.

[2] In order to capture the setting where $\ell(\kappa)$ is constant but e.g. the reduction overhead depends on κ, we parameterize ℓ' with both values.

is $(\texttt{notify}, id, 1)$, then more than ℓ steps have passed and we say that the "timer has timed out" or "expired". Conversely, $(\texttt{notify}, id, 0)$ denotes that the timer has not expired. Later on, we will only consider adversaries (or environments) that handle such messages correctly.

As the default machine model and execution experiment of UC are inherently sequential, we refer to *computation steps* instead of *run-time*, as the latter may capture many steps performed in parallel, which we want to count individually.

2.3 Timed Commitment Schemes

Boneh and Naor [BN00] have introduced the notion of *timed commitment schemes*. Instead of the hiding property holding against all polynomial-time adversaries, a (T, ℓ, ε)-timed commitment scheme guarantees the hiding property to hold only for some bound of steps ℓ performed by an adversarial receiver, except with probability ε.

However, the (ℓ, ε)-hiding property does not guarantee that there exists a value $T \in \mathbb{N}$ such that a valid timed commitment can be opened "forcefully" in at most $T > \ell$ steps. To this end, the definition of [BN00] also requires the existence of a `forced-open` algorithm that runs in time T, takes the transcript of a successful commit phase and outputs the unique value $v \in M$ committed to, where M is the message space of the commitment scheme. In other words, in addition to the binding property, a malicious committer must not be able to open its commitment to a value that is inconsistent with the output of the `forced-open` algorithm. This extractability is crucial for our simulation later on, as it guarantees that simulators can extract timed commitments in polynomial time (if T is bounded by a polynomial in κ).

In the definition of [BN00], timed commitment schemes have to exhibit a *soundness property* which requires that at the end of the commit phase, the receiver is "convinced" that running the `forced-open` algorithm will produce the value v committed to. While not formally defined, the definition of [BN00] also requires valid commitments to be efficiently recognizable by the receiver.

Looking ahead to our construction, we do not need valid timed commitments to be efficiently recognizable. In particular, we can deal with the over-extraction of invalid commitments, i.e. the case where `forced-open` outputs a value $v \in M$, even if the commitment cannot be unveiled. We call this property *weak extractability* and will account for this in the following definition.

Also, the hiding property informally described in [BN00] seems to be relatively weak, considering honestly created commitments only. Moreover, the adversary's steps are only counted after it is provided the transcript of a successful commit phase. Our definition of timed hiding (Definition 2) is standard and stronger in the sense that the commitment *receiver* may act maliciously. Also, we count the adversary's steps from the very beginning on. It is easy to see that the scheme due to [BN00] satisfies this stronger notion.

With [BN00] not giving a formal definition, we define weakly extractable timed commitment schemes as follows.

Definition 1 (Weakly Extractable Timed Commitment Scheme). *A tuple of ITMs* $\mathsf{TCOM} = \langle \mathsf{C}, \mathsf{R} \rangle$ *is called a* (T, ℓ, ε)-weakly extractable timed commitment scheme *with message space* M *if* $\langle \mathsf{C}, \mathsf{R} \rangle$ *is a* (ℓ, ε)*-hiding commitment scheme for which there exists a deterministic algorithm* `forced-open` *that, given a transcript* c *of a successful commit phase, outputs the unique value* $v \in M$ *committed to in at most* T *steps.*

We say that TCOM is *perfectly correct* if for all $\kappa \in \mathbb{N}$ and all $v \in M$,

$$\begin{aligned}\Pr[v^\star = v' = v \,|(z_\mathsf{C}, z_\mathsf{R}, c) \leftarrow \mathrm{out}\langle \mathsf{C}(v), \mathsf{R}(\varepsilon)\rangle(1^\kappa, \texttt{Commit}),\\ v' \leftarrow \mathrm{out}_\mathsf{R}\langle \mathsf{C}(z_\mathsf{C}), \mathsf{R}(z_\mathsf{R})\rangle(\texttt{Unveil}), v^\star = \texttt{forced-open}(c)] = 1\end{aligned}$$

The perfect correctness can be naturally relaxed to statistical correctness.

Definition 2 ((Timed) Hiding). *For an interactive commitment scheme* $\mathsf{COM} = \langle \mathsf{C}, \mathsf{R} \rangle$*, the timed hiding experiment is defined as:*

```
Experiment $\mathrm{Exp}^{Hiding}_{\mathcal{A},\mathsf{COM}}(\kappa, z)$

    $(m_0, m_1, state) \leftarrow \mathcal{A}(1^\kappa, \texttt{find}, z)$
    $b \xleftarrow{\$} \{0,1\}$
    if $|m_0| \neq |m_1|$
        return $b$
    $b' \leftarrow out_\mathcal{A}\langle \mathsf{C}(m_b), \mathcal{A}(\texttt{guess}, state)\rangle(1^\kappa, \texttt{Commit})$
    return $b = b'$
```

The advantage of a possibly malicious receiver $\mathcal{A}$ *is given by*

$$\mathrm{Adv}^{Hiding}_{\mathcal{A},\mathsf{COM}}(\kappa, z) := \left| \Pr[\mathrm{Exp}^{Hiding}_{\mathcal{A},\mathsf{COM}}(\kappa, z) = 1] - \frac{1}{2} \right| .$$

The probability is over the randomness of $\mathcal{A}$, R *and the choice bit* b*. An adversary* $\mathcal{A}$ *is called* valid *if* $m_0, m_1 \in M$ *and* $\mathcal{A}$ *eventually outputs a single bit. We say that* COM *is* $(\ell(\kappa), \varepsilon(\kappa))$*-hiding if* $\ell(\kappa)$ *is an upper bound for the number of steps performed by* $\mathcal{A}$ *on input* `guess` *and for all* $\kappa \in \mathbb{N}$ *and* $\ell(\kappa)$*-bounded valid* $\mathcal{A}$ *and for all* $z \in \{0,1\}^*$, $\mathrm{Adv}^{Hiding}_{\mathcal{A},\mathsf{COM}}(\kappa, z) \leq \varepsilon(\kappa)$*.*

We say that TCOM is *perfectly binding* and *weakly extractable* if for all (malicious) committers C^*, all $\kappa \in \mathbb{N}$ and all $z \in \{0,1\}^*$, it holds that

$$\begin{aligned}\Pr[v^\star = v' \mid (z_{\mathsf{C}^*}, z_\mathsf{R}, c) \leftarrow \mathrm{out}\langle \mathsf{C}^*(z), \mathsf{R}(\varepsilon)\rangle(1^\kappa, \texttt{Commit}),\\ v' \leftarrow \mathrm{out}_\mathsf{R}\langle \mathsf{C}^*(z_{\mathsf{C}^*}), \mathsf{R}(z_\mathsf{R})\rangle(\texttt{Unveil}), v^\star = \texttt{forced-open}(c) \wedge v' \in M] = 1\end{aligned}$$

While the aforementioned properties do not state any requirements for the output of `forced-open` on invalid commitments (i.e. allow over-extraction), it implies the soundness requirement of [BN00] for valid commitments.

Definition 1 is not concerned with the committer's run-time, which may depend on all parameters, in particular T and ℓ. This is important for (proving) security properties that consider more than one commitment, e.g. the timed simulation-soundness (Definition 5).

Boneh and Naor [BN00] also present a constant-round construction based on the generalized BBS assumption that does not make use of black-box techniques. Also, their construction admits a super-polynomial gap between the number of steps needed to perform the commitment and the number of steps ℓ the commitment is secure against.

While [BN00] consider a machine model that admits parallel computations, we consider (weaker) sequential models of computation only.

Recently, Ephraim et al. [Eph+20] and Katz, Loss, and Xu [KLX20] have re-visited timed commitment schemes, providing formal definitions and new constructions. However, as they consider (non-interactive) timed commitment schemes with setups, their definitions are not easily applicable to our setting.

Timed commitments can also be constructed by combining *sequential functions* [MMV13] and universal hash functions. However, such a construction has the drawback that both commit and unveil phase are computation-intensive. Still, it suffices for a feasibility result with a symmetric assumption.

Looking ahead to our constructions, we remark that using timed commitments with non-malleability properties in the plain model will not lead to easier definitions or proofs due to the power of the simulator. We leave it as an open question whether there are advantages if the simulator is restricted like e.g. in the Angel-based setting.

2.4 pCCA Security

For non-timed commitment schemes, we consider a stronger variant of the hiding property called security under *parallel chosen-commitment attack* (pCCA) [Kiy14,Bro+17,Bro+18]. In the pCCA hiding experiment, the adversary may additionally interact with an (inefficient) oracle $\mathcal{O}$ to perform an unbounded number of commitments *in parallel*, with $\mathcal{O}$ acting as receiver. After all commit phases with $\mathcal{O}$ have finished, $\mathcal{O}$ outputs, for each commitment, the unique value committed to. If no such value exists, a special symbol $\perp$ is returned for this commitment. The challenge commitment where the adversary acts as receiver must remain hiding, even with access to $\mathcal{O}$. pCCA security constitutes a stronger variant of parallel one-left many-right non-malleability.

Definition 3 (pCCA security). *For a commitment scheme* $\mathsf{COM} = \langle \mathsf{C}, \mathsf{R} \rangle$*, the pCCA hiding experiment is defined as*

```
Experiment $\mathrm{Exp}^{pCCA\text{-}Hiding}_{\mathcal{A},\mathsf{COM},\mathcal{O}}(\kappa, z)$
  $(m_0, m_1, tag, state) \leftarrow \mathcal{A}^{\mathcal{O}}(1^\kappa, \texttt{find}, z)$
  $b \xleftarrow{\$} \{0,1\}$
  if $|m_0| \neq |m_1|$
    return $b$
  $b' \leftarrow out_{\mathcal{A}}\langle \mathsf{C}(m_b), \mathcal{A}^{\mathcal{O}}(\texttt{guess}, state)\rangle(\texttt{Commit}, tag)$
  return $b = b'$
```

$\mathcal{O}$ *acts as honest receiver* R *for multiple sessions* in parallel*. When all commit phases have finished, the oracle returns the unique values committed to. If no such unique value exists, a special symbol* $\perp$ *is output for these commitments. An adversary* $\mathcal{A}$ *is valid if it eventually outputs a bit and never interacts with* $\mathcal{O}$ *on the challenge tag. We say that* COM *is pCCA-secure if for all valid PPT adversaries* $\mathcal{A}$*, there exists a negligible function* negl *such that for all* $\kappa \in \mathbb{N}$ *and all* $z \in \{0,1\}^*$*,*

$$\mathrm{Adv}^{pCCA\text{-}Hiding}_{\mathcal{A},\mathsf{COM}}(\kappa, z) := \left| \Pr[\mathrm{Exp}^{pCCA\text{-}Hiding}_{\mathcal{A},\mathsf{COM}}(\kappa, z) = 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\kappa)$$

2.5 Ideal Functionality for Multiple Commitments

The ideal functionality for multiple commitments $\mathcal{F}_{\mathrm{MCOM}}$ in Fig. 1, introduced by [CF01], models ideal bilateral commitments for multiple parties and instances. Individual commitments are distinguished by their commitment ID *cid*.

Functionality $\mathcal{F}_{\mathrm{MCOM}}$

$\mathcal{F}_{\mathrm{MCOM}}$ proceeds as follows, running with parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$.

1. Upon receiving a value $(\texttt{Commit}, sid, cid, P_i, P_j, b)$ from P_i, where $b \in \{0,1\}$, record the tuple (sid, cid, P_i, P_j, b) and generate a public delayed output $(\texttt{Committed}, sid, cid, P_i, P_j)$ to P_j. Ignore subsequent $(\texttt{Commit}, sid, cid, P_i, P_j, \star)$ messages.
2. Upon receiving a value $(\texttt{Unveil}, sid, cid, P_i, P_j)$ from P_i, proceed as follows: If the tuple (sid, cid, P_i, P_j, b) is recorded, then generate a public delayed output $(\texttt{Unveil}, sid, cid, P_i, P_j, b)$ to P_j. Otherwise, do nothing.

Fig. 1. The ideal commitment functionality for multiple commitments $\mathcal{F}_{\mathrm{MCOM}}$ (adapted from [CF01])

3 Timed Simulation-Sound Commitment Schemes

Looking ahead to our construction of a composable commitment scheme (Sect. 5), we need a commitment scheme that is equivocal for a polynomial-time simulator. At the same time, commitments created by a malicious committer must remain binding sufficiently long. To this end, we first define the security notion of *timed simulation-soundness*. Also, we present the construction SSCOM (where SS denotes *simulation-sound*) that combines a possibly malleable timed commitment scheme with a non-timed commitment scheme that is secure under parallel chosen-commitment attacks (pCCA) [Kiy14,Bro+17,Bro+18] and satisfies the notion of timed simulation-soundness.

3.1 Timed Simulation-Soundness

Based on the established notion of simulation-soundness [MY04,GMY03] and inspired by the non-malleability notion of Dachman-Soled et al. [Dac+13], we define a concurrent and timed variant of simulation-soundness that is suitable for commitments where the binding property only holds temporarily (Definition 5). Intuitively, this *timed* simulation-soundness ensures that commitments produced by a malicious committer remain binding for a bounded adversary even if it concurrently receives equivocated commitments. While somewhat similar to the notion of *non-malleability with respect to unveil* or *opening* or *decommitment* ([DIO98,PR05,OPV08]), our definition is stronger in the sense that commit and unveil phases may overlap (similar to the definition of [Dac+13]).

The Experiment. In the experiment for *timed simulation-soundness*, a man-in-the-middle adversary acts as receiver in an unbounded number of instances ("left sessions") of some trapdoor commitment scheme. The adversary starts left sessions by providing a tag of its choice, along with an efficiently samplable and length-normal (cf. Definition 4) distribution. Only considering distributions facilitates easier proofs and more general definitions and is sufficient for our application. In each left session, the code of the trapdoor committer C_{trap} is executed. After the commit phase of a session has finished, the adversary may, at some point of its choice, start the unveil phase. At its onset, a value from the provided distribution is sampled and unveiled by the trapdoor committer.

In addition, the adversary acts as committer in one session ("right session"), again using a tag of its choice that must be unique compared to all other tags that will eventually be used in the experiment. The scheduling between all sessions and their messages is fully controlled the adversary.

When the commit phase of the single right session has finished, the experiment determines the value committed to. The commitment scheme is secure if the adversary cannot unveil its single commitment to a value different from the committed one, even when presented with equivocated commitments.

Timer-Related Parameters. In our setting, we do not consider simulation-soundness against arbitrary polynomial-time adversaries. Indeed, our construction SSCOM is (intentionally) not simulation-sound or even binding against polynomial-time adversaries: If a corrupted receiver manages to break a timed commitment it receives from the (honest) sender early enough, the commitment becomes equivocal. In our setting, protocol parties may set up *timers* and inquire at some point whether the timer has expired. The timed simulation-soundness experiment is thus parameterized with a timed security parameter ℓ. This timed security parameter denotes how many steps experiment and adversary may perform before a timer set up by the honest receiver in the right session is considered to have timed out. If no timeout occurs, the binding property of the single right commitment should hold, even if left commitments are equivocated.

Timed simulation-sound commitments that use timed building blocks such as timed commitment schemes must choose their timed security parameter ℓ' relative to ℓ. To account for reduction overhead, e.g. to the timed hiding property of a timed commitment scheme, ℓ' must be chosen sufficiently large. As the reduction overhead may depend on the security parameter κ but $\ell(\kappa)$ might be constant, ℓ' is also parameterized with κ. Depending on the construction, increasing ℓ may lead to the timer *always* expiring, e.g. because a sub-protocol protected by the timer requires more than ℓ steps to execute (e.g. the commit phase of a timed commitment scheme, which may take longer for larger ℓ) for some values of ℓ. In this case, proving security becomes trivial as the adversary cannot win the game. However, this also implies that scheme is secure in this case. When using appropriate building blocks, e.g. non-interactive timed commitments or a timed commitment scheme with a sufficiently large gap (e.g. the scheme of [BN00] has an exponential gap between the time needed to create the commitment and its timed security), this problem does not occur for sufficiently large ℓ'.

In order to notify parties about timeouts, we require the adversary to obey the following rules: When receiving a message ($\texttt{notify}, id$) for some ID id, it must immediately answer ($\texttt{notify}, id, 1$) if it has previously received ($\texttt{timeout}, id$) and the whole execution experiment, including the adversary and *honest* committers in left sessions, has performed ℓ or more steps, where ℓ is the timed security parameter. For our construction, this can be easily computed as the run-time of the involved algorithms do not depend on their internal randomness or secrets. If an exact calculation is not possible, the adversary must use an appropriate upper bound.

This is in contrast to e.g. Definition 2 where only the steps of the adversary are counted. There, this is possible as only one commitment session is considered. Here, we consider an unbounded number of sessions. In a reduction to some timed property, all the left sessions will have to be emulated by the reduction adversary, counting against its time limit in the reduction.

As the guarantees of timed cryptographic assumptions are only for honest parties, the experiment does not answer `notify` messages.

In real life, one can of course not expect that a possibly malicious party obeys these rules. However, if a timed primitive is believed to be secure for e.g. several days considering the computation power available to the other party, assuming a timeout after, say, one minute, should be sufficiently secure.

Relationship to Other Non-malleability Notions. Similar to the simulation-based non-malleability notion of [Dac+13], security must hold if the commit and unveil phases on the left side are interleaved with the right session. However, in contrast to [Dac+13], we do not require the commitment on the right side to be concurrently extractable and also do not consider adaptive corruptions, leading to a different security notion.

Formal Definition. First, we define length-normal probability distributions as distributions where all elements of the sample space are of equal length.[3]

Definition 4 (Length-normal Probability Distribution). *Let $\mathcal{D}$ be a probability distribution over $\{0,1\}^*$ with sample space Ω. $\mathcal{D}$ is called* length-normal *if for all $x, y \in \Omega$, it holds that $|x| = |y|$. Let $|\mathcal{D}|$ denote $|x|$ for $x \in \Omega$.*

An example for a length-normal distribution is the uniform distribution $\mathcal{U}_n$ over $\{0,1\}^n$ with $|\mathcal{U}_n| = n$.

Definition 5 (Timed Simulation-Soundness). *A trapdoor commitment scheme $\mathsf{TRAPCOM}$ with message space $M \subseteq \{0,1\}^*$ is called $\ell(\kappa)$*-timed simulation-sound *if for all legal PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that for all $\kappa \in \mathbb{N}$ and for all $z \in \{0,1\}^*$, it holds that*

$$\mathrm{Adv}^{SIMSOUND}_{\mathcal{A},\mathsf{TRAPCOM}}(\kappa, \ell(\kappa), z) := \Pr[\mathrm{Exp}^{SIMSOUND}_{\mathcal{A},\mathsf{TRAPCOM}}(\kappa, \ell(\kappa), z) = 1] \leq \mathsf{negl}(\kappa)$$

where the probability is over the random coins of the experiment and the adversary. An adversary $\mathcal{A}$ is called legal *if i) it immediately sends the message $(\mathtt{notify}, id, 1)$ after receiving $(\mathtt{notify}, id)$ and the experiment (including the adversary) has performed more than or equal to $\ell(\kappa)$ steps after having received a message $(\mathtt{timer}, id)$*[4]*, where steps performed by the committer on left sides are counted as of the honest committer C and ii) $\mathcal{A}$ sends $\mathtt{commit\text{-}left}$ messages only parameterized with efficiently samplable and length-normal distributions (cf. Definition 4) where the sample space Ω is a subset of the message space M and iii) the tag used in the right commitment has never been used in a left commitment.*

The random variable $\mathrm{Exp}^{SIMSOUND}_{\mathcal{A},\mathsf{TRAPCOM}}(\kappa, \ell(\kappa), z)$ is defined as follows:

1. *Start the adversary $\mathcal{A}$ with input $(1^\kappa, \ell(\kappa), z)$.*
2. *Upon receiving $(\mathtt{commit\text{-}left}, tag, \mathcal{D}_{tag})$ from the adversary: Start the commit phase of $\mathsf{TRAPCOM}$ with common input $(1^\kappa, \mathtt{commit}, tag, \ell(\kappa))$, acting as trapdoor committer $\mathsf{C}_{\mathrm{trap}}$ with private input $|\mathcal{D}_{tag}|$, unless there already is a session with tag tag.*
3. *Upon receiving $(\mathtt{commit\text{-}right}, tag)$ from the adversary: Start the commit phase of the right session with common input $(1^\kappa, \mathtt{commit}, tag, \ell(\kappa), \kappa)$, acting as honest receiver R, unless the right session already exists or there is a left session with tag tag. Let $v' \in M$ denote the* unique *value committed to in the right session. If no such unique value exists, set $v' = \bot$.*

[3] When considering an appropriate encoding, the definition can be extended to e.g. group elements.

[4] We assume unique timer IDs within a protocol throughout this paper.

4. *Upon receiving* (**unveil-left**, *tag*) *from the adversary: Sample* $v_{tag} \leftarrow \mathcal{D}_{tag}$ *and start the unveil phase of the i-th left session with common input* ($\texttt{unveil}$, *tag*) *and private input* v_{tag} *for the trapdoor committer, unless the commit phase with tag tag has not finished or the unveil phase has already started.*
5. *Upon receiving* (**unveil-right**) *from the adversary: Start the unveil phase of the right session with common input* ($\texttt{unveil}$, *tag*), *acting as honest receiver where tag is the tag specified in the commit phase. Let u denote the value accepted by the receiver or* $\perp$ *in case of an abort.*
6. *Upon receiving* ($\texttt{message}$, *tag*, *m*) *from the adversary, forward the message m to the session with tag tag. Conversely, forward messages to the adversary.*
7. *After the right unveil phase has finished, output 1 if the receiver in the right session has accepted and* $u \neq v' \wedge u \neq \perp$. *Otherwise, output* 0.

For the sake of brevity, we also say that a commitment scheme fulfilling the above definition is $\ell(\kappa)$-simulation-sound.

Like [Dac+13], we call an adversary that wins the above experiment with at most negligible probability *non-abusing*, i.e. if its commitments remain binding even when presented with equivocated commitments. Note that this notion is only meaningful for commitments where the value committed to is uniquely determined (except with negligible probability) if the receiver accepts. To capture the general case, the definition has to be changed slightly.

3.2 Construction SSCOM

In the following, we present the construction SSCOM (Construction 1) for a timed simulation-sound string commitment scheme, which is based on the commitment scheme due to [Bro+17], which is inspired by [DS13]. Roughly, the scheme works as follows: Committer and receiver perform a commitment to a random index vector $I \in \{0,1\}^\kappa$ chosen by the receiver. They then perform 2κ commitments to pair-wise shares of the secret. In the unveil phase, the committer first sends its shares without unveiling the share commitments. Then, the receiver unveils the commitment to I. Finally, the committer unveils the share commitments denoted by I, while the other commitments remain unopened. If the commitment scheme used for I is extractable, the constructed commitment is equivocal. As inconsistent share commitments remain unopened and hiding, a malicious receiver cannot distinguish between an equivocated and a honest commitment. In order to achieve concurrent security, we require the share commitment scheme to be pCCA-secure (Definition 3).

In contrast to the original construction of [Bro+17], we use a timed commitment scheme for the commitment to the index vector I, which allows polynomial-time equivocation of SSCOM commitments. Also, we move this timed commitment to I to the end of the commit phase. For the sake of simpler proofs, we assume that the commitment scheme for the shares is perfectly binding. However, this requirement can be relaxed to statistically binding.

To facilitate easy integration with our composable commitment scheme and the timed simulation-soundness definitions, SSCOM includes explicit messages to set up timers and to check if they have expired. Again, the party answering the timer status inquiry checks if both parties have performed ℓ or more steps since the timer has been set up and answers accordingly. In the simulation-soundness experiment, the answer is given by the adversary that is required to answer truthfully. Again, it would have been possible to only count steps by the party that has *not* set up the timer. However, counting the steps of both parties is more consistent with our other definitions and more convenient in reductions.

Construction 1 (Commitment Scheme SSCOM). *Parameterized by a security parameter κ, a timed security parameter $\ell(\kappa)$, a pCCA-secure and perfectly binding commitment scheme $\mathsf{COM}_{\mathsf{pCCA}}$ and a $(T, \ell'(\ell(\kappa), \kappa), \mathsf{negl}(\kappa))$-weakly extractable timed commitment scheme TCOM.*

Commit Phase. On common input $(1^\kappa, \mathtt{commit}, tag, \ell(\kappa))$, committer and receiver interact as follows:

1. *The committer creates 2κ shares $s_{1,0}, s_{1,1}, \ldots, s_{\kappa,0}, s_{\kappa,1}$ of its private input v by sampling $s_{m,0} \xleftarrow{\$} \{0,1\}^{|v|}$ and setting $s_{m,1} = v \oplus s_{m,0}$, $m = 1, \ldots, \kappa$.*
2. *For $m = 1, \ldots, \kappa$, $n = 0, 1$, committer and receiver start 2κ instances of $\mathsf{COM}_{\mathsf{pCCA}}$ on common input $(1^\kappa, \mathtt{commit}, (tag, m, n))$ in parallel. The committer's private input in the instance with tag (tag, m, n) is $s_{m,n}$.*
3. *The receiver samples an index vector $I \xleftarrow{\$} \{0,1\}^\kappa$ and sends $(\mathtt{timer}, tag)$ to the committer. Then, committer and receiver start an instance of TCOM with common input $(1^\kappa, \mathtt{commit}, \ell(\ell'(\kappa), \kappa))$. The receiver of SSCOM acts as committer with private input I.*

Unveil Phase. On common input $(\mathtt{unveil}, tag)$, committer and receiver interact as follows:

1. *The committer sends the shares $(s_{1,0}, \ldots, s_{\kappa,1})$ to the receiver.*
2. *The receiver sends $(\mathtt{notify}, tag)$ to the committer, which the receiver answers with $(\mathtt{notify}, tag, b)$ where $b = 1$ if committer and receiver have spent more than or equal to $\ell(\kappa)$ steps since the timer has been set up. Otherwise, $b = 0$ indicates that less than $\ell(\kappa)$ steps in total have elapsed. If the committer answers with $(\mathtt{notify}, tag, 1)$, the receiver aborts. Otherwise, the receiver checks that $s_{1,0} \oplus s_{1,1} = \cdots = s_{\kappa,0} \oplus s_{\kappa,1}$ and aborts if this does not hold. Then, committer and receiver perform the unveil phase of TCOM. The committer also makes sure of the TCOM commitment being extractable (to the value I) in at most T steps, e.g. by using the $\mathtt{forced\text{-}open}$ algorithm. If this check fails, the committer aborts.*
3. *Committer and receiver perform κ unveil phases of $\mathsf{COM}_{\mathsf{pCCA}}$ as follows: For $m = 1, \ldots, \kappa$, the commitment to $s_{m,I[m]}$ with tag $(tag, m, I[m])$ is unveiled. Let $s'_{m,I[m]}$ denote the unveiled value of the commitment with tag $(tag, m, I[m])$.*

4. *After all unveil phases have finished, the receiver checks that $s'_{m,I[m]} = s_{m,I[m]}$, $m = 1, \ldots, \kappa$. If this holds, the receiver outputs $s_{1,0} \oplus s_{1,1}$. Otherwise, it aborts.*

Algorithm of the Trapdoor Committer C_{trap}.

1. *On private input l in the commit phase, commit honestly to 0^l.*
2. *On private input $v \in \{0,1\}^l$ in the unveil phase, extract the timed commitment using the* `forced-open` *algorithm to obtain the index vector I. If* `forced-open` *fails, sample $I \xleftarrow{\$} \{0,1\}^\kappa$ uniformly at random. For $m = 1, \ldots, \kappa$, send $s_{m,1-I[m]} = v \oplus s_{m,I[m]}$ as shares that will not be unveiled. Continue the unveil phase like the honest committer.*

Theorem 1. *Let $\mathsf{COM}_{\mathsf{pCCA}}$ be a pCCA-secure and perfectly binding commitment scheme with message space $M \subseteq \{0,1\}^*$. Let TCOM be a $(T, \ell'(\ell(\kappa), \kappa), \mathsf{negl}(\kappa))$-weakly extractable timed commitment scheme for some polynomially bounded $T > \ell'(\ell(\kappa), \kappa)$, sufficiently large timed security parameter $\ell'(\ell(\kappa), \kappa)$ and negligible function negl with message space $\{0,1\}^\kappa$. Then, SSCOM is an $\ell(\kappa)$-simulation-sound and trapdoor commitment scheme with message space M.*

It is easy to see that a successful commit phase of SSCOM statistically determines the value committed to. Looking ahead to the security proof of our composable commitment scheme, we will additionally need this value to be extractable in the presence of concurrently equivocated left sides. For the definition of extractability and the proof of Theorem 1, see the full version.

Possible Instantiations. Our construction SSCOM makes use of a weakly extractable timed commitment scheme TCOM as well as a pCCA-secure and perfectly binding commitment scheme $\mathsf{COM}_{\mathsf{pCCA}}$. A possible instantiation for the latter is the commitment scheme of Goyal et al. [Goy+14] which is pCCA-secure [Bro+17], constant-round, non-black-box, parallel extractable and perfectly binding if using e.g. the commitment scheme due to Blum [Blu81] based on one-way permutations as elementary commitment. By instead using a perfectly binding and homomorphic commitment scheme, the construction becomes perfectly binding and black-box [Bre+15, Bro+17].

Corollary 1. *Assume that constant-round, perfectly binding and homomorphic commitment schemes exist. Assume that constant-round, timed commitment schemes with appropriate parameters exist. Then, SSCOM is a constant-round timed simulation-sound commitment scheme from standard assumptions that makes black-box use of its building blocks only.*

An example for a constant-round homomorphic commitment scheme is the ElGamal commitment scheme based on the DDH assumption [ElG84], which does not use non-black-box techniques. With respect to the timed commitment scheme, we can e.g. use the scheme due to Boneh and Naor [BN00] based on the generalized BBS assumption, which is constant-round and also does not use non-black-box techniques.

Corollary 2. *Assume that the DDH assumption and the generalized BBS assumption hold. Then, there exists a constant-round, timed simulation-sound commitment scheme that does not use non-black-box techniques.*

4 TLUC Security in a Nutshell

Timed primitives such as timed commitment schemes can be meaningfully used in practice. Consider performing a coin-toss using a timed commitment scheme secure for, say, $t = 10^{15}$ steps. Assuming that the adversary can perform at most 10^{10} steps per second (equating 10 GHz, assuming that steps equate cycles)[5], a coin-toss using this timed commitment should be considered secure if the adversary's second-round message comes within e.g. one second of receiving the timed commitment, with plenty time left as security margin.

TLUC Security. Unfortunately, this intuition is not easily captured in the UC framework, which neither offers a notion of time nor makes assumptions with respect to the (concrete) computational power of entities. Instead of considering a model with time or modifying the framework, we propose a variant of UC security, called TLUC security, that enables honest parties to check if more than ℓ steps have been performed since a certain point in the execution. This allows to capture the security guarantees of timed primitives and to use them in protocols.

With TLUC, parties can set up *timers* parameterized by an ID and a number of computation steps ℓ by sending $(\texttt{timer}, id, \ell)$ to the adversary[6]. At any point, a party that has set up a timer may check if it has expired, i.e. if the whole execution experiment has performed ℓ or more steps since the timer has been set up. This is done by sending $(\texttt{notify}, id)$ to the adversary. The adversary queries the environment if the timer has expired answers with $(\texttt{notify}, id, b)$, where $b = 1$ denotes an expired timer and $b = 0$ an unexpired one.

Mechanisms. The correct handling of timers is ensured by considering only *legal environments* and *legal adversaries.* Intuitively, legal environments correctly account for timers set up by honest parties by never under-estimating the number of computation steps performed by the execution experiment relative to a presumptive execution of a protocol π (counting obliviously of the parties' inputs and outputs) and adversary $\mathcal{A}$, denoted by $\mathcal{Z}[\pi, \mathcal{A}]$. This guarantees that timed assumptions protect against environment and adversary, but can be broken by the simulator in polynomial time (as the environment $\mathcal{Z}[\pi, \mathcal{A}]$ always counts relative to π and $\mathcal{A}$, even when interacting with ϕ and $\mathcal{S}$). For technical reasons, we

[5] This is even more plausible when using cryptographic assumptions that are believed to be hard even for parallel adversaries.

[6] In contrast to stand-alone experiments where `timer` messages are not parameterized with the timed security parameter, we have chosen to do so in the TLUC setting because the mechanism should be agnostic of the currently executed protocol and its timed security parameter.

require handling of timers and inquiries to go through the adversary. An adversary is legal if it immediately and correctly forwards timer setup messages or status inquiries by honest parties, as well as the environment's responses. Based on this, we define TLUC emulation as a special case of UC emulation, and consider legal adversaries and environments only. At first glance, this might seem restrictive, but when considering standard UC protocols without timers, then all UC environments and adversaries are legal under our definition. Thus, the restrictions only apply for classes of protocols that are not considered by UC security.

Properties of TLUC Security. As we consider only a subset of the UC environments and adversaries, properties of UC security do not necessarily carry over to TLUC security, at least for protocols using timers. To the contrary, even properties such as the completeness of the dummy adversary are difficult to prove if concrete time bounds must be adhered to. We show several properties such as transitivity with UC protocols, i.e. protocols whose security does not rely on timers[7], completeness of the dummy adversary or full compatibility with UC security as well as UC reusability, meaning that all UC-secure protocols are also TLUC-secure and can be composed with TLUC protocols without loss of security. With respect to the latter, we state the single instance composition theorem.

The ability of the simulator to break timed assumptions while environment and real-world adversary are unable to do is sufficient to construct a composable commitment scheme in the plain model. When, e.g., combining our commitment scheme with a UC-secure general MPC protocol in the $\mathcal{F}_{\mathrm{COM}}$- or $\mathcal{F}_{\mathrm{MCOM}}$-hybrid model[8], we obtain a composable general MPC protocol in the plain model.

While composable MPC in the plain model is already possible in a number of other frameworks, previous approaches rely on some sort of super-polynomial or non-uniform simulation. The first may affect the security of concurrently executed protocols relying on polynomial-time hardness assumptions, resulting in limited *environmental friendliness* as defined by [CLP13]. TLUC security only considers entities that run in strict polynomial time. The second may affect the security of protocols that have been previously started, even ones that are secure against non-uniform adversaries. Our feasibility results also hold for uniform simulators.

Thus, TLUC security is the first notion that features composable constant-round black-box MPC in the plain model from standard (timed) assumptions as well as full environmental friendliness and does not hurt the security of previously started protocols relying on polynomial-time assumptions.

This informal description is sufficient to understand the properties of TLUC security as well as the construction in Sect. 5. For a full treatment of TLUC security, see the full version.

[7] A UC protocol π that UC-realizes an ideal functionality $\mathcal{F}$ may of course send `timer` messages. However, as UC emulation also considers environments that handle these messages arbitrarily, the security of π cannot rely on them.

[8] $\mathcal{F}_{\mathrm{MCOM}}$ and the multi-session extension $\hat{\mathcal{F}}_{\mathrm{COM}}$ of $\mathcal{F}_{\mathrm{COM}}$ are equivalent [CR03].

4.1 Protocol Emulation

We define TLUC emulation in analogy to UC emulation.

Definition 6 (TLUC Emulation). *Let π and ϕ be protocols. We say that π TLUC-emulates ϕ if for all legal PPT adversaries $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$ such that for all legal PPT environments $\mathcal{Z}[\pi,\mathcal{A}]$ there exists a negligible function* negl *such that for all $\kappa \in \mathbb{N}, a \in \{0,1\}^*$ it holds that*

$$|\Pr[\mathsf{Exec}\big(\pi,\mathcal{A},\mathcal{Z}[\pi,\mathcal{A}]\big)(\kappa,a)=1]-\Pr[\mathsf{Exec}\big(\phi,\mathcal{S},\mathcal{Z}[\pi,\mathcal{A}]\big)(\kappa,a)=1]|\leq \mathsf{negl}(\kappa)$$

If π TLUC-emulates ϕ, we write $\pi \geq_{\text{TLUC}} \phi$. When omitting the non-uniform input a, the notion of protocol emulation is uniform.

Note that in Definition 6, the environment $\mathcal{Z}$ is supposed to count the steps according to the execution with π and $\mathcal{A}$ even if it is actually interacting with ϕ and $\mathcal{S}$. This allows the PPT-bounded simulator $\mathcal{S}$ to perform more steps than the adversary $\mathcal{A}$ without triggering a time-out, allowing it to break timed assumptions. If ϕ is an UC protocol, its security is not affected by such a powerful simulator. In contrast, if ϕ is a protocol making use of timers, honest parties of the protocol ϕ may not rely on timing assumptions as the adversary $\mathcal{S}$ is allowed to violate them unnoticed.

Meaningfulness of TLUC Security. When introducing a new security notion, it is important to argue that it does not allow to prove the security of "obviously" insecure protocols. The basic idea behind TLUC security is the very same as behind established simulation-based security notions, where a protocol's security is defined through the ideal functionality it realizes. For simulation-based security notions, care has to be taken that the simulator's capabilities do not affect the security guarantees of the ideal functionality. For example, SPS security is not meaningful for ideal functionalities that use a polynomial-time hardness assumption like a signature scheme that can be broken by the super-polynomial simulator. As TLUC simulations are always polynomial-time, they do not affect an ideal functionality that makes use of polynomial-time assumptions. What is more, we show that non-trivial functionalities can be realized using a *uniform* polynomial-time simulation.

In total analogy to both UC security and other composable security notions that admit general MPC in the plain model, we can show strong impossibility results. This underlines that the new mechanism of timers does not help the simulator *per se.*

4.2 Properties of TLUC Security

Having defined protocol emulation, we can state important properties of TLUC security in analogy to properties of UC security.

Proposition 1 (Legality of the Dummy Adversary). *The dummy adversary $\mathcal{D}$ is legal.*

Proposition 1 immediately follows from the definition of the dummy adversary in the UC framework.

As in UC security, it is sufficient to show protocol emulation with respect to the dummy adversary.

Proposition 2 (Completeness of the Dummy Adversary). *Let π and ϕ be protocols. Then, $\pi \geq_{\text{TLUC}} \phi$ if and only if π TLUC-emulates ϕ with respect to the dummy adversary.*

TLUC security is also compatible with UC security, meaning that UC-secure protocols are also TLUC-secure.

Proposition 3 (Compatibility with UC Security). *Let π, ϕ be protocols such that $\pi \geq_{\text{UC}} \phi$. Then, $\pi \geq_{\text{TLUC}} \phi$.*

In contrast to UC security, TLUC security is not transitive. This means that there exist protocols π_1, π_2, π_3 such that $\pi_1 \geq_{\text{TLUC}} \pi_2$ and $\pi_2 \geq_{\text{TLUC}} \pi_3$, but $\pi_1 \not\geq_{\text{TLUC}} \pi_3$. For an example, see the full version.

However, TLUC emulation is transitive in conjunction with UC emulation.

Proposition 4 (TLUC-UC Transitivity). *Let π_1, π_2, π_3 be protocols. If $\pi_1 \geq_{\text{TLUC}} \pi_2$ and $\pi_2 \geq_{\text{UC}} \pi_3$, then it holds that $\pi_1 \geq_{\text{TLUC}} \pi_3$.*

In the following, we consider the case of a protocol ρ that makes one subroutine call to a protocol ϕ.

Theorem 2 (Single Instance Composition Theorem). *Let π, ϕ be subroutine-respecting protocols such that $\pi \geq_{\text{TLUC}} \phi$. Let ρ be a protocol that makes one subroutine call to ϕ. Then, $\rho^\pi \geq_{\text{TLUC}} \rho^\phi$.*

Let ρ be a protocol that UC-emulates the ideal protocol IDEAL($\mathcal{G}$) of some ideal functionality $\mathcal{G}$ and makes one subroutine call to the ideal protocol IDEAL($\mathcal{F}$) of some ideal functionality $\mathcal{F}$. Using Propositions 3 and 4 and Theorem 2, we can import ρ into TLUC, replace IDEAL($\mathcal{F}$) with an appropriate TLUC protocol while preserving security and conclude that the resulting composite protocol TLUC-emulates IDEAL($\mathcal{G}$).

Corollary 3 (UC Reusability). *Let π and ϕ be subroutine-respecting protocols such that $\pi \geq_{\text{TLUC}} \phi$. Let ρ be a protocol that makes one subroutine call to ϕ such that $\rho^\phi \geq_{\text{UC}} \sigma$. Then, $\rho^\pi \geq_{\text{TLUC}} \sigma$.*

Unfortunately, TLUC security is not closed under general composition. More concretely, this means that there exist subroutine-respecting protocols π and ϕ such that $\pi \geq_{\text{TLUC}} \phi$ holds, but $\rho^\pi \not\geq_{\text{TLUC}} \rho^\phi$, where ρ makes *multiple* subroutine calls to ϕ. For an example, see the full version.

UC security has the desirable property of *environmental friendliness* [CLP13], which, informally, ensures that game-based security properties of protocols running along UC protocols ("in the environment") are not impacted by the UC execution. Unfortunately, this property does not hold for *all* game-based security properties for many notions that allow composable MPC in the plain model due to the use of super-polynomial simulation. What is more, determining whether the game-based property holds may be non-trivial, requiring e.g. to consider the security proof of the protocol in question. However, as TLUC security is a special case of UC security with polynomial-time simulation only, it inherits the environmental friendliness of UC security.

For an explanation and definition of environmental friendliness, see [CLP13] and the full version.

Proposition 5 (Environmental Friendliness of TLUC Security). *Let π be a protocol that TLUC-emulates the ideal protocol of some functionality $\mathcal{G}$. Then π is friendly to every (non-timed) game-based property P of a protocol Π with property P.*

Protocols running alongside composable MPC protocols may not only be affected by super-polynomial simulation, but also by non-uniform simulation. For example, Lin, Pass, and Venkitasubramaniam [LPV09] propose a variant of UC security where the environment runs in uniform polynomial-time, while the simulator runs in non-uniform polynomial-time. The non-uniform input of the simulator may impact the security of protocols that have started before the input is given to the simulator—even if these protocols are secure against non-uniform adversaries. As the definition of environmental friendliness is non-uniform, it does not capture this property.

Both the simulation and the reductions for our composable commitment scheme (Sect. 5) are uniform. Our constructions thus do not adversely affect security properties of previously started protocols that hold against polynomial-time adversaries.

Remark 1. Environmental friendliness as defined by [CLP13] is not meaningful for *timed* game-based properties such as the timed hiding property of a timed commitment scheme.

When considering an ideal functionality $\mathcal{F}$ and a concurrently executed protocol π using timed assumptions, the functionality $\mathcal{F}$ may already be unfriendly to timed properties of π. For example, $\mathcal{F}$ may perform computations that break time-lock puzzles used in π.

In the experiment of environmental friendliness, no simulator is not used. The (presumptive) simulator is only used to show that a protocol π is as friendly as a functionality $\mathcal{F}$ (which may already be unfriendly in our setting). Thus, the problems of environmental friendliness to protocols using timed assumptions start well before considering the effects of the simulation, which additionally affect the environmental friendliness.

To the best of our knowledge, this novel environmental friendliness for timed game-based properties is not fulfilled by *any* security notion for composable MPC—not even by UC security.

While there exists no general and formal definition of *non-triviality* in the UC framework, Canetti et al. [Can+02] consider a protocol π to be a non-trivial realization of $\mathcal{F}$ if $\pi \geq_{\text{UC}}$ IDEAL($\mathcal{F}$) and for all adversaries $\mathcal{A}$ that deliver all messages and do not corrupt any party, the simulator $\mathcal{S}$ allows all outputs generated by $\mathcal{F}$.

With TLUC security, this notion is not sufficient as it does not consider the possibility that a protocol aborts due to timeouts, which may, depending e.g. on the environment, occur even if the adversary delivers all messages.

As an example, let π be a protocol that non-trivially UC-emulates $\mathcal{F}_{\mathrm{COM}}$ and takes $t(\kappa)$ steps to execute successfully if all parties are honest. Now, let π' be the protocol that is identical to π, with the following exception. When receiving its input, the honest committer sets up a timer with $10t(\kappa)$ steps. At the onset of the unveil phase, it checks if the timer has expired and halts upon expiration. Clearly, π' should be considered non-trivial.

However, there exists a legal environment such that π' never generates output even if the legal adversary delivers all messages. As we do not want π' to be considered trivial if there also exists a legal environment $\mathcal{Z}$ for which π' always generates an output under the conditions outlined in [Can+02], we thus consider an appropriate notion that accounts for this.

Note that non-triviality may be lost under composition. To this end, take a protocol ρ^{ϕ} that makes one subroutine call to some protocol ϕ and is non-trivial. Replacing ϕ with its realization π that takes more steps than ϕ may make the composed protocol ρ^{π} trivial as timers in ρ may *always* be triggered due to the additional steps performed by the protocol π. However, that this does *not* render ρ^{π} insecure.

The well-known impossibility results due to Canetti and Fischlin [CF01] state that there is no bilateral (i.e. involving two communicating parties) and terminating (in the sense of correctness for honest parties) protocol π that UC-realizes $\mathcal{F}_{\mathrm{COM}}$ in the plain model. This is due to the fact that if a protocol π is in the plain model, an environment is able to internally emulate every (presumptive) UC simulator for π.

We state the following variant of the impossibility result of [CF01] for TLUC-realizing $\mathcal{F}_{\mathrm{COM}}$ in the plain model:

Theorem 3. *There exists no bilateral, non-trivial protocol π in the plain model where only one party sets up timers such that $\pi \geq_{\mathrm{TLUC}} \mathcal{F}_{\mathrm{COM}}$.*

By introducing a temporary asymmetry between simulator and environment, e.g. when the environment counts the steps relative to the real-world adversary, non-trivial and environmentally friendly realizations of UC-complete functionalities in the plain model using timed assumptions become possible.

5 Composable Commitments in the Plain Model

We are now ready to present our construction π_{MCOM} that TLUC-realizes the ideal functionality $\mathcal{F}_{\mathrm{MCOM}}$ (Fig. 1) and prove its security. Our construction is based on the $\mathsf{UCC}_{\mathsf{OneTime}}$ commitment scheme in the $\mathcal{F}_{\mathrm{CRS}}$-hybrid model due to Canetti and Fischlin [CF01].

In the original scheme $\mathsf{UCC}_{\mathsf{OneTime}}$, which is suitable for a single commitment only, the CRS consists of two parts: a pair of public keys (pk_0, pk_1) for a trapdoor PRG (cf. [CF01]) as well as a uniformly random string $\sigma \in \{0,1\}^{4\kappa}$. With the knowledge of the associated secret keys (sk_0, sk_1), it is possible to extract commitments. By changing the distribution of σ in an indistinguishable way, the commitment becomes equivocal.

To enable simulation in the case of static corruptions, the knowledge of only *one* trapdoor, depending on which party is corrupted, is sufficient. The other trapdoor does not even have to exist. Assuming trapdoor one-way permutations with dense public description [DP92], we can perform two coin-tosses to generate (pk_0, pk_1) resp. σ. While our coin-toss protocol (see Sect. 5.1) is not fully simulatable, it is simulatable if the simulator plays the initiator. This suffices to set up the extraction trapdoor if the sender is corrupted by having the commitment receiver, played by the simulator, start the coin-toss for (pk_0, pk_1). The simulator can equivocate the result to public keys for which it knows the secret keys. Conversely, the coin-toss for σ is started by the commitment sender. If it is honest, the simulator can simulate the coin-toss such that σ contains an equivocation trapdoor. From that point on, the original $\mathsf{UCC}_{\mathsf{OneTime}}$ scheme is executed, using the values obtained by this preamble phase instead of the CRS as in the original protocol. For each new commitment between two parties, the preamble phase is re-executed. A similar approach is used in [Dac+13].

Our coin-toss protocol π_{CT} uses the trapdoor commitment scheme SSCOM (see Sect. 3.2) whose equivocation trapdoor is protected by a timed commitment that can be extracted by the simulator. As SSCOM is timed simulation-sound, SSCOM commitments of corrupted committers remain binding if opened in time.

TLUC security does not imply concurrent self-composability. Thus, we cannot simply prove the security of a single commitment and conclude that it holds for multiple commitments performed concurrently. Indeed, when using weaker building blocks, our construction can be shown to securely realize one instance of $\mathcal{F}_{\mathrm{COM}}$, but not $\mathcal{F}_{\mathrm{MCOM}}$, where the latter captures concurrent self-composition.

In the following, we thus prove that π_{MCOM} TLUC-realizes the ideal functionality $\mathcal{F}_{\mathrm{MCOM}}$ for multiple commitments. Later on, we can plug π_{MCOM} into any (UC-secure) protocol making one subroutine call to $\mathcal{F}_{\mathrm{MCOM}}$ while maintaining security.

5.1 The Coin-Toss Protocol π_{CT}

One important building block towards constructing our TLUC-secure commitment scheme is the coin-toss protocol π_{CT} (Construction 2). It is essentially identical to the protocol due to Blum [Blu81], except for the use of a string commitment and with the addition of handling the timers of SSCOM.

Construction 2 (Coin-Toss Protocol π_{CT})**.** *Parameterized by a security parameter κ, a timed security parameter ℓ, a length parameter $s = s(\kappa)$ and a $\ell'(\ell,\kappa)$-simulation-sound commitment scheme* SSCOM *with message space $M \supseteq \{0,1\}^s$.*

1. *On input* $(\texttt{coin-toss}, sid, s)$*, the sender samples* $r \xleftarrow{\$} \{0,1\}^s$*.*

2. *Sender and receiver start an instance of* SSCOM *on common input* $(1^\kappa, \texttt{commit}, sid, \ell(\kappa), \ell'(\ell(\kappa), \kappa))$*. The sender's private input for the commitment is* r*. All* `notify` *messages are forwarded between the adversary and the parties of* SSCOM*. Messages* $(\texttt{timer}, id)$ *coming from a* SSCOM *party are forwarded to the adversary as* $(\texttt{timer}, id, \ell)$*, i.e. augmented with the timed security parameter* ℓ*.*
3. *After the commit phase has finished, the receiver samples* $r' \overset{\$}{\leftarrow} \{0,1\}^s$ *uniformly at random and sends* (sid, r') *to the sender.*
4. *Upon receiving* (sid, r')*, sender and receiver perform the unveil phase of* SSCOM*.*
5. *If the receiver accepts, sender and receiver output* $(\texttt{coin-toss}, sid, r \oplus r')$*. Otherwise, the execution halts.*

As SSCOM is not straight-line extractable, we cannot show that π_{CT} TLUC-realizes the coin-toss functionality $\mathcal{F}_{\mathrm{CT}}$. However, π_{CT} exhibits the following useful properties: If the commitment receiver is corrupted, the coin-toss is simulatable. If the sender is corrupted and does not abort, the result of the coin-toss is distributed uniformly at random. Due to the simulation-soundness of SSCOM, the result of one session is independent from all other instances of π_{CT} that may run concurrently, with the exception of aborts skewing the distribution.

We do not prove these properties on their own, but show them implicitly in the proof of the construction of the commitment scheme.

5.2 The Commitment Scheme π_{MCOM}

We now give the construction of the composable commitment scheme π_{MCOM}.

Construction 3 (Commitment Scheme π_{MCOM})**.** *Parameterized by a timed security parameter* $\ell(\kappa)$ *and a trapdoor PRG* PRG *with key space* $\{0,1\}^{l(\kappa)}$ *for some polynomial* l*, domain* $\{0,1\}^\kappa$ *and range* $\{0,1\}^{4\kappa}$*.*

Commit Phase.

1. *Upon receiving* $(\texttt{commit}, sid, cid, P_i, P_j, b)$ *as input for the committer* P_i*, committer* P_i *and receiver* P_j *execute two instances of* π_{CT} *with timed security parameter* $\ell(\kappa)$ *to generate*
   (a) $(pk_0, pk_1) \in \{0,1\}^{l(\kappa)} \times \{0,1\}^{l(\kappa)}$ *(the "extraction CRS") with the receiver acting as initiator in* π_{CT} *with session ID* $(sid, cid, 0)$*, where* $l(\kappa)$ *is the length of public keys of* PRG*.*
   (b) $\sigma \in \{0,1\}^{4\kappa}$ *(the "equivocation CRS") with the sender acting as initiator in* π_{CT} *with session ID* $(sid, cid, 1)$*.*

   If both instances of π_{CT} *terminate successfully, both parties store* $(sid, cid, (pk_0, pk_1, \sigma))$*. Otherwise, they halt the execution.*
2. *The committer samples* $r \overset{\$}{\leftarrow} \{0,1\}^\kappa$ *and sets* $c = \mathsf{PRG}(pk_0, r)$ *if* $b = 0$ *and* $c = \mathsf{PRG}(pk_1, r) \oplus \sigma$ *if* $b = 1$*. Then, the committer sends* $(\texttt{commitment}, sid, cid, c)$ *to the receiver. The committer stores* $(sid, cid, (b, r, c))$*, the receiver stores* (sid, cid, c) *and outputs* $(\texttt{committed}, sid, cid, P_i, P_j)$*.*

Unveil Phase.

1. *Upon receiving* $(\texttt{unveil}, sid, cid, P_i, P_j)$ *as input, the committer sends* $(\texttt{unveil}, sid, cid, (b, r))$ *to the receiver.*
2. *Upon receiving* $(\texttt{unveil}, sid, cid, (b, r))$ *from the sender, the receiver checks if* $c = \mathsf{PRG}(pk_0, r)$ *for* $b = 0$ *or if* $c = \mathsf{PRG}(pk_1, r) \oplus \sigma$ *for* $b = 1$*, relative to the values stored for this* sid *and* cid*. If the check is successful, the receiver outputs* $(\texttt{unveil}, sid, cid, P_i, P_j, b)$ *and halts otherwise.*

Theorem 4. *Assume that* PRG *is a trapdoor PRG with dense public description and that* SSCOM *is a (computationally) trapdoor, extractable and timed simulation-sound commitment scheme. Then,* $\pi_{\text{MCOM}} \geq_{\text{TLUC}} \text{IDEAL}(\mathcal{F}_{\text{MCOM}})$*.*

For a proof, see the full version.

Remark 2. Our technique also weakens the assumptions for practical complexity leveraging: We can replace the timed commitment scheme with a "weak" commitment scheme that is initially hiding for all polynomial-time environments and adversaries, but extractable for the simulator (that must not be able to break the other complexity assumptions used in the protocol). The security of this "weak" commitment scheme thus can be very low, as the simulation remains indistinguishable as long as the "weak" commitments remain hiding during their use in the coin-toss. Afterwards, they do not need to be hiding anymore.

6 Constant-Round Black-Box Composable General MPC

In order to achieve composable general MPC, we can plug the construction π_{MCOM} into any UC-secure general MPC protocol in the $\mathcal{F}_{\text{MCOM}}$-hybrid model while maintaining security (using Corollary 3).

Hazay and Venkitasubramaniam [HV15] have presented a constant-round and black-box general MPC protocol in the $\mathcal{F}_{\text{CRS}}$-hybrid model based on public-key encryption and constant-round semi-honest oblivious transfer. Following the approach used in [Bro+17], we can generate the CRS of the [HV15] protocol with a simulatable coin-toss, assuming that IND-CPA-secure PKE schemes with oblivious public-key exist, thus casting the protocol in the $\mathcal{F}_{\text{MCOM}}$-hybrid model.

Theorem 5. *Assume that constant-round timed commitment schemes with appropriate parameters and perfectly binding homomorphic commitment schemes exist. Also, assume that trapdoor one-way permutations with dense public description and IND-CPA-secure PKE schemes with oblivious public-key generation exist. Then, for every well-formed*[9] *functionality* $\mathcal{F}$*, there exists a constant-round protocol* $\pi_{\mathcal{F}}^{BB}$ *in the plain model such that* $\hat{\pi}_{\mathcal{F}}^{BB} \geq_{\text{TLUC}} \text{IDEAL}(\hat{\mathcal{F}})$ *and* $\pi_{\mathcal{F}}^{BB}$ *uses its building blocks in a black-box way only.*

[9] Informally, a functionality $\mathcal{F}$ is *well-formed* if its behavior is independent of which parties are corrupted [Can+02].

In Theorem 5, $\hat{\mathcal{F}}$ denotes the multi-session existence of $\mathcal{F}$ (cf. [CR03]) that naturally captures concurrent self-composition.

Considering possible candidates for timed commitments and perfectly binding homomorphic commitment schemes, we obtain the following corollary.

Corollary 4. *Assume that the generalized BBS assumption and the DDH assumption hold and that trapdoor one-way permutations with dense public description exist. Then, for every well-formed functionality $\mathcal{F}$, there exists a constant-round protocol $\pi_{\mathcal{F}}^{BB}$ in the plain model such that $\hat{\pi}_{\mathcal{F}}^{BB} \geq_{\mathrm{TLUC}} \mathrm{IDEAL}(\hat{\mathcal{F}})$ and $\pi_{\mathcal{F}}^{BB}$ does not use non-black-box techniques.*

7 Conclusion

We constructed a composable constant-round black-box general MPC protocol in the plain model from standard (timed) assumptions only. In contrast to previous techniques for general MPC in the plain model, our approach fully fulfills the notion of environmental friendliness.

The approach outlined in this paper could also give a new direction to complexity leveraging. The weaker level of security would have to hold only while the protocol is executed.

Looking ahead, it remains to investigate if these results can be obtained more efficiently and from weaker or more generic assumptions and if stronger properties, e.g. with respect to transitivity or composition, can be achieved. With the recent popularity of timed assumptions, it is necessary to define a meaningful extension of environmental friendliness for timed game-based security properties.

Acknowledgements. Jeremias Mechler, Jörn Müller-Quade: This work was supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

References

[Bau+20] Baum, C., et al.: CRAFT: composable randomness and almost fairness from time. Cryptology ePrint Archive, Report 2020/784 (2020)

[Bau+21] Baum, C., David, B., Dowsley, R., Nielsen, J.B., Oechsner, S.: TARDIS: a foundation of time-lock puzzles in UC. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12698, pp. 429–459. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77883-5_15

[Blu81] Blum, M.: Coin flipping by telephone. In: Gersho, A. (ed.) CRYPTO'81. Vol. ECE Report 82-04, pp. 11–15. Dept. of Elec. and Computer Eng., U.C., Santa Barbara (1981)

[BMM21] Broadnax, B., Mechler, J., Müller-Quade, J.: Environmentally friendly composable multi-party computation in the plain model from standard (timed) assumptions. Cryptology ePrint Archive, Report 2021/843 (2021). https://ia.cr/2021/843

[BN00] Boneh, D., Naor, M.: Timed commitments. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 236–254. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_15

[Bre+15] Brenner, H., et al.: Fast non-malleable commitments. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015, Denver, CO, USA, pp. 1048–1057. ACM Press (2015)

[Bro+17] Broadnax, B., Döttling, N., Hartung, G., Müller-Quade, J., Nagel, M.: Concurrently composable security with shielded super-polynomial simulators. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10210, pp. 351–381. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_13

[Bro+18] Broadnax, B., Fetzer, V., Müller-Quade, J., Rupp, A.: Non-malleability vs. CCA-security: the case of commitments. In: Abdalla, M., Dahab, R. (eds.) PKC 2018. LNCS, vol. 10770, pp. 312–337. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76581-5_11

[BS05] Barak, B., Sahai, A.: How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In: 46th FOCS, Pittsburgh, PA, USA, pp. 543–552. IEEE Computer Society Press (October 2005)

[Can+02] Canetti, R., et al.: Universally composable two-party and multiparty secure computation. In: 34th ACM STOC, Montréal, Québec, Canada, pp. 494–503. ACM Press (May 2002)

[Can+07] Canetti, R., Dodis, Y., Pass, R., Walfish, S.: Universally composable security with global setup. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 61–85. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_4

[Can01] Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: 42nd FOCS, Las Vegas, NV, USA, pp. 136–145. IEEE Computer Society Press (October 2001)

[CF01] Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_2

[CLP10] Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: 51st FOCS, Las Vegas, NV, USA, pp. 541–550. IEEE Computer Society Press (October 2010)

[CLP13] Canetti, R., Lin, H., Pass, R.: From unprovability to environmentally friendly protocols. In: 54th FOCS, Berkeley, CA, USA, pp. 70–79. IEEE Computer Society Press (October 2013)

[CR03] Canetti, R., Rabin, T.: Universal composition with joint state. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 265–281. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_16

[Dac+13] Dachman-Soled, D., Malkin, T., Raykova, M., Venkitasubramaniam, M.: Adaptive and concurrent secure computation from new adaptive, non-malleable commitments. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 316–336. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_17

[DIO98] Di Crescenzo, G., Ishai, Y., Ostrovsky, R.: Non-interactive and non-malleable commitment. In: 30th ACM STOC, Dallas, TX, USA, May 1998, pp. 141–150. ACM Press (1998)

[DP92] De Santis, A., Persiano, G.: Zero-knowledge proofs of knowledge without interaction (extended abstract). In: 33rd FOCS, Pittsburgh, PA, USA, October 1992, pp. 427–436. IEEE Computer Society Press (1992)

[DS13] Damgård, I., Scafuro, A.: Unconditionally secure and universally composable commitments from physical assumptions. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 100–119. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_6

[ElG84] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_2

[Eph+20] Ephraim, N., et al.: Non-malleable time-lock puzzles and applications. Technical report (2020)

[Gar+12] Garg, S., Goyal, V., Jain, A., Sahai, A.: Concurrently secure computation in constant rounds. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 99–116. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_8

[GKP18] Garg, S., Kiyoshima, S., Pandey, O.: A new approach to black-box concurrent secure computation. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10821, pp. 566–599. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_19

[GMY03] Garay, J.A., MacKenzie, P., Yang, K.: Strengthening zero-knowledge protocols using signatures. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 177–194. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_11

[Gol08] Goldreich, O.: Computational complexity - a conceptual perspective. Cambridge University Press (2008). https://doi.org/10.1017/CBO9780511804106

[Goy+14] Goyal, V., et al.: An algebraic approach to non-malleability. In: 55th FOCS, Philadelphia, PA, USA. IEEE Computer Society Press, pp. 41–50 (October 2014)

[HV15] Hazay, C., Venkitasubramaniam, M.: On black-box complexity of universally composable security in the CRS model. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 183–209. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_8

[IPS08] Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_32

[Kiy14] Kiyoshima, S.: Round-efficient black-box construction of composable multiparty computation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8617, pp. 351–368. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44381-1_20

[KLP05] Kalai, Y.T., Lindell, Y., Prabhakaran, M.: Concurrent general composition of secure protocols in the timing model. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, Baltimore, MA, USA, May 2005, pp. 644–653. ACM Press (2005)

[KLX20] Katz, J., Loss, J., Xu, J.: On the security of time-lock puzzles and timed commitments. In: Pass, R., Pietrzak, K. (eds.) TCC 2020. LNCS, vol. 12552, pp. 390–413. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64381-2_14

[LPV09] Lin, H., Pass, R., Venkitasubramaniam, M.: A unified framework for concurrent security: universal composability from stand-alone non-malleability. In: Mitzenmacher, M. (ed.) 41st ACM STOC, Bethesda, MD, USA, pp. 179–188. ACM Press (2009)

[MMV13] Mahmoody, M., Moran, T., Vadhan, S.P.: Publicly verifiable proofs of sequential work. In: Kleinberg, R.D. (ed.) ITCS 2013, Berkeley, CA, USA, pp. 373–388. ACM (2013)

[MPR06] Micali, S., Pass, R., Rosen, A.: Input-indistinguishable computation. In: 47th FOCS, Berkeley, CA, USA, October 2006, pp. 367–378. IEEE Computer Society Press (2006)

[MY04] MacKenzie, P., Yang, K.: On simulation-sound trapdoor commitments. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 382–400. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_23

[OPV08] Ostrovsky, R., Persiano, G., Visconti, I.: Constant-round concurrent non-malleable commitments and decommitments. Cryptology ePrint Archive, Report 2008/235 (2008). https://eprint.iacr.org/2008/235

[Pas03] Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_10

[PR05] Pass, R., Rosen, A.: New and improved constructions of non-malleable cryptographic protocols. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, Baltimore, MA, USA, pp. 533–542. ACM Press (2005)

[PS04] Prabhakaran, M., Sahai, A.: New notions of security: achieving universal composability without trusted setup. In: Babai, L. (ed.) 36th ACM STOC, Chicago, IL, USA, pp. 242–251. ACM Press (2004)

[RSW96] Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto (1996)