Revisiting Lower and Upper Bounds for Selective Decommitments

Rafail Ostrovsky[1,2], Vanishree Rao[1], Alessandra Scafuro[3,*], and Ivan Visconti[3]

[1] Department of Computer Science, UCLA, USA
[2] Department of Mathematics, UCLA, USA
{rafail,vanishri}@cs.ucla.edu
[3] Dipartimento di Informatica, University of Salerno, Italy
{scafuro,visconti}@dia.unisa.it

Abstract. In [6,7], Dwork et al. posed the fundamental question of existence of commitment schemes that are secure against selective opening attacks (SOA, for short). In [2] Bellare, Hofheinz, and Yilek, and Hofheinz in [13] answered it affirmatively by presenting a scheme which is based solely on the non-black-box use of a one-way permutation needing a super-constant number of rounds. This result however opened other challenging questions about achieving a better round complexity and obtaining fully black-box schemes using underlying primitives and code of the adversary in a black-box manner.

Recently, in TCC 2011, Xiao ([23]) investigated on how to achieve (nearly) optimal SOA-secure commitment schemes where optimality is in the sense of both the round complexity and the black-box use of cryptographic primitives. The work of Xiao focuses on a simulation-based security notion of SOA. Moreover, the various results in [23] focus only on either parallel or concurrent SOA.

In this work we first point out various issues in the claims of [23] that actually re-open several of the questions left open in [2,13]. Then, we provide new lower bounds and concrete constructions that produce a very different state-of-the-art compared to the one claimed in [23].

1 Introduction

Commitment schemes are a fundamental building block in cryptographic protocols. By their usual notion, they satisfy two security properties, namely, hiding and binding. While the binding property guarantees that a committed message can not be opened to two distinct messages, the hiding property ensures that before the decommitment phase begins, no information about the committed message is revealed. Binding and hiding are preserved under concurrent composition, in the sense that even a concurrent malicious sender will not be able to open a committed message in two ways, and even a concurrent malicious receiver will not be able to detect any relevant information about committed messages as long as only commitment phases have been played so far.

* Work done while visiting UCLA.

In [6], Dwork et al. pointed out a more subtle definition of security for hiding where the malicious receiver is allowed to ask for the opening of only some of the committed messages, with the goal of breaking the hiding property of the remaining committed messages. This notion was captured in [6] via a simulation-based security definition, and is referred to as hiding in presence of selective opening attack (SOA, for short). [6] shows that, in a *trusted setup* setting, it is possible to construct a non-interactive SOA-secure commitment scheme from a trapdoor commitment scheme. Indeed, in the trusted setup the simulator sets the parameters of the trapdoor commitment, thus obviously it knows the trapdoor. However, the fundamental question of whether there exist SOA-secure commitment schemes in the *plain model*, is left open in [6]. We stress that the question is particularly important since commitments are often used in larger protocols, where often only some commitments are opened but the security of the whole scheme still relies on hiding the unopened commitments. For instance, the importance of SOA-secure commitments for constructing zero-knowledge sets is discussed in [10][1].

The SOA-security experiment put forth in [6] considers a one-shot commitment phase, in which the receiver gets all commitments in one-shot, picks adaptively a subset of them, and obtains the opening of such subset. Such definition implicitly considers non-interactive commitments and only parallel composition. Subsequent works have explored several extensions/variations of this definition showing possibility and impossibility results. Before proceeding to the discussion of the related work, it is useful to set up the dimensions that will be considered. One dimension is *composition*. As commitment is a two-phase functionality, other than parallel composition, one can consider two kinds of concurrent composition. Concurrent-with-barrier composition (considered in [2,13]), refers to the setting in which the adversarial receiver can interleave the execution of several commitments, and the execution of decommitments, with the restriction that all commitment phases are played before any decommitment phase begins. Thus, there is a barrier between commitment and decommitment stage. Fully-concurrent composition (considered in [23]) refers to the setting in which the adversary can arbitrarily interleave the execution of the commitment phase of one session with the decommitment of another session (and vice-versa).

Next dimension is the *access to primitive*, namely, if the construction uses a cryptographic primitive as a black-box (in short, BB), or in a non black-box way (in short, NBB).

Another dimension is *simulation*. In this discussion we consider always black-box simulators (if not otherwise specified).

The question of achieving SOA-secure commitments without any set-up was solved affirmatively in [2] by Bellare, Hofheinz, and Yilek, and by Hofheinz in [13], who presented an interactive SOA-secure scheme based on non-black-box use of any one-way permutation and with a commitment phase requiring a super-constant number of rounds. The security of such construction is proved in

[1] In [10] some forms of zero-knowledge sets were proposed, and their strongest definition required SOA-secure commitments.

the concurrent-with-barrier setting. [2,13] also show that non-interactive SOA-secure commitments which use cryptographic primitives in a black-box way do not exist. The same work introduces the notion of *indistinguishability* under selective opening attacks, that we do not consider in this work. The results of [2,13] left open several other questions on round optimality and black-box use of the underlying cryptographic primitives.

In TCC 2011 [23], Xiao addressed the above open questions and investigated on how to achieve nearly optimal schemes where optimality concerns both the round complexity and the black-box use of cryptographic primitives. In particular, Xiao addressed SOA-security of commitment schemes for both parallel composition and fully-concurrent composition and provided both possibility and impossibility results, sticking to the simulation-based definition. Concerning positive results, [23] shows a 4-round (resp., $(t + 3)$-round for a t-round statistically-hiding commitment) computationally binding (resp., statistically binding) SOA-secure scheme for parallel composition. Moreover, [23] provides a commitment scheme which is "strong" (the meaning of strong is explained later) SOA-secure in the fully-concurrent setting and requires a logarithmic number of rounds. All such constructions are fully black-box. Concerning impossibility results, [23] shows that 3-round (resp., 4-round) computationally binding (resp., statistically binding) parallel SOA-secure commitment schemes are impossible to achieve. As explained later, in this paper – among other things – we present issues in the proof of security of the constructions shown in [23]. We also show that, the strong security claimed for the construction suggested for the fully-concurrent setting, is actually impossible to achieve, regardless of the round complexity. We contradict the lower bounds claimed in [23] by providing a 3-round fully black-box commitment scheme which is SOA-secure under concurrent-with-barrier composition, which implies parallel composition.

In a subsequent work [25], after our results became publicly available, Xiao showed a black-box construction of 4-round statistically-binding commitment scheme which is SOA-secure under parallel composition. As we shall see later, our 3-round and 4-round schemes are only computationally binding, but are secure in the stronger setting of concurrent-with-barrier composition.

In [24], the same author provides an updated version of [23]. Concerning positive results, [24] includes the $(t + 3, 1)$-round construction of [23] and shows a new simulation strategy for it. Concerning impossibility results, [24] includes the lower bounds of [23] that are still valid for 2-round (resp., 3-round) computationally hiding and computationally (resp., statistically) binding, parallel, SOA-secure commitment schemes with black-box simulators. [24] contains also other contributions of [23] that are not contradicted by this work.

In [1], Bellare et al. proves that existence of CRHFs implies impossibility of non-interactive SOA-secure commitments (regardless of the black-box use of the cryptographic primitives). In fact, they show something even stronger; they show that this impossibility holds even if the simulator is non-black-box and knows the distribution of the message space. An implication of such results is that, standard security does not imply SOA-security. Previous results in [2,13] only showed the impossibility for the case of black-box reductions.

In [19], Pass and Wee provide several black-box constructions for two-party protocols. In particular, they provide constructions for look-ahead trapdoor commitments (in a look-ahead commitment, knowledge of the trapdoor is necessary already in the commitment phase in order for the commitment to be equivocal, in this paper we call such commitments "weak trapdoor commitments"), and trapdoor commitments. Such constructions have not been proved to be SOA-secure commitment schemes, as SOA-security is proven in presence of (at least) parallel composition, while security of the trapdoor commitment of [19] is proved only in the stand-alone setting. In the full version of this paper we discuss how the look-ahead thread of [19] can be plugged in our construction based on weak trapdoor commitments, to obtain a 6-round SOA-secure commitment scheme in the concurrent-with-barrier setting.

1.1 Our Contribution

In this work we focus on simulation-based SOA-secure commitment schemes, and we restrict our attention to black-box simulation, and (mainly) black-box access to cryptographic primitives (like in [23]). Firstly, we point out various issues in the claims of [23]. These issues essentially re-open some of the open questions that were claimed to be answered in [23]. We next show how to solve (in many cases in a nearly optimal way) all of them. Interestingly, our final claims render quite a different state-of-the-art from (and in some cases also in contrast to) the state-of-the-art set by the claims of [23].

In detail, by specifying as (x, y) the round complexity of a commitment scheme when the commitment phase takes x rounds and the decommitment phase takes y rounds, we revisit some claims of [23] and re-open some challenging open questions as follows.

1. The proof in [23] of the non-existence of $(3, 1)$-round schemes assumes implicitly that the sender sends the last message during the commitment phase. We show here that surprisingly this assumption is erroneous, and that one round might be saved in the commitment phase if the receiver goes last. This re-opens the question of the achievability of $(3, 1)$-round SOA-secure schemes, even for just parallel composition.
2. There are issues in the proof of binding and SOA-security of the $(4, 1)$-round scheme of [23] for parallel composition, and it is currently unknown whether the scheme is secure. The same issue in the SOA-security proof exists for the $(t + 3, 1)$-round statistically binding scheme of [23] which is based on any t-round statistically-hiding commitment. Indeed, for both constructions, SOA-security is claimed to follow from the simulation technique of Goldreich-Kahan [11]. The problem is that the simulator of [11] was built for a *stand-alone* zero-knowledge protocol, where an atomic sub-protocol is repeated several times in parallel, and the verifier cannot *selectively* abort one of the sub-protocols. Instead in the SOA-setting, the adversarial receiver interacts

with multiple senders and can decide to abort a subset of the sessions of its choice adaptively based on the commitment-phase transcript.

We note that this implies that [23] contains no full proof of a constant round SOA-secure scheme (but we remark that, subsequent to our results, the same author presented a new proof of the $(t+3,1)$-round scheme based on statistically-hiding commitment in the work [25,24]).

3. There is an issue in the proof of security of the fully-concurrent SOA-secure commitment scheme proposed in [23]. The security of such construction is claimed even for the case in which the simulator cannot efficiently sample from the distribution of messages committed to by the honest sender (but needs to query an external party for it).[2] This notion is referred in [23] as "strong" security. This issue in [23] re-opens the possibility of achieving schemes that are strong SOA-secure under fully concurrent composition (for any round complexity).

In this paper we solve the above open problems (still sticking to the notion of black-box simulation as formalized in [23]) as follows.

1. We present a $(3,1)$-round scheme based on BB use of any trapdoor commitment (TCom, for short), which contradicts the lower bound claimed in [23]. We also provide several constructions based on BB use of various weaker assumptions. We show: a $(3,3)$-round scheme based on BB use of any OWP, a $(4,1)$-round scheme based on BB use of any weak trapdoor commitment (wTCom, for short)[3], and a $(5,1)$-round scheme based on BB use of any OWP.
2. We show that when the simulator does not know the distribution of the messages committed to by the honest sender, there exists no scheme that achieves fully concurrent SOA-security, regardless of the round complexity and of the BB use of cryptographic primitives. Thus contradicting the claimed security of the construction given in [23].
3. As a corollary of our $(3,1)$-round scheme based on BB use of any TCom, there exists a $(3,1)$-round scheme based on NBB use of any one-way function (OWF). This improves the round complexity in [2] from logarithmic in the security parameter to only 3 rounds and using minimal complexity-theoretic assumptions. Moreover, we observe that (as a direct consequence from proof techniques in [23]) a $(2,1)$-round SOA-secure scheme is impossible regardless of the use of the underlying cryptographic primitive (for black-box simulation only). Thus, our $(3,1)$-round scheme for black-box simulation is essentially round-optimal.

Notice that both our $(3,1)$-round protocols – the one based on BB use of TCom and the other based on NBB use of OWFs – contradict the impossibility given

[2] For simplicity, we shall hereafter refer to this case as the simulator not knowing the distribution.

[3] This result indeed requires a relaxed definition of trapdoor commitment where the trapdoor is required to be known already during the commitment phase in order to later equivocate. We call it "weak" because any TCom is also a wTCom.

in [23], that was claimed to hold regardless of the access to the cryptographic primitives.

All the constructions shown in this paper are secure under concurrent-with-barrier composition, which obviously implies parallel composition. Our simulators work for any message distributions, and do not need to know the distribution of the messages committed to by the honest sender. In light of our impossibility for the fully concurrent composition (see Item 2 of the above list), the concurrency achieved by our schemes seems to be optimal for this setting.

As an additional application, we also show that our $(3,1)$-round schemes can be used to obtain non-interactive (concurrent) zero knowledge [8] with 3 rounds of pre-processing. This improves upon [5] where (at least) 3 rounds of interactions are needed both in the pre-processing phase and in the proof phase. Moreover, the simulator of [5] works only with non-aborting verifiers, while our simulator does not have this limitation. This application also establishes usefulness of concurrency-with-barrier setting.

Comparison with Previous Work. For a better clarity we compare previous results and constructions with the contribution of this paper in Table 1. In the table, listings under "Impossible" column refer to the impossibility results in the papers heading the corresponding row. Similarly, the listings under "Constructions" column refer to the constructions in the papers heading the corresponding row. All such constructions are proved via black-box simulations. BB (resp., NBB) stands for black-box (resp., non black-box) access to cryptographic primitives. PAR, CwB, CC, refer respectively to parallel, concurrent-with-barrier, fully-concurrent composition. PB and SB are shorthands for perfectly binding and statistically binding, and TCom, wTCom, OWP are shorthands for Trapdoor Commitment, weak Trapdoor Commitment, One-Way Permutation. For instance, an entry like, "BB $(1,1)$ (or PB) PAR" under Impossible column and for the row [2,13], says that [2,13] demonstrate that non-interactive, or perfectly-binding commitment schemes that are SOA-secure under parallel composition, are impossible to construct given only BB access to cryptographic primitives. The entry, "NBB $(\log n, 1)$ CwB OWP" under Constructions column and in the row headed by [2] says that [2] shows a $(\log n, 1)$-round scheme based on NBB use of OWPs that is SOA-secure under concurrent-with-barrier composition.

On Simulator not Knowing Message Distribution. All our protocols and impossibility results are in the setting where the simulator by itself cannot efficiently sample from the message distribution but needs to query an external oracle for the same. Positive results can only be stronger with this requirement.

Selective Opening, Adaptive Security, Trapdoor Commitments and Non-malleable Commitments. The concept of commitments secure in presence of selective opening attacks is very related to adaptively secure commitments[4], and trapdoor

[4] In such a notion, an adversary can corrupt a party anytime during the protocol execution, obtaining the party's internal state.

Table 1. This work in relation to the state-of-the-art

	Impossible	Constructions
[2,13]	BB $(1, 1)$ (or PB) PAR	NBB $(\log n, 1)$ CwB OWP
[1]	NBB $(1, 1)$ PAR	
[6]		Set-up assumption: BB $(1, 1)$ CC TCom
This Paper on [19]		* BB (x, y) LA-TCom implies BB $(2 + x, y)$ CwB
[23]	$(3, 1)$ PAR $(o(\log n/\log\log n), 1)$ CC BB SIM	BB $(4, 1)$ PAR OWP BB $(t + 3, 1)$ PAR $(t, 1)$-SH BB $(\omega(t \log n), 1)$ CC $(t, 1)$-SH
This Paper on [23]	~~$(3, 1)$ PAR~~ BB SIM	~~BB $(4, 1)$ PAR OWP~~ ~~BB $(t + 3, 1)$ PAR $(t, 1)$-SH~~ ~~BB $(\omega(t \log n), 1)$ CC $(t, 1)$-SH~~
This Paper	BB (any, any) CC BB SIM	BB $(3, 1)$ TCom; NBB $(3, 1)$ OWF BB $(4, 1)$ wTCom ; BB $(3, 3)$ OWP BB $(5, 1)$ OWP (all CwB)
[25]		BB SB $(t + 2, 1)$ PAR $(t, 1)$-SH
[24]		BB $(t + 3, 1)$ PAR $(t, 1)$-SH

commitments, – in which there exists a trapdoor that allows to open a commitment in many ways. However, they are three different settings.

First, trapdoor commitments are not necessarily SOA-secure (in the plain model). The reason is that trapdoor commitments only guarantee that *there exists* a trapdoor which would allow to equivocate a commitment, however, such trapdoor is clearly not available to a simulator. To achieve SOA-security from a trapdoor commitment scheme one should provide a mechanism for the simulator to get the trapdoor, still not violating binding. The converse moreover is not true, namely, a SOA-secure commitment is not necessarily also a trapdoor commitment. This comes from the fact that the simulator of a SOA-secure commitment could use rewinding capabilities instead of a trapdoor that might not exist at all.

Second, a commitment scheme that is adaptively secure in the parallel composition, is also parallel SOA-secure commitment. The converse is not necessarily true. Namely, a SOA-secure commitment scheme is not necessarily adaptively secure. The reason is that in a selective opening attack, a malicious receiver can "corrupt" the sender in the decommitment phase only, and by definition, it is allowed to see only the openings of the commitments instead of the *whole* state of the sender.

Finally, we stress that our adversary can play as sender only or as receiver only. An interesting question is which type of SOA security can be achieved also against man-in-the-middle attacks. The recent work of [12] gives hope towards a construction of a constant-round non-malleable SOA-secure protocol with black-box simulation and black-box use of any one-way function.

2 Preliminaries

Notation. We denote by $n \in \mathbb{N}$ the security parameter and by PPT the property of an algorithm of running in probabilistic polynomial-time. A function ϵ is negligible (negl., for short) in n (or just negligible) if for every polynomial $p(\cdot)$ there exists a value $n_0 \in \mathbb{N}$ such that for all $n > n_0$ it holds that $\epsilon(n) < 1/p(n)$. We denote by $[k]$ the set $\{1, \ldots, k\}$; $\texttt{poly}(n)$ stands for polynomial in n. We denote by $x \leftarrow \mathcal{D}$ the sampling of an element x from the distribution $\mathcal{D}$. We also use $x \overset{\$}{\leftarrow} \mathsf{A}$ to indicate that the element x is uniformly sampled from set A. We denote by $(v_A, v_B) \leftarrow \langle A(), B() \rangle$ the pair of outputs of parties A and B, respectively, after the completion of their interaction. We use $v \overset{\$}{\leftarrow} A()$ when the algorithm A is randomized. Finally, let P_1 and P_2 be two parties running a protocol that uses protocol $\langle A, B \rangle$ as a sub-routine. When we say that party "P_1 runs $\langle A(\cdot), B(\cdot) \rangle$ with P_2" we always mean that P_1 executes the procedure of party A and P_2 executes the procedure of party B. In the paper we use the words decommitment and opening interchangeably.

2.1 Commitment Schemes

In the following definitions we assume that parties are stateful and that malicious parties obtain auxiliary inputs, although for better readability we omit them.

Definition 1 (Bit Commitment Scheme). *A commitment scheme is a tuple of PPT algorithms* $\mathsf{Com} = (\mathsf{Gen}, \mathsf{S}, \mathsf{R})$ *implementing the following two-phase functionality.* Gen *takes as input a random n-bit string r and outputs the public parameters pk. Given to* S *an input $b \in \{0,1\}$, in the first phase (*`commitment phase`*)* S *interacts with* R *to commit to the bit b; we denote this interaction as* $\langle \mathsf{S}(pk, \texttt{com}, b), \mathsf{R}(\texttt{rcv}) \rangle$. *In the second phase (*`opening phase`*)* S *interacts with* R *to reveal the bit b, we denote this interaction as* $\langle \mathsf{S}(\texttt{open}), \mathsf{R}(\texttt{open}) \rangle$ *and* R *finally outputs a bit b' or $\perp$. Consider the following two experiments:*

Experiment $\mathbf{Exp}^{\text{binding}}_{\mathsf{Com},\mathsf{S}^*}(n)$*:*	**Experiment** $\mathbf{Exp}^{\text{hiding-}b}_{\mathsf{Com},\mathsf{R}^*}(n)$*:*
R *runs* $(pk) \leftarrow \mathsf{Gen}(r)$ *and sends pk to* S^*;	$pk^* \leftarrow \mathsf{R}^*(1^n)$;
$\langle \mathsf{S}^*(pk, \texttt{com}, b), \mathsf{R}(\texttt{rcv}) \rangle$;	$(\cdot, b') \overset{\$}{\leftarrow} \langle \mathsf{S}(pk^*, \texttt{com}, b), \mathsf{R}^*(\texttt{rcv}) \rangle$;
$(\cdot, b_0) \overset{\$}{\leftarrow} \langle \mathsf{S}^*(\texttt{open}, 0), \mathsf{R}(\texttt{open}) \rangle$;	*output* b'.
rewind S^* *and* R *back after the second step;*	
$(\cdot, b_1) \overset{\$}{\leftarrow} \langle \mathsf{S}^*(\texttt{open}, 1), \mathsf{R}(\texttt{open}) \rangle$;	
output 1 *iff* $\perp \neq b_0 \neq b_1 \neq \perp$.	

$\mathsf{Com} = (\mathsf{Gen}, \mathsf{S}, \mathsf{R})$ *is a commitment scheme if the following conditions hold:*

Completeness. *If* S *and* R *are honest, for any* S*'s input $b \in \{0,1\}$ the output of* R *in the opening phase is $b' = b$.*

Hiding. *For any PPT malicious receiver* R^**, there exists a negligible function ϵ such that the following holds:*

$$\mathbf{Adv}^{\text{hiding}}_{\mathsf{Com},\mathsf{R}^*} = |\Pr[(\mathbf{Exp}^{\text{hiding-0}}_{\mathsf{Com},\mathsf{R}^*}(n) \rightarrow 1)] - \Pr[\mathbf{Exp}^{\text{hiding-1}}_{\mathsf{Com},\mathsf{R}^*}(n) \rightarrow 1)]| \leq \epsilon(n).$$

Binding. *For any PPT malicious sender* S^* *there exists a negl. function ϵ such that:* $\Pr[\mathbf{Exp}^{\text{binding}}_{\mathsf{Com},\mathsf{S}^*} \rightarrow 1] \leq \epsilon(n)$.

The above probabilities are taken over the choice of the randomness r for the algorithm Gen *and the random coins of the parties. A commitment scheme is statistically hiding (resp., binding) if hiding (resp., binding) condition holds even against an unbounded malicious Receiver (resp., Sender).*

The above definition is a slight modification of the one provided in [2,13], and is more general in the fact the it includes the algorithm Gen. Such a definition is convenient when one aims to use commitment schemes as sub-protocols in a black-box way. However, for better readability, when we construct or use as sub-protocol a commitment scheme that does not use public parameters, we refer to it only as $\mathsf{Com} = (\mathsf{S}, \mathsf{R})$, omitting the algorithm Gen.

Definition 2 (Trapdoor Commitment). *A tuple of PPT algorithms* $\mathsf{TC} = (\mathsf{TGen}, \mathsf{S}, \mathsf{R}, \mathsf{TFakeD})$ *is a trapdoor commitment scheme if* TGen*, on input a random n-bit string r, outputs a public key/secret key pair (pk,sk),* TGen_{pk} *is the related functionality that restricts the output of* TGen *to the public key,* $(\mathsf{TGen}_{pk}, \mathsf{S}, \mathsf{R})$ *is a commitment scheme, and* $(\mathsf{S}, \mathsf{TFakeD})$ *are such that:*

Trapdoor Property. *There exists $b^\star \in \{0,1\}$, such that for any $b \in \{0,1\}$, for all $(pk, sk) \leftarrow \mathsf{TGen}(r)$, and for any PPT malicious receiver R^* there exists a negl. function ϵ such that the following holds:*

$$\mathbf{Adv}^{\mathsf{trapdoor}}_{\mathsf{TC},\mathsf{R}^*} = \Pr[\mathbf{Exp}^{\mathsf{Trap}}_{\mathsf{TC}}(n) \to 1] - \Pr[\mathbf{Exp}^{\mathsf{Com}}_{\mathsf{TC}}(n) \to 1] \leq \epsilon(n).$$

The probability is taken over the choice of r for the algorithm TGen *and the random coins of the players.*

```
Experiment $\mathbf{Exp}^{\mathsf{Com}}_{\mathsf{TC}}(n)$ :
$\mathsf{R}^*$ chooses a bit $b$;
$\langle \mathsf{S}(pk, \mathtt{com}, b), \mathsf{R}^*(pk, sk, b, \mathtt{rcv})\rangle$;
$(\cdot, b') \overset{\$}{\leftarrow} \langle \mathsf{S}(\mathtt{open}), \mathsf{R}^*(\mathtt{open})\rangle$;
output $b'$;
```

```
Experiment $\mathbf{Exp}^{\mathsf{Trap}}_{\mathsf{TC}}(n)$:
$\mathsf{R}^*$ chooses a bit $b$;
$(\xi, \cdot) \leftarrow \langle \mathsf{S}(pk, \mathtt{com}, b^\star), \mathsf{R}^*(pk, sk, b, \mathtt{rcv})\rangle$;
$(\cdot, b') \overset{\$}{\leftarrow} \langle \mathsf{TFakeD}(sk, \mathtt{open}, b, \xi), \mathsf{R}^*(\mathtt{open})\rangle$;
output $b'$;
```

In the experiment $\mathbf{Exp}^{\mathsf{Trap}}_{\mathsf{TC}}(n)$, S runs the procedure of the honest sender on input $b^\star$. The variable ξ contains the randomness used by S to compute the commitment phase and it is used by TFakeD to compute the decommitment. The knowledge of the trapdoor is required only in decommitment phase. In the trapdoor commitment of Pedersen [20], the trapdoor property holds for any $b^\star$, namely one can use the honest sender procedure to commit an arbitrary bit $b^\star$ and use the trapdoor to decommit to any $b \neq b^\star$. Instead, in the trapdoor commitment proposed by Feige and Shamir [9][5], the trapdoor property holds only if the honest procedure was used to commit to bit $b^\star = 0$. In both commitment schemes the trapdoor is used only in the decommitment phase.

[5] The commitment procedure consists of running the simulator of Blum's protocol [3] for Graph Hamiltonicity where the challenge is the bit to commit to. This commitment use OWFs in a NBB way.

Definition 3 (Hiding in the presence of Selective Opening Attacks (slight variation of [2,13])). *Let $k = \mathtt{poly}(n)$, let $\mathcal{B}$ be a k-bit message distribution and $\mathbf{b} \xleftarrow{\$} \mathcal{B}$ be a k-bit vector, let $\mathcal{I} = \{\mathcal{I}_k\}_{k\in\mathbb{N}}$ be a family of sets, where each $\mathcal{I}_k$ is a set of subsets of $[k]$ denoting the set of legal subsets of (indexes of) commitments that the receiver (honest or malicious) is allowed to ask for the opening. A commitment scheme $\mathsf{Com} = (\mathsf{Gen}, \mathsf{S}, \mathsf{R})$ is secure against selective opening attacks if for all k, all sets $I \in \mathcal{I}$, all k-bit message distributions $\mathcal{B}$, all PPT relations $\mathcal{R}$, there exists an expected PPT machine Sim such that for any PPT malicious receiver R^* there exists a negl. function ϵ such that:*

$$\mathbf{Adv}^{\mathsf{soa}}_{\mathsf{Com}} = \left|\mathsf{Pr}[\mathbf{Exp}^{\mathrm{real}}_{\mathsf{Com},\mathsf{S},\mathsf{R}^*}(n) \rightarrow 1] - \mathsf{Pr}[\mathbf{Exp}^{\mathrm{ideal}}_{\mathsf{Com},\mathsf{Sim},\mathsf{R}^*}(n) \rightarrow 1]\right| \leq \epsilon(n).$$

The probability is taken over the choice of the random coins of the parties.

```
Experiment $\mathbf{Exp}^{\mathrm{real}}_{\mathsf{Com},\mathsf{S},\mathsf{R}^*}(n)$:
$pk \xleftarrow{\$} \mathsf{R}^*(1^n)$;
$\mathbf{b} \xleftarrow{\$} \mathcal{B}$;
$I \xleftarrow{\$} \langle \mathsf{S}_i(pk, \mathtt{com}, \mathbf{b}[i])_{i\in[k]}, \mathsf{R}^*(pk, \mathtt{rcv})\rangle$;
$(\cdot, ext) \xleftarrow{\$} \langle \mathsf{S}_i(\mathtt{open})_{i\in I}, \mathsf{R}^*(\mathtt{open})\rangle$;
output $\mathcal{R}(I, \mathbf{b}, ext)$.
```

```
Experiment $\mathbf{Exp}^{\mathrm{ideal}}_{\mathsf{Com},\mathsf{Sim},\mathsf{R}^*}(n)$:
$pk \xleftarrow{\$} \mathsf{R}^*(1^n)$;
$\mathbf{b} \xleftarrow{\$} \mathcal{B}$;
$I \xleftarrow{\$} \mathsf{Sim}^{\mathsf{R}^*}(pk)$;
$ext \xleftarrow{\$} \mathsf{Sim}^{\mathsf{R}^*}(\mathbf{b}[i])_{i\in I}$;
output $\mathcal{R}(I, \mathbf{b}, ext)$.
```

We denote by $(\cdot, ext) \xleftarrow{\$} \langle \mathsf{S}_i(\cdot), \mathsf{R}^*(\cdot)\rangle$ the output of R^* after having interacted concurrently with k instances of S each one denoted by S_i. In the paper an instance of the protocol is called session. A malicious receiver R^* can run many sessions in concurrency with the following limitation. R^* runs commitment phases concurrently for polynomially many sessions, but it can initiate the first decommitment phase only after the commitment phases of all the sessions have been completed (and therefore after the set of indexes has been requested). This means that the set of indexes I (i.e., the commitments asked to be opened), depends only of the transcript of the commitment phase. We call this definition **concurrent-with-barrier** (*CwB*, for short), meaning that many commitment phases (decommitment phases) can be run concurrently but the commitment phase of any session cannot be interleaved with the decommitment of any other session. Notice that as in [23], our definition assumes that the honest receiver chooses to open only a subset of the commitments, but this is done independently of the transcript (i.e., $I \xleftarrow{\$} \mathcal{I}$).

We now discuss the choices that we made to obtain the above definitions.

Concurrency-with-barrier Composition vs Parallel and Concurrent Composition. In [23] Xiao provides two main definitions: SOA-security under parallel (PAR) composition and SOA-security under "fully" concurrent composition (CC). In the fully concurrent definition there is no barrier between commitment and decommitment phase: R^* is allowed to interleave the commitment phase of one session with the decommitment phase of another, basically having the power of deciding which decommitment/commitment to execute, depending on the transcript of the commitment *and* decommitment of other sessions. This definition is pretty general, but unfortunately, as we show in this paper, achieving this result is impossible (under the assumption that the simulator does not know the distribution of the messages committed to by the honest sender); this is in contrast to [23] where it is claimed that this definition is achievable. The concurrent-with-barrier composition that we adopted (following [13]) implies security under parallel composition while due to the barrier between commitment and decommitment phase, it is weaker than the fully concurrent definition of [23].

Decommitment Phase can be Interactive. Following [13] our definition is more general than the one of [23] since it allows also the decommitment phase to be interactive.

Honest Party Behaviour. We follow [23] in defining the behaviour of the honest receiver, i.e, R chooses the subset of commitments to be opened according to some distribution $\mathcal{I}$. To see why this definition makes sense, think about extractable commitments where the sender and receiver engage in many commitments of pairs of shares of a message but finally only one share per pair is required to be opened in the commitment phase.

Concerning the honest sender, we assume that R^* interacts with k independent senders, that are oblivious to each other, and play with input $\mathbf{b}[j]$, while [23] considers a single sender S^k who gets as input the complete k-bit string and plays k independent sessions with R^*. This variation is cosmetic only.

Comparison with the Definitions of [2,13]. In [2,13] the behaviour of the honest receiver is not explicitly defined, implying that the honest receiver always obtains all the openings. In order to be more general and to make SOA-secure commitments useful in more general scenarios, we deviate from this definition allowing the honest receiver to ask for the opening of a subset of the commitments. Moreover, the set of indexes I chosen by the (possibly malicious) receiver is explicitly given as input to the relation $\mathcal{R}$.

Summing up, the definition that we adopt mainly follows the one of [2,13] and is more general than the one of [23] in the fact that it allows interaction also during the decommitment phase, and provides concurrency-with-barrier that implies the definition of security under parallel composition. Moreover, our definition is more general than the one of [13] since it allows also the honest receiver to choose the commitments to be opened. However, our definition is weaker than the concurrent definition of [23] that however we show to be impossible to achieve (when the distribution of the messages committed by S is unknown to Sim).

3 Upper Bounds

3.1 $(3, 1)$-Round Scheme from Trapdoor Commitments

We present a construction for round-optimal SOA-secure commitment scheme based on BB use of trapdoor commitments. In particular we show that if 2-round (where the first round only serves for the receiver to send the public parameters) trapdoor commitment schemes exist[6], then a 3-round SOA-secure commitment scheme exists.

Roughly, the main idea of the protocol is to require the sender to commit to its private bit using a trapdoor commitment scheme and to make the trapdoor extractable to a black-box simulator. The goal is to allow the simulator to cheat in the opening phase without changing the transcript of the commitment phase. This is inspired by the techniques used in [17]. The parameters of the trapdoor commitment are generated by the receiver (if this was not the case then a malicious sender can cheat in decommitment phase using the trapdoor), and are made extractable through cut-and-choose techniques.

In more details, the protocol goes as follows. R runs the generation algorithm of the trapdoor commitment scheme (TGen) to compute the pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ of public and trapdoor parameters, and sends the public parameters to the sender. To guarantee the extraction of the trapdoor, we require that R provides $2n$ public parameters to S. S will use half of them to commits to n shares of its secret bit, while for the other half, it will ask the receiver to reveal the trapdoors associated. Then in the decommitment phase S simply opens the n commitments, and R computes the xor of the opened values.

To argue hiding in presence of SOA adversaries, we show the following simulator. Recall that, the malicious receiver can open many sessions and run many commitment phases concurrently (CwB definition). For each session, the simulator honestly commits to n random bits, and obtains n trapdoors. This is the main tread. When all commitment phases are completed, the simulator obtains from the adversary, the indexes of the sessions to be opened, and from the experiments, the actual bits to open. Next, it starts a rewinding thread for each session that needs to be open. In each rewinding thread, the simulator runs as in the main thread, except that it asks the adversary to open a different subset of trapdoors. Therefore, after an expected polynomial number of rewindings, it will obtain at least one *new* trapdoor for each session. One trapdoor will suffice to equivocate one share, and therefore to successfully open to any bit for that session.

Binding follows straight-forwardly from the binding of the trapdoor commitment scheme used as sub-protocol.

In the full version of this paper [18] we additionally show a $(4, 1)$-round constructions that is based on weak trapdoor commitment schemes.

The formal description of the construction is provided in Protocol 1. We denote by $\mathsf{TC} = (\mathsf{TGen}, \mathsf{S}_{\mathsf{TC}}, \mathsf{R}_{\mathsf{TC}}, \mathsf{S}, \mathsf{TFakeD})$ a trapdoor commitment scheme. We denote by $\langle \mathsf{S_{TC}}_i^{\bar{d}_i}, \mathsf{R_{TC}}_i^{\bar{d}_i} \rangle$ the i-th invocation of sub-protocol TC run with public key $\mathsf{pk}^{\bar{d}_i}$. Here d_i denotes the i^{th} challenge for the cut-and-choose, i.e., $\mathsf{S_{soa}}$ computes the trapdoor associated to the key pk^{d_i}, while it commits to the i^{th} share of the input using key $\mathsf{pk}^{\bar{d}_i}$ (for which the trapdoor will not be revealed).

[6] [20] is an example of a trapdoor commitment scheme where the public parameters pk are generated by the receiver and sent to the sender in the first round. Given pk, the commitment procedure is non-interactive.

Protocol 1. *[(3,1)-round SOA-Commitment]* /$\mathsf{SOACom}=(\mathsf{S_{soa}}, \mathsf{R_{soa}})$/
Commitment Phase

$\mathsf{R_{soa}}$: *For* $i = 1, \ldots, n$:
  1. $r_i^0, r_i^1 \xleftarrow{\$} \{0,1\}^n$; $(\mathsf{pk}_i^0, \mathsf{sk}_i^0) \leftarrow \mathsf{TGen}(r_i^0)$; $(\mathsf{pk}_i^1, \mathsf{sk}_i^1) \leftarrow \mathsf{TGen}(r_i^1)$;
  2. *send* $(\mathsf{pk}_i^0, \mathsf{pk}_i^1)$ *to* $\mathsf{S_{soa}}$;

$\mathsf{S_{soa}}$: *On input a bit* b. *Upon receiving* $\{\mathsf{pk}_i^0, \mathsf{pk}_i^1\}_{i\in[n]}$:
  1. *secret share the bit* b: *for* $i = 1, \ldots, n$: $b_i \xleftarrow{\$} \{0,1\}$, *such that* $b = (\bigoplus_{i=1}^n b_i)$;
  2. *for* $i = 1, \ldots, n$ *do in parallel:*
     – *send* $d_i \xleftarrow{\$} \{0,1\}$ *to* $\mathsf{R_{soa}}$;
     – *run* $\langle \mathsf{S_{TC}}_i^{\bar{d}_i}(\mathsf{pk}_i^{\bar{d}_i}, \mathtt{com}, b_i), \mathsf{R_{TC}}_i^{\bar{d}_i}(\mathsf{pk}_i^{\bar{d}_i}, \mathtt{rcv})\rangle$ *with* $\mathsf{R_{soa}}$;

$\mathsf{R_{soa}}$: *Upon receiving* $d_1, \ldots, d_n$: *if all commitment phases of protocol* TC *were successfully completed, send* $\{r_i^{d_i}\}_{i\in[n]}$ *to* $\mathsf{S_{soa}}$;

$\mathsf{S_{soa}}$: *Upon receiving* $\{r_i^{d_i}\}_{i\in[n]}$ *check consistency: for* $i = 1, \ldots, n$: $(\mathsf{pk}_i'^{d_i}, \mathsf{sk}_i'^{d_i}) \leftarrow \mathsf{TGen}(r_i^{d_i})$; *if* $\mathsf{pk}_i'^{d_i} \neq \mathsf{pk}_i^{d_i}$ *then abort.*

Decommitment Phase

$\mathsf{S_{soa}}$: *for* $i = 1, \ldots, n$: *run* $(\cdot, b_i') \leftarrow \langle \mathsf{S_{TC}}_i^{\bar{d}_i}(\mathtt{open}), \mathsf{R_{TC}}_i^{\bar{d}_i}(\mathtt{open})\rangle$ *with* $\mathsf{R_{soa}}$;

$\mathsf{R_{soa}}$: *If all opening phases were successful completed output* $b' \leftarrow \bigoplus_{i=1}^n b_i'$. *Otherwise, output* $\perp$.

The full proof of the following theorem can be found in the full version [18].

Theorem 1 (Protocol 1 is secure under selective opening attacks). *If* $\mathsf{TC} = (\mathsf{TGen}, \mathsf{S_{TC}}, \mathsf{R_{TC}}, \mathsf{TFakeD})$ *is a trapdoor commitment scheme, then Protocol 1 is a commitment scheme secure in presence of selective opening attacks.*

(3,1)-round SOA-secure Scheme based on NBB use of OWFs. We observe that, by instantiating Protocol 1 with the Feige-Shamir trapdoor commitment one can obtain a (3,1) SOA-secure scheme with non-black-box access to OWFs.

3.2 (3,3)-Round Scheme from One-Way Permutations

In this section we present a $(3,3)$-round SOA-secure commitment scheme based on BB use of any OWP. As a main ingredient, we use a $(3,1)$-round extractable commitment scheme, that we refer to as ExtCom. Very informally, extractable means that given black-box access to an adversarial sender, one can extract the bit played by the latter. A $(3,1)$-round extractable commitment can be constructed from BB access to any OWP. Such construction is pretty standard, thus for further details we refer the reader to the full version [18].

The idea behind the protocol is as follows. The sender and the receiver first engage in a coin-flipping protocol where the receiver commits to its random-string, then the sender sends its random string in the clear, and finally the receiver reveals its random string. Simultaneously, the sender commits to its input bit b, n pairs of times (with the two commitments in each pair indexed by 0 and 1). In the decommitment phase, at the completion of the coin-flipping protocol, the sender opens only one of the commitments in each pair according to the outcome of the coin-flipping.

To allow for simulation (while arguing hiding), the commitment of the receiver in the coin-flipping protocol is implemented via extractable commitment scheme, so that the simulator can extract the receiver's string in the commitment phase itself. Furthermore, we require that the sender sends its random string for the coin-flipping only in the decommitment phase; by the beginning of the decommitment phase, the simulator will have received the bit b to open to, and this gives the simulator an opportunity to craft its random string to point to the commitments of b. To see why, first note that if the simulator somehow knows the receiver's random string before it sends its own, then it can easily open the commitment to either 0 or 1: in each pair, it just commits to 0 in one of the commitments and 1 in the other. Then, with the knowledge of the receiver's random string and the bit b, it can craft its own random string such that the xor with the string of R points to the commitments of b. Since the receiver commits via an extractable commitment scheme, the simulator is able to extract the receiver's random string and hence is able to equivocate in the opening phase. Furthermore, as it will appear more clearly in the protocol, since the sender would send its commitments (resp., decommitments) always *after* it receives commitments (resp., decommitments) from the receiver, we require that the sender's $2n$ commitments to its input bit are implemented via extractable commitment scheme so that we avoid malleability issues that may compromise the binding property.

We prove binding by reducing it to the binding property of ExtCom (due to the ExtCom commitments played by $\mathsf{S}_{\mathsf{soa}}$) and to the computational hiding property of ExtCom (due to the ExtCom commitments played by $\mathsf{R}_{\mathsf{soa}}$). At a high level, we show that if an adversarial sender breaks binding, then it should have been able to bias outcome of the coin-flipping by predicting the randomness committed to by the receiver using ExtCom, before the sender sends its own ExtCom commitments. Then in the reduction, we make use of this fact to break computational hiding of ExtCom.

We now provide formal specification of our protocol. Let ExtCom be a $(3,1)$-round extractable commitment scheme. In the following we denote by $\langle {\mathsf{S}_{\mathsf{ext}}}^{i}(\mathtt{com}, a_i)$, ${\mathsf{R}_{\mathsf{ext}}}^{i}\ (\mathtt{rcv})\rangle$ the i-th of the n parallel executions of the extractable commitment scheme run by $\mathsf{R}_{\mathsf{soa}}$ to commit to its random string for coin-flipping, while we denote by $\langle {\mathsf{S}_{\mathsf{ext}}}^{i,\sigma}(\mathtt{com}, b), {\mathsf{R}_{\mathsf{ext}}}^{i,\sigma}(\mathtt{rcv})\rangle$ the commitment in position σ of the i-th pair (among the n pairs) of parallel executions run by $\mathsf{S}_{\mathsf{soa}}$ to commit to its input b.

Protocol 2. *[(3,3)-round SOA-Commitment] [*$\mathsf{SOACom}{=}(\mathsf{S}_{\mathsf{soa}}, \mathsf{R}_{\mathsf{soa}})$*]*
Commitment Phase

$\mathsf{R}_{\mathsf{soa}}$: *For* $i = 1, \ldots, n$ *do in parallel:*

1. $a_i \xleftarrow{\$} \{0,1\}$;
2. *run* $\langle {\mathsf{S}_{\mathsf{ext}}}^{i}(\mathtt{com}, a_i), {\mathsf{R}_{\mathsf{ext}}}^{i}(\mathtt{rcv})\rangle$ *with* $\mathsf{S}_{\mathsf{soa}}$;

$\mathsf{S}_{\mathsf{soa}}$: *on input a bit b. For $i = 1, \dots, n$ do in parallel:*
   1. run $\langle {\mathsf{S}_{\mathsf{ext}}}^{i,0}(\mathtt{com}, b), {\mathsf{R}_{\mathsf{ext}}}^{i,0}(\mathtt{rcv})\rangle$ with $\mathsf{R}_{\mathsf{soa}}$;
   2. run $\langle {\mathsf{S}_{\mathsf{ext}}}^{i,1}(\mathtt{com}, b), {\mathsf{R}_{\mathsf{ext}}}^{i,1}(\mathtt{rcv})\rangle$ with $\mathsf{R}_{\mathsf{soa}}$;

Decommitment Phase

$\mathsf{S}_{\mathsf{soa}}$: *If all extractable commitments played with $\mathsf{R}_{\mathsf{soa}}$ are successfully completed, send $d \overset{\$}{\leftarrow} \{0,1\}^n$ to $\mathsf{R}_{\mathsf{soa}}$;*
$\mathsf{R}_{\mathsf{soa}}$: *Open all commitments:*
   for $i = 1 \dots, n$: run $\langle {\mathsf{S}_{\mathsf{ext}}}^{i}(\mathtt{open}), {\mathsf{R}_{\mathsf{ext}}}^{i}(\mathtt{open})\rangle$ with $\mathsf{S}_{\mathsf{soa}}$;
$\mathsf{S}_{\mathsf{soa}}$: *If all openings provided by $\mathsf{R}_{\mathsf{soa}}$ are valid, for $i = 1, \dots, n$:*
   1. $\sigma_i \leftarrow d_i \oplus a_i$;
   2. run $\langle {\mathsf{S}_{\mathsf{ext}}}^{i,\sigma_i}(\mathtt{open}), {\mathsf{R}_{\mathsf{ext}}}^{i,\sigma_i}(\mathtt{open})\rangle$ with $\mathsf{R}_{\mathsf{soa}}$;
$\mathsf{R}_{\mathsf{soa}}$: *If all the corresponding openings provided by $\mathsf{S}_{\mathsf{soa}}$ open to the same bit b, and if for every i, $\sigma_i = d_i \oplus a_i$, then output b. Otherwise, output $\perp$.*

Theorem 2 (Protocol 2 is secure under selective opening attacks). *If* ExtCom *is an extractable commitment scheme, then Protocol 2 is a commitment scheme secure in presence of selective opening attacks.*

Details on the proof can be found in the full version of this paper [18]. In [18] we also show a variation of Protocol 2, which yield a 5-round commitment scheme with non-interactive decommitment, and is based on OWPs.

4 Issues in Some of the Claims of [23]

In this section we point out some issues regarding some of the main results in [23].

Revisiting Proof of Theorem 3.3 in [23]. Theorem 3.3 in [23] claims that their $(4,1)$-round protocol is SOA-secure under parallel composition with BB use of OWPs. The protocol recalls the equivocal commitment scheme of [5]. There is a preamble for coin flipping followed by Naor's commitment. In the preamble, the receiver commits to a random string α using a non-interactive (therefore computationally hiding only) commitment scheme; the sender sends a random string β in the clear; and finally, the receiver opens its commitment. Then the sender sends a Naor's commitment computed on the output of the coin flipping. Theorem 3.3 in [23] claims that the resulting protocol is computationally hiding, computationally binding and SOA-secure under parallel composition. We observe that this construction suffers of issues in the proof of binding and SOA-security. Concerning binding, [23] claims that it follows from the same arguments of Naor's commitment [16]. However this would be the case only if the commitment used by the receiver is at least statistically hiding (as in [5]), which is not the case in the $(4,1)$-round construction of [23]. SOA-security is claimed to follow from the simulation strategy of Goldreich and Kahan [11]. The issue here is that, such simulation strategy does not apply to the selective opening setting where multiple sessions are played in parallel with possibly different abort probabilities. Similar issues have been pointed out in [22] on previous work on round-optimal concurrent zero knowledge with bare public keys. The same issue on simulatability holds for the $(t+3,1)$-round scheme. We remark that although the $(t+3,1)$-round scheme of [23] is not simulatable directly via the Goldreich-Kahan's simulation strategy, the author of [23], elaborated an alternative simulation strategy for the same protocol. See [24] for details.

Revisiting Proof of Theorem 3.5 in [23]. Theorem 3.5 of [23] claims that if a coin-flipping preamble implemented via the $\omega(\log(n))$-round preamble of [21], is followed by Naor's commitment, then the resulting protocol is an $\omega(\log(n))$-round scheme that is SOA-secure under fully-concurrent composition with BB use of OWPs. Moreover, Theorem 3.5 also applies to the strong definition where the same simulator must work with respect to all distribution of messages, including the ones selected by the adversary and unknown to the simulator.

According to their proof, the simulatability of the protocol follows from the PRS's simulation strategy [21]. Specifically, if the coin-flipping is implemented with the PRS preamble, the claim of [23] is that the PRS's simulator obtains the random string committed to by the receiver, by the end of the coin-flipping, and this values can be used by the SOA-simulator to equivocate.

However the oblivious strategy of [21] cannot be applied in the SOA-setting (and, as we prove in Theorem 3 there exists no simulation strategy that can be applied if the black-box simulator has no access to the message distribution).

To see why, first recall that the proof of concurrent zero knowledge of [21] (used by [23]) critically relies on the fact that, the probability that the simulator aborts, namely, it reaches the end of a preamble without solving the session, is negligible. Second, observe that in the setting of fully-concurrent SOA, the adversarial receiver adaptively selects the sessions to be opened, and the opening of one session can be interleaved with the preamble of another session. In particular, during the rewinding threads needed by the PRS's simulator, the adversary can ask the opening of sessions that were not asked in the main thread. The simulator could handle such sessions in two possible ways. For one, it can query the external oracle to obtain the messages for the new sessions requested by the adversary[7], and then proceed to the completion of the rewinding thread. This would lead to a deviation in the distribution of the sessions queried to the external oracle, since the number of queries made in the simulation will be larger compared to the real game. On the other hand, the simulator can simply abort the rewinding threads that require the opening of new sessions (and thus additional queries to the external party). However, this will contradict the necessary condition (i.e., the simulator should abort with negligible probability only) for the PRS strategy [21] to be applicable. We stress that the above observations crucially rely on the fact that the protocol of [23] is claimed to be SOA-secure in the

[7] Note that here we are critically considering the case in which the distribution is not known to the simulator, and therefore the only way to answer consistently for it is to query the oracle. If instead the distribution is known, the simulator could sample from the distribution and therefore manage in some way the opening of new sessions started during rewinding thread.

strong sense, namely, the simulator cannot sample from the distribution of the messages committed by the honest sender.

Revisiting Proof of Theorem 4.4 in [23]. Theorem 4.4 in [23] states that, there exists no $(3, 1)$-round commitment scheme that is SOA-secure even under parallel composition, when security is proved using a black-box simulator. The proof essentially assumes that the structure of the commitment phase is such that the sender speaks first. However, we argue that this assumption loses generality. In fact, we present a $(3, 1)$-round commitment scheme (Protocol 1) in which the receiver speaks first, such that security in the concurrent-with-barrier setting (that is strictly stronger than the parallel composition setting of [23]) is proved using a black-box simulator. We observe that, the proof of Theorem 4.4. of [23], however, can be used to show impossibility of 2-round SOA-secure protocols (when security is proved via black-box simulation).

5 Impossibility of Strong Fully-Concurrent SOA-Security with Black-Box Simulation

The protocols presented in this paper achieve security under concurrent-with-barrier composition in the "strong" sense, that is, assuming that the simulator cannot sample from the distribution of the messages committed to by the sender. The last question to answer is whether there exist protocols that actually achieve the definition of security under strong fully-concurrent composition (as defined in [23]), or if the strong concurrent-with-barrier security definition is the best one can hope to achieve (when black-box simulation is taken into account). In this section we show that in contrast to the claim of Theorem 3.5 of [23], the strong fully-concurrent security definition of [23] is impossible to achieve. This holds regardless of the round complexity of the protocol [8] and of the black-box use of cryptographic primitives. Under the assumption that OWFs exist, the only requirements that we use for the impossibility is that the simulator is black-box and does not know the distribution of the messages committed by the sender. Both requirements are already specified in the strong fully concurrent security definition of [23].

Theorem 3. *If OWF exists, then no commitment scheme can be strong SOA-secure under fully-concurrent composition.*

Proof Idea. The proof consists in adapting a proof provided by Lindell in [15]. [15] shows that, there exist functionalities, for which proving that a protocol is secure under m-concurrent composition using a black-box simulator, requires that the protocol has at least m rounds. As corollary it holds that, for such functionalities, unbounded concurrency proved using a black-box simulator is impossible to achieve. Such a theorem cannot be directly applied to the case of SOA-secure

[8] This is therefore different from the case of concurrent zero knowledge [4,21].

commitments since it is provided only for two functionalities in which both parties have private inputs, such as, blind signatures and OT functionalities. In the setting of SOA-secure commitments the receiver has no private input and there is no ideal functionality involved. For our setting, we observe that, the fact that simulator cannot sample from the message distribution, means that it needs to access to an oracle for that. In our proof, we convert the role of the oracle into the role of the ideal functionality, and when deriving the contradiction we do not break the privacy of the receiver but the binding of the protocol.

The proof is based on the following two observations. First of all, since the simulator is black-box, the only advantage that it can exploit to carry out a successful simulation, is to rewind the adversary. Moreover, rewinds must be effective, in the sense that upon each rewind, the simulator must change the transcript in order to "extract" information from the adversary (obviously, if the transcript is not changed, then the rewind is useless). Second, in the strong SOA setting, the malicious receiver chooses the sessions to be opened, adaptively on the transcript seen so far, and the simulator can obtain the correct messages to be opened, only by querying the oracle. Changing the transcript in a rewinding attempt, may yield the adversary to change the sessions asked for opening (in particular to open sessions that were not asked in the main thread) and in turn, it may require the simulator to make additional queries to the oracle. Such additional queries are caused only by the rewinding attempts and they do not appear in the real-world execution. However, the distribution of the set of sessions' indexes asked by Sim in the ideal game, should not be distinguishable from the one asked by the adversary in the real game. Thus, the idea of the proof is to show that there exists an adversarial receiver, that makes the rewinding attempts of any black-box Sim ineffective, unless Sim makes additional queries the oracle, and hence the set of sessions asked in the ideal game is distinguishable from the set asked in the real game. Then, the next step is to show that, if nonetheless there exists a simulator that is able to deal with such an adversary (without rewinding), then such a simulator can be used by a malicious sender to break the binding of the protocol. The formal proof can be found in the full version of this paper [18].

6 Application to cZK with Pre-processing

We show how to use SOA-secure commitment schemes to construct concurrent zero-knowledge (cZK) protocol with pre-processing by using OWFs only, therefore improving a previous result of [5]. We combine our $(3,1)$-round SOA-secure computationally binding scheme based on NBB use of OWFs with the use of the special $\mathcal{WIPoK}$ of [14]. The preprocessing takes 3 rounds and is composed by two subprotocols played in parallel. The first subprotocol is a coin-flipping protocol where the prover commits to a random string using the SOA commitment that ends with the 3rd round of the verifier. In the 3rd round the verifier also sends his random string and the xor of the two strings is the outcome of this subprotocol. The second subprotocol is a special $\mathcal{WIPoK}$ to prove that $x \in L$

or the output of the coin flipping is also the output of a PRG. Only two rounds of this subprotocol are played during the preprocessing.

At the end of the above preprocessing the prover knows the result of the coin flipping and later non-interactively can complete the proof by opening his SOA commitment and sending the last round of the special $\mathcal{WIP}oK$. The simulator will get advantage of the simulator of the SOA commitment to bias the outcome of all coin-flipping protocols, therefore being able to complete all proofs running the prover of the special $\mathcal{WIP}oK$ using the trapdoor witness. For more details the reader is referred to the full version of this paper [18].

Acknowledgments. Supported in part by NSF grants 0830803, 09165174,1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, European Commission through the FP7 programme under contract 216676 ECRYPT II, MIUR Project PRIN "GenData 2020" and Lockheed-Martin Corporation Research Award. This material is also based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

1. Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard Security Does Not Imply Security against Selective-Opening. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 645–662. Springer, Heidelberg (2012)
2. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
3. Blum, M.: How to Prove a Theorem So No One Else Can Claim It. In: Proceedings of the International Congress of Mathematicians, pp. 1444–1451 (1986)
4. Canetti, R., Kilian, J., Petrank, E., Rosen, A.: Black-box concurrent zero-knowledge requires omega∼(log n) rounds. In: STOC, pp. 570–579 (2001)
5. Di Crescenzo, G., Ostrovsky, R.: On Concurrent Zero-Knowledge with Preprocessing (Extended Abstract). In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 485–502. Springer, Heidelberg (1999)
6. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science, FOCS 1999, pp. 523–534. IEEE Computer Society (1999)
7. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.: Magic functions. J. ACM 50(6), 852–921 (2003)
8. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. In: Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, STOC, pp. 409–418. ACM (1998)
9. Feige, U., Shamir, A.: Zero Knowledge Proofs of Knowledge in Two Rounds. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 526–544. Springer, Heidelberg (1990)

10. Gennaro, R., Micali, S.: Independent Zero-Knowledge Sets. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 34–45. Springer, Heidelberg (2006)
11. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for np. J. Cryptology 9(3), 167–190 (1996)
12. Goyal, V., Lee, C.K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, pp. 51–60. IEEE Computer Society (2012)
13. Hofheinz, D.: Possibility and impossibility results for selective decommitments. J. Cryptology 24(3), 470–516 (2011)
14. Lapidot, D., Shamir, A.: Publicly Verifiable Non-interactive Zero-Knowledge Proofs. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 353–365. Springer, Heidelberg (1991)
15. Lindell, Y.: Bounded-concurrent secure two-party computation without setup assumptions. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC, pp. 683–692. ACM (2003)
16. Naor, M.: Bit commitment using pseudorandomness. J. Cryptology 4(2), 151–158 (1991)
17. Ostrovsky, R., Persiano, G., Visconti, I.: Simulation-Based Concurrent Non-malleable Commitments and Decommitments. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 91–108. Springer, Heidelberg (2009)
18. Ostrovsky, R., Rao, V., Scafuro, A., Visconti, I.: Revisiting lower and upper bounds for selective decommitments. IACR Cryptology ePrint Archive 2011, 536 (2011)
19. Pass, R., Wee, H.: Black-Box Constructions of Two-Party Protocols from One-Way Functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 403–418. Springer, Heidelberg (2009)
20. Pedersen, T.P.: Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992), `http://portal.acm.org/citation.cfm?id=646756.705507`
21. Prabhakaran, M., Rosen, A., Sahai, A.: Concurrent zero knowledge with logarithmic round-complexity. In: 43rd FOCS, pp. 366–375 (2002)
22. Scafuro, A., Visconti, I.: On Round-Optimal Zero Knowledge in the Bare Public-Key Model. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 153–171. Springer, Heidelberg (2012)
23. Xiao, D.: (Nearly) Round-Optimal Black-Box Constructions of Commitments Secure against Selective Opening Attacks. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 541–558. Springer, Heidelberg (2011)
24. Xiao, D.: On the round complexity of black-box constructions of commitments secure against selective opening attacks. Cryptology ePrint Archive, Report 2009/513 - Revision May 29, 2012 (2012), `http://eprint.iacr.org/`
25. Xiao, D.: Round-Optimal Black-Box Statistically Binding Selective-Opening Secure Commitments. In: Mitrokotsa, A., Vaudenay, S. (eds.) AFRICACRYPT 2012. LNCS, vol. 7374, pp. 395–411. Springer, Heidelberg (2012)