(Nearly) optimal black-box constructions of commitments secure against selective opening attacks

David Xiao
LRI
Université Paris-Sud
dxiao@lri.fr

October 20, 2009

Abstract

Selective opening attacks against commitment schemes occur when the commitment scheme is repeated in parallel (or concurrently) and an adversary can choose depending on the commit-phase transcript to see the values and openings to some subset of the committed bits. Commitments are secure under such attacks if one can prove that the remaining, unopened commitments stay secret.

We prove the following black-box constructions and black-box lower bounds for commitments secure against selective opening attacks:

1. For parallel composition, 4 rounds are necessary and sufficient to build computationally binding and computationally hiding commitments.
2. For parallel composition, $O(1)$-round statistically-hiding commitments are equivalent to $O(1)$-round statistically-binding commitments.
3. For concurrent composition, $\omega(\log n)$ rounds are sufficient to build statistically binding commitments and are necessary even to build computationally binding and computationally hiding commitments, up to $\log \log n$ factors.

Our lower bounds improve upon the parameters obtained by the impossibility results of Bellare *et al.* (EUROCRYPT '09), and are proved in a fundamentally different way, by observing that essentially all known impossibility results for black-box zero-knowledge can also be applied to the case of commitments secure against selective opening attacks.

In addition to the impossibility results mentioned above, we also rule out the existence of commitments with zero statistical binding error and receiver public-coin commitments for parallel composition.

Keywords: commitments, black-box lower bounds, zero knowledge, selective opening attacks, parallel composition, concurrent composition

1 Introduction

Commitment schemes have a wide array of applications in cryptography, one of the most notable being the construction of zero knowledge protocols [GMW86, BC87]. A problem that arises in the use of commitment schemes is whether their hiding property holds when composed in parallel: if some subset of the committed messages are opened, do the remaining unopened messages remain secure? This question arose early in the study of zero knowledge protocols, and whether they remain zero knowledge when composed in parallel. It is natural in other cryptographic contexts as well, whenever commitments are used as building blocks for protocols that might be then used in parallel (*e.g.* secure multi-party computation, etc.).

Although naively one might think that because commitments are hiding that no additional information should be leaked by composing them, nevertheless it is unknown how to prove that standard stand-alone commitments (*e.g.* [Nao89]) remain hiding when composed.

More formally, a selective opening attack on a commitment scheme allows a cheating receiver to interact in k parallel (or concurrent) commitments, and then ask the sender to open some subset $I \subseteq [k]$ of the commitments. The question is whether the unopened messages remain hidden in the following sense: is there a simulator strategy such that given black-box access to any cheating receiver strategy, the simulator can output a commit-phase transcript, a set $I \subset [k]$, and decommitments to $(b_i)_{i \in I}$ that is computationally indistinguishable from the output of the receiver with an honest sender?

In this paper we show that techniques both for constructions and lower bounds from the study of zero knowledge protocols can be applied to the study of commitments secure against selective opening attacks. We study the minimal requirements and round complexity needed to construct such commitments, and give solutions for commitments secure against selective opening attacks that are optimal or nearly optimal up to some small factors.

1.1 Our results

We let PAR denote parallel composition and CC denote concurrent composition. We let CB (resp. SB, PB) denote computational (resp. statistical, perfect) binding and CH (resp. SH) denote computational (resp. statistical) hiding. We give the following constructions:

Theorem 1.1. *The following hold:*

1. *There is a black-box reduction that uses one-way permutations to build* 4*-round* PAR-CBCH *commitments.*
2. *There is a black-box reduction that uses* t*-round stand-alone* SH *commitments to build* $(t+3)$*-round* PAR-SB *commitments.*
3. *There is a black-box reduction that uses* t*-round stand-alone* SH *commitments to build* $\omega(t \log n)$*-round* CC-SB *commitments.*

Assuming the proof of security for such a commitment scheme is given by a black-box simulator, we prove the following corresponding lower bounds:

Theorem 1.2 (Impossibility results, informal)**.** *The following hold relative to any oracle:*

1. *There is no* 3*-round* PAR-CBCH *scheme.*

2. *There is a black-box reduction that uses a $O(1)$-round PAR-SB commitment to build a $O(1)$-round statistically hiding commitment.*

3. *There is no $o(\log n/\log\log n)$-round CC-CBCH scheme.*

We stress that besides the constraint that the simulator be black-box, these results are otherwise *unconditional*. As a special case, Theorem 1.2 implies that no such commitments exist in the plain model (without oracles). But it is even stronger, implying that such commitments do not exist even in say the random oracle model (or stronger oracle models), where *a priori* one might have hoped to bypass the impossibility results in the plain model.

Combining the second item of Theorem 1.2 with the main theorem of [HHRS07], which proves that there is no black-box reduction that uses a one-way permutation to build a $o(n/\log n)$-round statistically hiding commitment, we obtain the following corollary:

Corollary 1.3. *There is no black-box reduction that uses a one-way permutation to build a $O(1)$-round PAR-SB commitment.*

In addition to the above impossibility results, we also prove:

Theorem 1.4. *Relative to any oracle, there exists no PAR-PB commitments nor receiver public-coin PAR-SB commitments.*

1.2 Comparison to previous constructions

Notions related to security against selective opening attacks have previously been studied in the literature. In particular, security against selective opening is very related to commitments that can be simulated in such a way that they can be opened in many ways. Such commitments have been studied under the name of chameleon blobs [BCC88, BCY89], trapdoor commitments [Fis01], or equivocable commitments [Bea96, DCIO98, DCO99]. Indeed, the constructions presented here use the equivocable commitment of [DCO99] as the basic tool.

The above notions may be stronger than security against selective opening. Security against selective opening was directly studied in [DNRS03, BHY09]. Bellare *et al.* [BHY09] give a construction of a scheme that is CC-SB secure, but this construction is non-black-box and requires applying a concurrent zero knowledge proof on a statement regarding the code implementing a one-way permutation. In contrast, all constructions presented in this paper are fully black-box.

Remark 1.5 (Equivalence of statistical hiding and statistical binding)**.** In this work we only study commitments with computational hiding. Concurrent with our work, Zhang *et al.* [ZCZ09] gave a black-box reduction that uses t-round stand-alone SH commitments and one-way permutations to construct $(t+3)$-round PAR-SH commitments. Their construction is an extension of a recent trapdoor commitment of Pass and Wee [PW09].

With Item 2 of Theorem 1.2, this implies that constant-round statistical hiding and constant-round statistical binding are *equivalent* via black-box reductions when security against selective opening attacks is required. This contrasts sharply with the stand-alone case, as 2-round statistically binding commitments are equivalent to one-way functions, but no black-box reduction can build $o(n/\log n)$-round statistically hiding commitment from one-way functions [HHRS07].

1.3 Comparison to previous lower bounds

Bellare *et al.* [BHY09] proved that non-interactive commitments and perfectly binding commitments secure against selective opening attacks cannot be based on *any* black-box cryptographic

assumption. Our lower bounds are stronger than theirs in that we can rule out 3-round rather than non-interactive commitments, as well as ruling out certain types of commitment with non-zero statistical binding error. However, our proof *technique* is incomparable to theirs.

Ways in which our lower bounds are stronger: first, the lower bounds of [BHY09] assume black-box access to a cryptographic primitive, and therefore do not apply to constructions based on *concrete assumptions* (*e.g.* factoring, discrete log, lattice problems) where one might hope to exploit the specific structure of those problems to achieve security. In contrast, our results immediately rule out all constructions in the plain model.

Second, the lower bounds of [BHY09] prove that non-interactive and perfectly binding commitments secure against selective opening attacks are impossible with respect to a very specific message distribution *that is defined in terms of a random oracle*. One could argue that the message distribution they consider is artificial and would not arise in applications of these commitments. In particular, it may suffice for applications to build commitments that are secure only for particular natural message distributions, such as the uniform distribution or the distributions encountered when using commitments to build zero knowledge proofs for **NP**. [BHY09] does not rule out the existence of commitments that are secure only for these message distributions, while our impossibility results do and in fact apply simultaneously to all message distributions satisfying what we argue are very natural constraints (see Remark 2.5). In particular, the results of [BHY09] also use the assumptions in Remark 2.5.

Ways in which our lower bounds are weaker: our results are weaker in the sense that they only apply to constructions with black-box simulators, *i.e.* we require that there exists a single simulator that works for all cheating receivers. The results of [BHY09] hold even for slightly non-black-box simulation techniques: they only require that for every cheating receiver oracle algorithm $(\mathsf{Rec}')^{(\cdot)}$ that accesses the underlying crypto primitive as a black-box, there exists an efficient oracle algorithm $\mathsf{Sim}^{(\cdot)}$ that accesses the underling crypto primitive as a black box that generates an indistinguishable transcript.[1]

1.4 Our techniques

Our constructions for the case of parallel composition are essentially the equivocable commitment scheme of [DCO99], while the case for concurrent composition follows in a straightforward way by combining the commitment of [DCO99] with the preamble from the concurrent zero knowledge proof of [PRS02].

Our lower bounds are proven by observing that most known lower bounds for zero knowledge (*e.g.* [GK90, CKPR03, HRVW09, PTW09]) extend naturally to the case of commitment schemes. Lower bounds for zero knowledge show that if a zero knowledge proof for L satisfies certain restrictions (*e.g.* 3 rounds, constant-round public coin [GK90], etc.), then $L \in$ **BPP**.

As was observed by [DNRS03, BHY09], plugging a t-round PAR-CBCH commitment into the GMW zero knowledge protocol for **NP** allows the zero knowledge property to be preserved under parallel repetition, thus allowing one to reduce soundness error while preserving zero knowledge and without increasing round complexity. Furthermore, the resulting protocol has $t+2$ rounds, remains public-coin if the commitment was receiver public-coin, and has a black-box simulator if the commitment had a black-box simulator. This immediately implies the following:

[1] Note however that because it still requires that the crypto primitive be treated as an oracle, their result is *not* strong enough to rule out techniques such as Barak's simulator for constant-round public-coin zero-knowledge [Bar01], because the simulator there includes a PCP encoding of the code of the underlying cryptographic primitive, and thus treats the *crypto primitive itself* (and not just the receiver algorithm calling the crypto primitive) in a non-black-box way.

Proposition 1.6 ([GK90], weak impossibility of PAR-CBCH, informal)**.** *There exist no black-box simulator non-interactive or constant-round public-coin PAR-CBCH commitment schemes.*

To see why, suppose there were such a scheme, then by the above discussion one would obtain either a 3-round or constant-round public-coin zero knowledge argument for **NP** with a black-box simulator that remains zero knowledge under parallel repetition. By [GK90], this implies that **NP** = **BPP**. But this contradicts the existence of a PAR-CBCH commitment schemes, since any commitment scheme can be broken by an **NP** oracle.

However, besides the fact that they apply to broader categories of commitments (*i.e.* 3-round vs. non-interactive and arbitrary round public-coin vs. constant-round public-coin), Theorem 1.2 and Theorem 1.4 are also stronger than Proposition 1.6 in that they hold *relative to any oracle*, and in the case of Item 2 of Theorem 1.2, is *black-box*. This is important for two reasons: first, Proposition 1.6 does not say whether such constructions are possible in the random oracle model, which is often used to prove the security of schemes for which we cannot prove security in the plain model. Second, if we want to compose our impossibility result with other black-box lower bounds, then our impossibility result had better also be black-box. For example, in order to obtain Corollary 1.3 we must combine Item 2 of Theorem 1.2 with the black-box lower bound of Haitner *et al.*. This is only possible if Item 2 of Theorem 1.2 is a black-box reduction, something that would not be true using the approach of the weak impossibility result Proposition 1.6.

To prove Theorem 1.2, we look more deeply into the zero knowledge lower bound strategy that was originally outlined by Goldreich and Krawczyk [GK90]. Their strategy involves taking the zero knowledge protocol and showing that it admits what we call a *simulator embedding*.[2]

The simulator embedding is a strategy that a cheating sender can use to embed its interaction with an honest receiver into an execution of the simulator, such that with non-negligible probability the outcome of the interaction that the receiver sees is computationally indistinguishable from what the simulator outputs when run with the honest receiver.

We show that if a commitment scheme admits a simulator embedding, then either binding and hiding must fail. The first observation is that one can define a receiver for the k-fold repetition such that the simulator does not know which subset of the k bits the receiver will ask to be opened, and therefore, in order to be able to open to the appropriate values, the simulator must create a commit-phase transcript that can be opened in many ways. But if a simulator embedding exists, then a cheating sender can embed his interaction with a real receiver into the execution of the simulator, producing a commit-phase transcript that can be opened many ways. This implies that either the receiver should reject when interacting with the simulator embedding, which contradicts the hiding property, or else the simulator embedding gives a strategy to break the binding property of the commitment. Furthermore, by directly arguing about the simulator embedding, we avoid the non-black-box step in Proposition 1.6 and therefore our results hold relative to any oracle.

2 Preliminaries

For a random variable X, we let $x \leftarrow_{\mathrm{R}} X$ denote a sample drawn according to X. For a set S, we let $x \leftarrow_{\mathrm{R}} S$ denote a uniform element of S. All security definitions in this paper are with respect to non-uniform adversaries; whenever we say that some statement holds against

[2] This is also sometimes called a "simulation-based" prover in the context of zero knowledge.

polynomial-time adversaries, it is understood that we mean families of polynomial-time non-uniform adversaries (or equivalently, polynomial-size circuits). Two families of random variables $(X_n)_{n\in\mathbb{N}}, (Y_n)_{n\in\mathbb{N}}$ over $\{0,1\}^n$ are computationally indistinguishable, or equivalently $X \approx_c Y$, if for all circuits C of size poly(n) it holds that $|\Pr[C(X)=1] - \Pr[C(Y)=1]| \leq n^{-\omega(1)}$.

2.1 Commitment schemes

A commitment scheme is a two-phase interactive protocol between a sender and a receiver. They are a digital analogue of locked safes: in the *commit phase*, the sender puts his message inside the safe, locks the safe, and sends it to the receiver without the key. Thus, after the commit phase the sender can only reveal the message he committed to (the commitment is binding), but without the key the receiver has no idea what that message is (the commitment is hiding). In the *opening* or *decommit* phase, the sender reveals the key to the receiver who can then learn the value of the message and be assured that it was exactly what the sender originally committed to.

It is well-known that a commitment can be statistically binding or statistically hiding (*i.e.* secure even against unbounded adversaries), but not both. Therefore, when referring to statistically binding commitments, it is understood that we expect the hiding property to be computational, and vice versa.

We now formally define commitments for single-bit messages; since we will be concerned with commitments that are parallelizable, multi-bit messages can be handled by just repeating the single-bit protocol in parallel or concurrently.

Definition 2.1. A t-round (stand-alone) commitment protocol is a pair of efficient algorithms Send and Rec. Given a sender input $b \in \{0,1\}$, we define:

1. The *commit phase* transcript is $\tau = \langle \mathsf{Send}(b;\omega_{\mathsf{Send}}), \mathsf{Rec}(\omega_{\mathsf{Rec}})\rangle$ where $\omega_{\mathsf{Send}}, \omega_{\mathsf{Rec}}$ are the random coins of the sender and receiver, respectively. The commit phase has at most t rounds of interaction.

2. The *decommit phase* transcript is (without loss of generally) simply Send sending $(b, \omega_{\mathsf{Send}})$ to Rec, who then checks that τ was correctly generated using $(b, \omega_{\mathsf{Send}}, \omega_{\mathsf{Rec}})$.

We will always assume that a commitment scheme is put in a canonical form, where each party alternates speaking. We assume the number of rounds is even. If the number of rounds is $2t$, then we label the sender's messages $\alpha_1, \ldots, \alpha_t$ and the receiver's messages $\beta_1, \ldots, \beta_t$, and we let $\alpha_{[i]} = (\alpha_1, \ldots, \alpha_i)$ and likewise for $\beta_{[i]}$. For a commitment protocol $(\mathsf{Send}, \mathsf{Rec})$, we write that the receiver's i'th response β_i is given by computing $\beta_{[i]} = \mathsf{Rec}(\alpha_{[i]};\omega)$ where $\alpha_{[i]}$ are the first i sender messages, and ω are the receiver's random coins.

Let k denote the number of parallel or concurrent repetitions of a commitment protocol. Let n denote the security parameter of the protocol. For stand-alone $(\mathsf{Send}, \mathsf{Rec})$, let Send^k denote the k-fold repeated sender (context will determine whether we mean parallel or concurrent composition). Let Rec^k denote the k-fold parallel receiver, and let Rec^k_Σ denote the k-fold concurrent receiver with schedule Σ. Underlined variables denote vectors of message bits (*e.g.* $\underline{b} \in \{0,1\}^k$) and plain letters with indices the bit at each coordinate (*e.g.* b_i is the i'th bit of $\underline{b}$).

2.1.1 Binding

Definition 2.2 (Binding)**.** A commitment scheme $(\mathsf{Send}, \mathsf{Rec})$ is computationally (resp. statistically) binding if for all polynomial-time (resp. unbounded) sender strategies Send', only

with negligible probability can Send' interact with an honest Rec to generate a commit-phase transcript τ and then produce ω, ω' such that both $(b, \omega), (1-b, \omega')$ are both consistent with τ. A scheme is *perfectly* binding if the above probability of cheating is 0.

It is straight-forward to prove that all the variants of the binding property are preserved under parallel/concurrent composition. Furthermore, for the case of statistical binding the error probability is amplified by repetition: if the statistical binding error for stand-alone case is $n^{-\omega(1)}$, then the statistical binding error for simultaneously breaking k instances is $2^{-\omega(k \log n)}$. Note this is not necessarily true for computational binding error.

2.1.2 Hiding under parallel composition

We only study the case of computational hiding (see Remark 1.5). In the following, $\mathcal{I} \subseteq [k]$ is a family of subsets of $[k]$, which denotes the set of legal subsets of commitments that the receiver is allowed to ask to be opened.

Definition 2.3 (Real-world interaction for k-fold parallel composition)**.** Sender input: $\underline{b} \in \{0,1\}^k$. $\mathsf{Send}, \mathsf{Rec}$ run k executions of the commit phase in parallel using independent random coins, obtaining $\tau^k = (\tau_1, \ldots, \tau_k)$. Given τ^k, an honest receiver picks a random set $I \leftarrow_{\mathrm{R}} \mathcal{I}$ and sends it to the sender, who then opens all the commitments for b_i for $i \in I$.

One could allow the honest receiver to pick $I \in \mathcal{I}$ according to any efficient strategy; we state it for I uniform in $\mathcal{I}$ for simplicity. Note that a malicious receiver may always pick I adversarially.

Definition 2.4 (Hiding under selective opening, parallel composition)**.** Let $\mathcal{I} \subseteq [k]$ be a family of subsets and $\underline{\mathcal{B}}$ be a family of message distribution over $\{0,1\}^k$ for all k. We say that a commitment protocol $(\mathsf{Send}, \mathsf{Rec})$ is black-box computationally (resp. statistically) hiding under selective opening for $(\mathcal{I}, \underline{\mathcal{B}})$ if for all $k = \mathrm{poly}(n)$, there exists an efficient simulator Sim_k such that the following holds for all polynomial-time (resp. unbounded) receiver strategies Rec'.

- Let $\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}' \rangle = (\tau^k, I, \{(b_i, \omega_i)\}_{i \in I})$ be the complete interaction between Rec' and the honest sender, including the commit-phase transcript τ^k, the subset I of coordinates to be opened and the openings $(b_i, \omega_i)_{i \in I}$.
- Let $(\mathsf{Sim}_k^{\mathsf{Rec}'} \mid \underline{b})$ denote the following: first, $\mathsf{Sim}_k^{\mathsf{Rec}'}$ interacts with Rec' and outputs a subset I of bits to be opened. Then Sim_k is given $\{b_i\}_{i \in I}$. Using this, Sim_k interacts with Rec' some more and outputs a commit-phase transcript τ^k, the set I, and the openings $\{(b_i, \omega_i)\}_{i \in I}$.
- It holds that $(\mathsf{Sim}_k^{\mathsf{Rec}'} \mid \underline{b}) \approx_c \langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}' \rangle$ where $\underline{b} \leftarrow_{\mathrm{R}} \underline{\mathcal{B}}$.

Remark 2.5. We will always require that $\mathcal{I}$ and $\underline{\mathcal{B}}$ be efficiently samplable and also satisfy:

1. $|\mathcal{I}| = n^{\omega(1)}$
2. $\Pr_{\underline{b}, \underline{b}' \leftarrow_{\mathrm{R}} \underline{\mathcal{B}}, I \leftarrow_{\mathrm{R}} \mathcal{I}}[\exists \{i_1, \ldots, i_{k/100}\} \in I \ s.t. \ b_{i_1} \neq b'_{i_1}, \ldots, b_{i_{k/100}} \neq b'_{i_{k/100}}] \geq 1/\mathrm{poly}(n)$

Property 1 says that if the receiver asks for a random set in $\mathcal{I}$ to be opened, then the sender cannot guess the set with noticeable probability. This restriction is natural because in many contexts if the sender can guess the set to be opened then it can cheat in the larger protocol where the commitment is being used (*e.g.* in a zero knowledge proof). Property 2 says that

with non-negligible probability, messages chosen from $\underline{\mathcal{B}}$ will look significantly different even if we only reveal the subset of bits contained in I, and is natural because otherwise the sender and receiver might as well run fewer instances of the commitment, since many of the bits are just constant. Here the choice of $k/100$ is arbitrary; even $\omega(\log k)$ would suffice for our results.

2.1.3 Hiding under concurrent composition

Definition 2.6 (Real-world interaction for k-fold concurrent composition)**.** Identical to the parallel case, except that the receiver has the power to schedule messages as he wishes, rather than sending them in parallel. In addition, we allow the receiver to pick the set I incrementally subject to the constraint that at the end, $I \in \mathcal{I}$. For example, the receiver can generate one commit-phase transcript, ask the sender to decommit that instance, then use this information in its interaction to generate the second commit-phase transcript, and so forth.

Definition 2.7 (Hiding under selective opening, concurrent composition)**.** Same as the parallel case, except that the simulator can incrementally ask for the values $(b_i)_{i\in I}$ before completing all commit-phase executions, subject to the constraint that at the end $I \in \mathcal{I}$.

Discussion of definitional choices: One could weaken Definition 2.6 to require that although all the commit-phase transcripts may be generated concurrently, the openings happen simultaneously. Indeed, this was the definition used in [BHY09]. We do not work with this weakening because it makes the definition not truly concurrent: by forcing all the openings to occur simultaneously, this "synchronizes" all the sessions.

2.1.4 Stronger definitions of hiding

Our definitions are chosen to be as weak as possible in order to make our lower bounds stronger. Nevertheless, our positive results also satisfy a stronger definition of security, where security should hold even if the receiver has auxiliary input, if we require security simultaneously for all input distributions $\underline{\mathcal{B}}$, and even if the simulator is forced to output the commit-phase transcript *before* it gets to see $I, (b_i)_{i\in I}$. For such a notion, we prepend STR to the name of the security property (*e.g.* STR-PAR-SB).

3 Constructions

Di Crescenzo and Ostrovsky ([DCO99], see also [DCIO98]) showed how to build an *equivocable* commitment scheme. Equivocable means that for every possible cheating receiver Rec', there exists a simulator that generates a commit-phase transcript that is computationally indistinguishable from one that would be generated by $\langle \mathsf{Send}, \mathsf{Rec}' \rangle$, but which the simulator can decommit to both 0 and 1. Equivocation implies STR-PAR-CBCH security and in fact they are equivalent in the stand-alone case. However, STR-PAR-CBCH explicitly requires the equivocation property to hold under parallel composition. Although it is not clear how to generically convert any stand-alone equivocable commitment to an equivocable commitment that is composable in parallel, the particular construction of Di Crescenzo and Ostrovsky can be composed by using a suitable preamble.[3]

[3]Notice that using [DCO99] means that the resulting commitment scheme cannot be SH, regardless of the preamble we choose. One could alternatively base our constructions on the trapdoor commitment of Pass and Wee [PW09]. Then, as observed by [ZCZ09], with the appropriate preamble one can also obtain STR-PAR-SH

The DO construction consists of a preamble, which is a coin-flipping scheme that outputs a random string, followed by running Naor's commitment based on OWF [Nao89] using the random string of the preamble as the receiver's first message. Depending on how the preamble is constructed, we get either a STR-PAR-CBCH, STR-PAR-SB, or STR-CC-SB commitment. Therefore, Theorem 1.1 follows from Theorem 3.3 and Theorem 3.5 below.

Protocol 3.1 ([DCO99, DCIO98, Nao89])**.** Security parameter: n. Sender's bit: b. Let $G : \{0,1\}^n \to \{0,1\}^{3n}$ be a PRG.

> **Preamble:** Use a coin-flipping protocol to obtain $\sigma \leftarrow_{\mathrm{R}} \{0,1\}^{3n}$.
> **Commit phase:** The sender picks random $s \leftarrow_{\mathrm{R}} \{0,1\}^n$ and sends $c = (\sigma \wedge b) \oplus G(s)$ (where we let $\sigma \wedge b$ denote the coordinate-wise AND of σ and b).
> **Decommit phase:** The sender sends b, s. Receiver checks that $c = (\sigma \wedge b) \oplus G(s)$.

We now present three different preambles that when used in the protocol above, provide STR-PAR-CBCH, STR-PAR-SB, and STR-CC-SB security, respectively.

Protocol 3.2 ([DCO99])**.** Preambles for STR-PAR-CBCH or STR-PAR-SB:

1. Using the non-interactive stand-alone CH commitment based on one-way permutations (to achieve STR-PAR-CBCH) or a t-round stand-alone SH commitment (to achieve STR-PAR-SB), the receiver sends a commitment to $\alpha \leftarrow_{\mathrm{R}} \{0,1\}^{3n}$.
2. The sender replies with $\beta \leftarrow_{\mathrm{R}} \{0,1\}^{3n}$.
3. The receiver opens α.
4. Output $\sigma = \alpha \oplus \beta$.

Theorem 3.3. ([DCO99]) *Protocol 3.1 with the STR-PAR-CBCH (resp. STR-PAR-SB) version of the preamble of Protocol 3.2 gives a STR-PAR-CBCH (resp. STR-PAR-SB) commitment.*

We sketch the proof of Theorem 3.3 in Appendix B for the sake of completeness.

Protocol 3.4 ([PRS02])**.** Preamble for CC-SB:

1. The receiver picks $\alpha \leftarrow_{\mathrm{R}} \{0,1\}^{3n}$ and for $\ell = \omega(\log n)$ picks $\alpha^0_{i,j} \leftarrow_{\mathrm{R}} \{0,1\}^{3n}$ for $i, j \in [\ell]$ and sets $\alpha^1_{i,j} = \alpha \oplus \alpha^0_{i,j}$. The receiver commits in parallel to $\alpha, \alpha^0_{i,j}, \alpha^1_{i,j}$ via a t-round statistically hiding commitment.
2. For each $j = 1$ to ℓ sequentially, do the following:
   (a) The sender sends $q_1, \ldots, q_\ell \leftarrow_{\mathrm{R}} \{0,1\}$.
   (b) The receiver opens the commitment to $\alpha^{q_i}_{i,j}$ for all $i \in [\ell]$.
3. The sender sends $\beta \leftarrow_{\mathrm{R}} \{0,1\}^{3n}$.
4. The receiver opens the commitment to $\alpha, \alpha^0_{i,j}, \alpha^1_{i,j}$ for all $i, j \in [\ell]$.
5. The sender checks that indeed $\alpha = \alpha^0_{i,j} \oplus \alpha^1_{i,j}$ for all $i, j \in [\ell]$. If so output $\sigma = \alpha \oplus \beta$, otherwise abort.

commitments (in addition to STR-PAR-CBCH, STR-PAR-SB, and STR-CC-SB commitments). However, since we are not discussing the STR-PAR-SH case here, we present our results using [DCO99], which is (somewhat) simpler to describe than [PW09].

Theorem 3.5 ([PRS02, Ros06]). *Protocol 3.1 using the preamble of Protocol 3.4 gives a CC-SB commitment.*

Proof. Binding is straightforward. For hiding, observe that this is the preamble of the concurrent zero knowledge proof of Prabhakaran *et al.* [PRS02]. They prove the following:

Theorem 3.6 ([PRS02], informal). *There is black-box simulator strategy that, given access to any efficient receiver for Protocol 3.4 with any concurrent scheduling, with high probability outputs for every session a string* α before *Step* 3 *such that the receiver opens to* α *in step* 5.

Namely, [PRS02] show that by using an appropriate rewinding schedule, the simulator can obtain the value of α in *all* of the concurrent executions before the sender is supposed to send β, regardless of how the receiver schedules the messages. Once the simulator knows α, it can choose $\beta = \alpha \oplus G(s_0) \oplus G(s_1)$ and later send commitment $c = G(s_0)$. It can then open to 0 by revealing s_0 and to 1 by revealing s_1. Therefore, regardless of the value of $I, \underline{b}$, the simulator is able to open $(b_i)_{i \in I}$ correctly. This is made rigorous by a slight generalization of Theorem 5.2 of [Ros06], which considers languages with challenge-response interactive proofs, to general challenge-response protocols such as the [DCO99] protocol. ■

4 Optimality of constructions

The Goldreich-Krawczyk technique revisited. [GK90] implicitly used simulator embeddings, as we shall shortly define, to prove lower bounds on zero knowledge. We use their ideas to prove lower bounds on commitments. For a pair of algorithms $T = (T_{com}, T_{decom})$, let $\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}$ denote the transcript generated as follows. First $\langle T_{com}, \mathsf{Rec}^k \rangle$ outputs a commit-phase transcript. Then we append Rec^k's choice of subset $I \in \mathcal{I}$ (notice up to here everything is generated independently of $\underline{b}$). Finally, append the opening $T_{decom}(\underline{b}, I) = (b_i, \omega_i)_{i \in I}$. In general T may abort during its execution, so we let $\mathsf{NoAbort}_T$ denote the event that it does not abort. $(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ denotes $\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}$ conditioned on T not aborting.

Definition 4.1 (Simulator embeddings). We say that $T = (T_{com}, T_{decom})$ form a (k, ε)-simulator embedding for $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$ if when T interacts with a k-fold parallel honest receiver Rec^k, $\Pr[\mathsf{NoAbort}_T] \geq \varepsilon$, and $\left(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}, \mathsf{NoAbort}_T\right) \approx_c \left((\mathsf{Sim}_k)^{\mathsf{Rec}^k} \mid \underline{b}\right)$. We say T is a k-simulator embedding if it is a $(k, 1/\mathrm{poly}(n))$-simulator embedding.

Using simulator embeddings to break binding and hiding. Here we show that secure commitments cannot have simulator embeddings. In the next few sections, we will show that certain kinds of commitments must admit simulator embeddings, which, combined with the following theorems, imply that those kinds of commitments cannot be secure.

Theorem 4.2. *For all* $\underline{\mathcal{B}}, \mathcal{I}$*, there is an efficient black-box reduction that, given any commitment scheme* $(\mathsf{Send}, \mathsf{Rec})$ *with* k*-fold simulator* Sim_k*, and any candidate* k*-simulator embedding* $T = (T_{com}, T_{decom})$*, either breaks the binding of* $(\mathsf{Send}, \mathsf{Rec})$ *or distinguishes between* $(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ *and* $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$.

Proof. We construct a malicious sender such that either this sender breaks binding, or else the honest k-fold receiver Rec^k can distinguish between $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$ and $\langle T, \mathsf{Rec}^k \rangle$.

Constructing the malicious sender: the malicious sender inserts a single instance of the commitment into a k-fold parallel repetition as follows

1. The malicious sender picks a random j.

2. The malicious sender interacts with the real receiver by inserting the real receiver's messages at index j in a k-fold parallel repetition, and for all coordinates $i \neq j$ the malicious sender internally generates receiver messages by emulating the honest receiver algorithm. The malicious sender generates responses to the receiver by running the simulator embedding T_{com} and sending the message in T_{com}'s response that corresponds to the j'th instance.

3. If T_{com} does not abort, the malicious sender successfully generates a commit-phase transcript distributed according to $\langle T_{com}, \mathsf{Rec}^k \rangle$. The malicious sender picks a random $I \leftarrow_{\mathrm{R}} \mathcal{I}$ to be opened.

4. If $j \notin I$, the malicious sender aborts. Otherwise, it independently picks two $\underline{b}, \underline{b}' \leftarrow_{\mathrm{R}} \underline{\mathcal{B}}$, and runs $T_{decom}(\underline{b}, I)$ to obtain a decommitment for $(b_i)_{i \in I}$ and runs $T_{decom}(\underline{b}', I)$ to obtain openings for $(b'_i)_{i \in I}$. In particular, the malicious sender obtains openings for b_j and b'_j.

Analyzing the malicious sender: By hypothesis $\Pr[\mathsf{NoAbort}_T] = p_T \geq 1/\mathrm{poly}(n)$, and in the following we condition on $\mathsf{NoAbort}_T$. Since real interactions create valid transcripts that the honest receiver algorithm Rec^k always accepts, the hiding property implies that $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$ must output valid commitments with overwhelming probability. There are two cases: either Rec^k rejects $(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ with noticeable probability, or it accepts with overwhelming probability. In the first case we are done, so let us assume the second case.

In the second case, both $T_{decom}(\underline{b}, I)$ and $T_{decom}(\underline{b}', I)$ are valid openings with overwhelming probability. By the constraints of Remark 2.5 we know that $\Pr_{\underline{b}, \underline{b}', I}[\exists i \in I,\ b_i \neq b'_i] = p_{\mathcal{I}, \underline{\mathcal{B}}} \geq 1/\mathrm{poly}(n)$. Therefore if $T_{decom}(\underline{b}, I)$ and $T_{decom}(\underline{b}', I)$ both produce valid decommitments then it must produce with probability $p_{\mathcal{I}, \underline{\mathcal{B}}}$ valid decommitments to 0 and 1 at some coordinate j. Since the sender picked at random the coordinate j that contains the real interaction, with probability $1/k$ it chooses $i = j$ and therefore produces decommitments for both 0 and 1 in an interaction with the real receiver, breaking binding. Taking into account all of the conditionings, the overall probability of this cheating sender breaking binding is $p_T \cdot p_{\mathcal{I}, \underline{\mathcal{B}}}/k$, which is non-negligible. ■

Theorem 4.3. *For any $\mathcal{I}, \underline{\mathcal{B}}$ and relative to any oracle, for any SB commitment* $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$*, and for any (possibly inefficient) candidate $(k, 2^{-k})$-simulator embedding $T = (T_{com}, T_{decom})$, it holds that Rec^k can distinguish between $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$ and $(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}, \mathsf{NoAbort}_T)$.*

Proof. **Building simulator embedding T:** the idea is the same as the proof of Theorem 4.2, with two differences: first, we cannot break binding, since it is statistical. Second, instead of picking one index at random, the cheating sender tries to break binding in $k/100$ instances simultaneously by picking a random subset $U \subseteq [k]$ of size $|U| = k/100$. Then, the cheating sender continues as in Theorem 4.2, by forwarding messages in the chosen coordinates to the honest receiver while emulating honest receiver responses in the remaining unchosen coordinates, and generating sender messages according to $T = (T_{com}, T_{decom})$. If the sender obtains decommitments to all coordinates in U, it forwards them to the real receiver, otherwise it aborts.

Analyzing simulator embedding T: by hypothesis, $\Pr[\mathsf{NoAbort}_T] \geq 2^{-k}$. As before, Rec^k must accept $\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b}$ with overwhelming probability, and now we will show that Rec^k must reject the output of $(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b})$ (*i.e.* it says that the output is an invalid commitment) with noticeable probability.

Suppose for the sake of contradiction that Rec^k accepts the output of $(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}, \mathsf{NoAbort}_T)$. By Remark 2.5 we have that with probability $p_{\mathcal{I}, \underline{\mathcal{B}}} \geq 1/\mathrm{poly}(n)$ over $\underline{b}, \underline{b}' \leftarrow_{\mathrm{R}} \underline{\mathcal{B}}$ and $I \leftarrow_{\mathrm{R}} \mathcal{I}$,

there exists $k/100$ coordinates on which $\underline{b}, \underline{b}'$ differ. With probability $1/\binom{k}{k/100} \geq (1/100e)^{-k/100} \geq 2^{-k}$, U is contained in the set of coordinates where $\underline{b}, \underline{b}'$ differ.

As in the proof of Theorem 4.2, this implies that T can break binding in $k/100$ coordinates with probability $2^{-k} \cdot p_{\mathcal{I},\underline{\mathcal{B}}} \cdot 2^{-k} \gg 2^{-\omega(k \log n)}$. On the other hand, by amplification of the statistical binding error, the probability that any sender breaks binding on $k/100$ instances is at most $2^{-\omega(k \log n)}$, but this is a contradiction. ■

4.1 Impossibility results for parallel composition

Theorem 4.4. *For all $\mathcal{I}, \underline{\mathcal{B}}$ and relative to any oracle, there exists no 3-round PAR-CBCH commitment protocol secure for $(\mathcal{I}, \underline{\mathcal{B}})$.*

Proof sketch. We give a black-box reduction that takes any 3-round PAR-CBCH commitment $(\mathsf{Send}, \mathsf{Rec})$ with simulator Sim_k and builds an efficient k-simulator embedding for $(\mathsf{Send}, \mathsf{Rec})$. By Theorem 4.2, this is a contradiction, and so 3-round PAR-CBCH commitment do not exist. Our simulator embedding $T = (T_{com}, T_{decom})$ follows the lower bounds in [GK90].

Observations and simplifying assumptions about Sim_k: A 3-round PAR-CBCH actually has 5 rounds if we include the opening phase. Namely, it has the following structure: the sender sends a message α_1, the receiver responds with β_1, the sender sends α_2, the receiver sends a set $I \in \mathcal{I}$ to be opened, and the sender responds with an opening. Notice that the honest receiver's second message is a uniform $I \in \mathcal{I}$, and is independent of β_1.

We assume *w.l.o.g.* that **(1)** Sim_k makes exactly $p(n) = \mathrm{poly}(n)$ queries to its receiver black box, **(2)** that all its queries are distinct, and **(3)** that Sim_k always outputs a transcript τ^k that consists of queries it made to the receiver and the corresponding receiver responses (see *e.g.* [GK90] for why these assumptions are *w.l.o.g.*).

Building simulator embedding T:

1. T_{com} picks $u_1, u_2 \leftarrow_{\mathrm{R}} [p(n)]$ and a $p(n)$-wise independent hash function h.

2. T_{com} internally runs the simulator:

   - For the simulator's u_1, u_2'th queries, T_{com} forwards them to the real receiver and forwards the receiver's responses to the simulator.
   - For all other queries, T_{com} answers according to Rec_h, which behaves as follows. Given a query $\alpha_{[i]}$, Rec_h first computes $h(\alpha_{[i]})$. If $i = 1$ then Rec_h returns $\mathsf{Rec}^k(\alpha_1; h(\alpha_1))$. If $i = 2$ then Rec_h uses $h(\alpha_{[2]})$ to return a random $I \leftarrow_{\mathrm{R}} \mathcal{I}$.

   Note that because the honest receiver's choice of $I \in \mathcal{I}$ is independent of its first message β_1, there is no need to ensure consistency between Rec_h's response for first queries and for second queries. Indeed, this is why our result applies to 3-round and not just non-interactive commitments, as a naive application of [GK90] might at first suggest.

3. At some point Sim_k requests that a subset I of bits be revealed. T_{com} checks to see if I was the answer that the real receiver gave in response to the u_1, u_2'th queries. If not, T_{com} aborts.

4. Assuming T_{com} did not abort, then in the opening phase, T_{decom} receives $\underline{b}$ and must compute an opening of $(b_i)_{i \in I}$. To do so, T_{decom} simply feeds $I, (b_i)_{i \in I}$ to the simulator and outputs what the simulator outputs.

Analyzing simulator embedding T:

Claim 4.5. *Let I be the subset that Sim_k asks to be opened in T_{com}'s emulation of Sim_k. With overwhelming probability, Sim_k made an earlier query $\alpha_{[2]}$ that was answered (either by Rec_h or by the real receiver) with I.*

Proof. The claim follows from property (3) that we assumed above about Sim_k. For suppose the claim were false, then in order to satisfy property (3), Sim_k would need to find $\alpha'_{[2]}$ such that $\mathsf{Rec}_h(\alpha'_{[2]}) = I$ or the real receiver answers $\alpha_{[2]}$ with I. But because Rec_h chooses a set in $\mathcal{I}$ at random using h, the probability that it picks I is $1/|\mathcal{I}| = n^{-\omega(1)}$. Similarly, because the real receiver chooses a set in $\mathcal{I}$ at random, the probability it picks I is $n^{-\omega(1)}$. Because Sim_k runs in time poly(n) it has negligible chance of finding any such $\alpha'_{[2]}$. ■

Therefore, when Sim_k outputs I, by Claim 4.5, I corresponds to *some* previous query. [GK90] proved that I corresponds to the u_1, u_2'th queries with non-negligible probability $1/p(n)^2$, and therefore T does not abort with non-negligible probability. Furthermore, as observed by [GK90], conditioned on T not aborting, the receiver's view of the final transcript $(\tau^k, I, (b_i, \omega_i)_{i\in I})$ is identically distributed to $(\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b})$, which by hiding is indistinguishable from $\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}_h\rangle = \langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}^k\rangle \approx_c (\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$, where the equality follows from the definition of Rec_h, and the indistinguishability from the hiding property. ■

Using similar ideas, Theorem 1.4 can be proved by constructing simulator embeddings from the appropriate commitment (for receiver public-coin protocols, the embedding we use is implicit in [PTW09]). These results are deferred to the Appendix B.2.

4.2 PAR-SB commitments imply SH commitments

To prove Item 2 of Theorem 1.2, we show that PAR-SB commitments can be used in a black-box way to generate an entropy gap between real and accessible entropy [HRVW09]. Then we apply the transformation of [HRVW09] that converts an entropy gap into a statistically hiding commitment. We defer the definitions of real min-entropy and accessible max-entropy to the appendix.

Theorem 4.6. *For all $\mathcal{I}, \underline{\mathcal{B}}$, there is a black-box reduction that, given any $O(1)$-round $(\mathsf{Send}, \mathsf{Rec})$ that is PAR-SB secure for $(\mathcal{I}, \underline{\mathcal{B}})$, constructs a $O(1)$-round statistically hiding commitments.*

Proof sketch. Assume without loss of generality that Rec^k sends all his random coins at the end of the opening phase, and that Rec uses m random coins in a single stand-alone instance.

Claim 4.7. *The receiver in $(\mathsf{Send}^k, \mathsf{Rec}^k)$ has real min-entropy at least $km(1 - 1/k^{1/3})$. There is a black-box reduction that uses the hiding property of $(\mathsf{Send}^k, \mathsf{Rec}^k)$ to show that Rec^k has context-independent accessible max-entropy $\leq km - k/4$.*

Assuming $(\mathsf{Send}^k, \mathsf{Rec}^k)$ is hiding and for sufficiently large k, Claim 4.7 implies there is an entropy gap, and so the theorem follows by combining it with the black-box construction of statistically hiding commitments from entropy gaps given by Lemmas 6.7, 4.7, and 4.18 of [HRVW09].[4] ■

[4] Our result is limited to constant-round protocols because there is no way to know which rounds of the protocol contain high entropy. This means that in [HRVW09]'s transformation from entropy gap to statistically hiding commitment, one needs to invoke an "equalizing lemma" (Proposition 3.9 of [HRVW09]), which blows up any super-constant round complexity by an amount that we cannot tolerate.

Proof sketch of Claim 4.7. The first part of the claim concerning real min-entropy follows from the definitions and amplification by parallel repetition (Proposition 3.8 in [HRVW09]). For the second part, we give a black-box construction that uses any A^* (and predicate $\mathsf{success}$, see Definition A.2) that samples high context-independent max-entropy to construct a $(k, 2^{-k})$-simulator embedding T such that a distinguisher for T can be used to break the hiding of $(\mathsf{Send}^k, \mathsf{Rec}^k)$. Applying Theorem 4.3 gives such a distinguisher for T, and therefore such A^* cannot exist.

Simplification: accessing full entropy: for the purposes of this sketch, we will think of A^* as accessing *full* entropy. Namely, A^* is an algorithm such that $\mathsf{A}^*(\alpha_{[i]}; s_1, \ldots, s_i)$ outputs $(\beta_1, \omega_1), \ldots, (\beta_i, \omega_i)$, where for each $j \in [i]$, A^* uses random coins s_j to sample random coins ω_j for Rec^k that are drawn *uniformly* from the set $\{\omega \mid \mathsf{Rec}^k(\alpha_{[j-1]}; \omega) = \beta_{[j-1]}\}$, and then computes $\beta_{[j]} = \mathsf{Rec}^k(\alpha_{[j]}; \omega_j)$.

Of course accessing full entropy and accessing high entropy can be very different. Nevertheless, [HRVW09] prove that any A^* accessing sufficiently high entropy behaves similarly to accessing full entropy when it comes to hiding, binding, and correctness (*i.e.* generating transcripts that the honest receiver would accept), so we will think of A^* as accessing full entropy to simplify our presentation here. In particular, notice this means that running Send^k with A^* produces a transcript that is distributed identically to $\langle \mathsf{Send}^k, \mathsf{Rec}^k \rangle$. For a treatment of the case where A^* only accesses high max-entropy, see Appendix B.2.2.

Building simulator embedding T: by Theorem 4.3 it suffices to construct a $(k, 2^{-k})$-simulator embedding $T = (T_{com}, T_{decom})$, which we do as follows. Let $p(n)$ be an upper bound on the number of queries that Sim_k makes.

1. T_{com} picks h at random from a family of $p(n)$-wise independent hash functions. T_{com} also picks a random subset $U \subseteq [p(n)]$ of size $|U| = t$ at random. T_{com} stores a table (initially empty) that associates some strings with every simulator query.

2. T_{com} internally executes the simulator Sim_k. Let Sim_k's j'th query be denoted $\alpha_{[i]}$. First T_{com} looks up $s_{[i-1]}$ corresponding to $\alpha_{[i-1]}$ in its table (or aborts if no such entry exists).

   - If $j \in U$, T_{com} forwards this to the real receiver, which replies with a response β_i. T_{com} samples s_i uniformly conditioned on the last output of $\mathsf{A}^*(\alpha_{[i]}; s_0, \ldots, s_i)$ being (β_i, ω_i) (for some ω_i). (Note this sampling is inefficient, and therefore T is inefficient.)
   - For $j \notin U$, T_{com} samples uniform s_i and computes (β_i, ω_i) the last output of $\mathsf{A}^*(\alpha_{[i]}; s_{[i]})$ and adds an entry into its table associating $s_{[i]}$ with $\alpha_{[i]}$.

3. Sim_k outputs a subset I to be opened. T_{com} verifies that I was the real receiver's response to a query in U, and that the query consists only of simulator queries / receiver responses that were also in U, or else T_{com} aborts.

4. Then T_{decom} computes the opening by simply forwarding $I, \underline{b}$ to the simulator and outputting whatever the simulator outputs.

Analyzing simulator embedding T: set $k \gg t$. Then a natural extension of Claim 4.5 holds, stating that the simulator cannot find two transcripts such that the receiver asks the same subset I to be opened for both. Combined with the observation of [HRVW09, GK90] that the simulator outputs a transcript consisting exclusively of queries/responses in U with probability $1/p(n)^t \gg 2^{-k}$, this implies that T does not abort with probability $> 2^{-k}$. Furthermore, they observe that by definition of T and Rec_h,

$$(\langle T, \mathsf{A}^* \rangle \mid \underline{b}, \mathsf{NoAbort}_T) = (\mathsf{Sim}_k^{\mathsf{A}^*} \mid \underline{b}) = \langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^* \rangle = \langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}^k \rangle \tag{4.1}$$

Now if there existed D that could distinguish $(\langle T, \mathsf{A}^*\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ from $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$, then by Equation 4.1, D distinguishes $\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}^k\rangle$ and $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$ and break hiding. ■

4.3 Impossibility results for concurrent composition

As with the parallel case, we require the constraints on $\underline{\mathcal{B}}$ and $\mathcal{I}$ stated in Remark 2.5. For simplicity of presentation here, we will sketch our proof for the case $\mathcal{I} = 2^{[k]}$, which allows us to say that an honest receiver Rec^k samples $I \leftarrow_{\mathrm{R}} 2^{[k]}$ by having the i'th session of Rec pick with probability $1/2$ whether or not the i'th bit should be opened *as soon as the i'th session of the commitment terminates*. We state a more relaxed constraint on $\mathcal{I}$ in Appendix B.2.3.

Theorem 4.8. *For $\mathcal{I} = 2^{[k]}$ and any $\underline{\mathcal{B}}$ such that $\mathcal{I}, \underline{\mathcal{B}}$ satisfy the constraints of Remark 2.5, and relative to any oracle, no $o(\log n/\log\log n)$-round commitment is CC-CBCH secure for $\mathcal{I}, \underline{\mathcal{B}}$.*

Proof sketch. As with Theorem 4.4, we give a black-box reduction that takes any $o(\log n/\log\log n)$-round commitment that is CC-CBCH secure for $\mathcal{I}, \underline{\mathcal{B}}$ and builds an efficient 1-simulator embedding. Combined with Theorem 4.2, this leads to a contradiction, and so such commitments do not exist.

Building the simulator embedding: the message schedule we use is exactly that of [CKPR03], which we call Σ, and is defined in Appendix B.2.3. The high-level idea of T is to execute Sim_{k^2} and inserting the real receiver's messages into one session j chosen at random, and where T aborts if the simulator tries to rewind queries in session j. Messages for other sessions are computed using the partial transcripts generated by the simulator so far. We defer a more explicit description to Appendix B.2.3.

Analyzing simulator embedding T: [CKPR03] prove the following lemma:

Lemma 4.9 ([CKPR03], informal)**.** *Suppose that Sim_{k^2} runs in polynomial time, and that $\left((\mathsf{Sim}_{k^2}^{\mathsf{Rec}_\Sigma^{k^2}}) \mid \underline{b}\right) \approx_c \langle \mathsf{Send}^{k^2}(\underline{b}), \mathsf{Rec}_\Sigma^{k^2}\rangle$. In particular, $\mathsf{Rec}_\Sigma^{k^2}$ does not reject any commit-phase transcripts as invalid. Then, assuming g, h are drawn from the appropriate families of hash functions, it holds with non-negligible probability that there exists a "good session" in the execution of $\mathsf{Sim}_{k^2}^{\mathsf{Rec}_\Sigma^{k^2}}$, i.e. a session where Sim_{k^2} does not rewind $\mathsf{Rec}_\Sigma^{k^2}$.*

By the assumption that $(\mathsf{Send}, \mathsf{Rec})$ is CC-CBCH secure, the hypothesis of Lemma 4.9 is satisfied. Recall that the only place where T may abort is if Sim_{k^2} tries to rewind the receiver in session j. Therefore, with probability $1/k^2$, T picks the good session that is guaranteed to exist by Lemma 4.9 with non-negligible probability. Furthermore, since the k^2 concurrent simulation is indistinguishable from a real interaction, it follows that conditioned on not aborting, the distribution of the single session $(\langle T, \mathsf{Rec}\rangle \mid b, \mathsf{NoAbort}_T)$ is computationally indistinguishable from $\mathsf{Sim}^{\mathsf{Rec}} \mid b$, and therefore T is a 1-simulator embedding.

Regarding the distribution of the bit b, one can check that the above construction implies that Rec asks for b to be opened with probability $1/2$ (*e.g.* one can think of the valid subsets being $\mathcal{I}' = \{\varnothing, \{1\}\}$), and conditioned on being opened it has distribution $\mathcal{B}'$ over a $\{0,1\}$ where $\Pr_{b \leftarrow_{\mathrm{R}} \mathcal{B}'}[b = 0] = \Pr_{I \leftarrow_{\mathrm{R}} \mathcal{I}, i \leftarrow_{\mathrm{R}} I, \underline{b} \leftarrow_{\mathrm{R}} \underline{\mathcal{B}}}[b_i = 0]$. It follows that if $\mathcal{I}, \underline{\mathcal{B}}$ satisfies the constraints of Remark 2.5 over $\{0,1\}^k$, then $\mathcal{I}', \underline{\mathcal{B}}'$ satisfies the constraints over $\{0,1\}$. ■

References

[Bar01] B. Barak. How to go beyond the black-box simulation barrier. In *Proc. 42nd FOCS*, pages 106–115. IEEE, 2001.

[BL02] B. Barak and Y. Lindell. Strict Polynomial-time in Simulation and Extraction. Cryptology ePrint Archive, Report 2002/043, 2002. Extended abstract appeared in STOC' 02.

[Bea96] D. Beaver. Adaptive zero knowledge and computational equivocation (extended abstract). In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 629–638, New York, NY, USA, 1996. ACM.

[BHY09] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 1–35. Springer, 2009.

[BCC88] G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, Oct. 1988.

[BC87] G. Brassard and C. Crépeau. Zero-knowledge simulation of Boolean circuits. In *Proceedings on Advances in cryptology—CRYPTO '86*, pages 223–233, London, UK, 1987. Springer-Verlag.

[BCY89] G. Brassard, C. Crépeau, and M. Yung. Everything in NP Can Be Argued in *Perfect* Zero-Knowledge in a *Bounded* Number of Rounds. In *Eurocrypt '89*, pages 192–195, 1989. LNCS No. 434.

[CKPR03] R. Canetti, J. Kilian, E. Petrank, and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires (Almost) Logarithmically Many Rounds. *SIAM J. Comput.*, 32(1):1–47, 2003.

[DCIO98] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Non-interactive and non-malleable commitment. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 141–150, New York, NY, USA, 1998. ACM.

[DCO99] G. Di Crescenzo and R. Ostrovsky. On Concurrent Zero-Knowledge with Preprocessing. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 485–502. Springer, 1999.

[DNRS03] C. Dwork, M. Naor, O. Reingold, and L. Stockmeyer. Magic Functions: In Memoriam: Bernard M. Dwork 1923–1998. *J. ACM*, 50(6):852–921, 2003.

[Fis01] M. Fischlin. *Trapdoor Commitment Schemes and Their Applications*. Ph.D. Thesis (Doktorarbeit), Department of Mathematics, Goethe-University, Frankfurt, Germany, 2001.

[GK96] O. Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. *J. Cryptology*, 9(3):167–189, Summer 1996.

[GK90] O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM J. of Com.*, 25(1):169–192, Feb. 1996. Preliminary version appeared in ICALP' 90.

[GMW86] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *Journal of the ACM*, 38(3):691–729, July 1991. Preliminary version in FOCS' 86.

[HHRS07] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols - A tight lower bound on the round complexity of statistically-hiding commitments. In *Proc. FOCS '07*, pages 669–679, 2007.

[HRVW09] I. Haitner, O. Reingold, S. Vadhan, and H. Wee. Inaccessible entropy. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 611–620, New York, NY, USA, 2009. ACM.

[Nao89] M. Naor. Bit Commitment Using Pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991. Preliminary version in CRYPTO' 89.

[PTW09] R. Pass, W.-L. D. Tseng, and D. Wikström. On the Composition of Public-Coin Zero-Knowledge Protocols. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2009.

[PW09] R. Pass and H. Wee. Black-Box Constructions of Two-Party Protocols from One-Way Functions. In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 403–418. Springer, 2009.

[PRS02] M. Prabhakaran, A. Rosen, and A. Sahai. Concurrent Zero Knowledge with Logarithmic Round-Complexity. In *Proc. 43rd FOCS*. IEEE, 2002.

[Ros06] A. Rosen. *Concurrent Zero-Knowledge - With Additional Background by Oded Goldreich*. Information Security and Cryptography. Springer, 2006.

[ZCZ09] Z. Zhang, Z. Cao, and H. Zhu. Constant-round adaptive zero knowledge proofs for NP, 2009. Manuscript.

A Additional definitions

A.1 Inaccessible entropy

All our definitions here are taken from [HRVW09], and we refer the reader there for motivation, intuition, and lemmas regarding how they are manipulated. Let A, B denote interactive TM's, and let A_i, B_i be the random variable describing i'th message sent by A, B respectively. We note that [HRVW09] denote "smoothed" versions of entropy that take into account A, B that can abort; for simplicity we define our notions without this subtlety.

Definition A.1. Given a $2t$-round interactive protocol (A, B), we define the sample-entropy of a transcript $\tau = \langle \mathsf{A}, \mathsf{B} \rangle = (a_1, b_1, \ldots, a_t, b_t)$ from A's point of view to be

$$\mathrm{RealH}_{\mathsf{A}}(\tau) = \sum_{i=1}^{t} -\log(\Pr[A_i = a_i \mid A_1 = a_1, B_1 = b_1, \ldots, A_{i-1} = a_{i-1}, B_{i-1} = b_{i-1}])$$

We say that the A has real min-entropy if

$$\Pr_{\tau = \langle \mathsf{A}, \mathsf{B} \rangle}[\mathsf{RealH}_{\mathsf{A}}(\tau) \geq k] \geq 1 - n^{-\omega(1)}$$

In our setting, typically A will be the receiver and B will be the sender. We write A before B as this is the convention used in [HRVW09].

Definition A.2. Let (A, B) be a $2t$-round interactive protocol. Let A^* be an interactive TM, which tosses random coins s_i in round i. A^* expects queries $(a_{[i-1]}, b_{[i-1]})$ from B, and replies with (a_i, w_i) where $a_{[i]} = \mathsf{A}(q; w_i)$ is consistent with the $a_{[i-1]}$ contained inside q. We define the accessible sample-entropy of a view $v = (s_0, b_1, a_1, w_1, s_1, \ldots, b_t, a_t, w_t, s_t)$ as:

$$\mathrm{AccH}_{\mathsf{A},\mathsf{A}^*}(v) = \sum_{i=1}^{t} -\log(\Pr_{S_i}[\mathsf{A}_i^*(v; S_i) = (a_i, \star) \mid s_0, b_1, a_1, w_1, \ldots, b_{i-1}, a_{i-1}, w_{i-1}, s_{i-1}]$$

We say that A has context-independent accessible max-entropy at most k if there is no efficient A^* and efficient predicate $\mathsf{success}$ such that:

1. For any view v, $\mathsf{success}(v)$ implies that v is consistent with A (*i.e.* for all i, $\mathsf{A}(b_{[i]}; w_i) = a_{[i]}$).
2. $\Pr_{v=\langle \mathsf{A}^*, \mathsf{B}\rangle}[\mathsf{success}(v)] \geq 1/\mathrm{poly}(n)$.
3. For all (possibly inefficient) B^*, it holds that

$$\Pr_{v=\langle \mathsf{A}^*, \mathsf{B}^*\rangle}[\neg\mathsf{success}(v) \text{ or } \mathrm{AccH}_{\mathsf{A},\mathsf{A}^*}(v) > k] > 1 - n^{-\omega(1)}$$

B Additional theorems and details

B.1 Constructions

Theorem 3.3 (Restated)**.** ([DCO99]) *Protocol 3.1 with the STR-PAR-CBCH (resp. STR-PAR-SB) version of the preamble of Protocol 3.2 gives a STR-PAR-CBCH (resp. STR-PAR-SB) commitment.*

Proof of Theorem 3.3. The binding properties are easy to verify, given the fact that Naor's commitment scheme is statistically binding.

The following simulator works to prove security against selective opening attacks for both the computational and statistical binding variants. Consider the k-fold repetition $\mathsf{Send}^k, \mathsf{Rec}^k$ of the protocol. Following the proof of Goldreich and Kahan [GK96], one can construct a simulator such that, by rewinding the first step of the preamble (*i.e.* Step 1 of Protocol 3.2), can learn the value of the $\alpha_1, \ldots \alpha_k$ used in each of the k parallel sessions. Care must be taken to ensure this finishes in expected polynomial time, but the same technique as in [GK96] works in our setting and we refer the reader to that paper for details.

Now for each $i \in [k]$ in the i'th session the simulator can sample $s_0, s_1 \leftarrow_{\mathrm{R}} \{0,1\}^n$ and reply with $\beta_i = G(s_0) \oplus G(s_1) \oplus \alpha_i$. This sets $\sigma_i = G(s_0) \oplus G(s_1)$. Then the sender sends $c = G(s_0)$. Now the simulator can decommit to both 0 (by sending s_0) and to 1 (by sending s_1). ■

B.2 Lower bounds

B.2.1 Impossibility under parallel composition

Theorem B.1. *For all $\mathcal{I}, \underline{\mathcal{B}}$ and relative to any oracle, there exists no PAR-PB commitment protocol secure for $(\mathcal{I}, \underline{\mathcal{B}})$.*

Proof. Let $(\mathsf{Send}, \mathsf{Rec})$ be the scheme and let m denote the number of random bits used by Rec. We construct a $(k, 2^{-mkt})$-simulator embedding for $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$. This suffices to prove the theorem because, analogous to Theorem 4.3, this implies that the commitment scheme cannot be hiding, for otherwise one would obtain a cheating sender that breaks binding with non-zero probability. Suppose without loss of generality that Rec sends its random coins as the very last message in the commit phase.

Building simulator embedding T: Let $p(n)$ denote the maximum number of queries made by Sim_k.

1. T_{com} guesses the random coins ω of the real receiver. T_{com} picks $U \subseteq [p(n)]$ of size t.
2. T_{com} forwards queries in U to the real receiver. The remaining queries are answered using the coins ω that the sender guessed.
3. At the end of the commit-phase Rec^k sends all its random coins at the end of the commit phase. T_{com} checks whether it guessed the random coins correctly, and if not it aborts.
4. Sim_k outputs a set I of bits to be opened. T_{com} checks that I was the real receiver's response to a query in U, and that the query consists only of simulator queries in U and the corresponding real receiver responses. If not, T_{com} aborts.
5. Upon receiving $\underline{b}$, T_{decom} outputs the opening obtained from Sim_k.

Analyzing simulator embedding T: with probability 2^{-mk}, T_{com} correctly guesses the receiver's random coins. With probability $1/p(n)^t$, all messages in the transcript that the simulator outputs correspond to queries in U, and so T_{com} does not abort. Therefore the probability that T does not abort is at least $2^{-mk}/p(n)^t \gg 2^{-mkt}$, and from the definition of T it is clear that $(\langle T, \mathsf{Rec}^k\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ is identical to $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$.

■

Theorem B.2. *For all $\mathcal{I}, \underline{\mathcal{B}}$ and relative to any oracle, there exists no public-coin PAR-CBCH commitment protocol secure for $(\mathcal{I}, \underline{\mathcal{B}})$.*

Proof. Given any public-coin commitment protocol $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$, we construct a 1-simulator embedding, which is implicit in [PTW09]. Combined with Theorem 4.2 this implies that $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$ is not PAR-CBCH secure.

Case of statistical binding: in the case of statistical binding protocols, the theorem follows immediately from Claim 4.7. This is because for public-coin protocols, it is trivial to access full max-entropy. Furthermore, there is no restriction to constant rounds because in public-coin protocols it is clear how much entropy is in each round, so we can apply the Claim 4.7 and the lemmas of [HRVW09] to arbitrary-round receiver public-coin protocols. This implies that if the commitment is statistically binding, it cannot be computationally hiding.

Building the simulator embedding T for computational binding: following [PTW09], our simulator embedding will require $k = \text{poly}(t)$ parallel sessions. T_{com} internally execute Sim_k by randomly choosing one $j \in [k]$ of the sessions to forward to the real receiver, while the rest are internally simulated. [PTW09] describe a strategy for T_{com} to *rewind the simulator* such that,

with high probability, Sim_k outputs with non-negligible probability exactly the session that was forwarded to the real receiver. Roughly, for each of the t rounds of the protocol, T_{com} forwards the next message from session k to the receiver and returns the response to the simulator. It then repeatedly runs many continuations of the simulator until it finds a continuation where the simulator's response is likely to be included in the final output (and if no such continuation exists, T_{com} aborts). We refer the reader to [PTW09] for details.

T_{com} also checks that the subset I that Sim_k asks to be opened is in response to a query that consists of simulator queries and real receiver responses, and if not T_{com} aborts. Otherwise, T_{decom} outputs an opening using the simulator.

Analyzing the simulator embedding T for computational binding: [PTW09] prove that the simulator outputs with non-negligible probability a transcript that consists only of queries and responses to the real receiver, and a natural generalization of Claim 4.5 implies that the subset I that Sim_k asks to be opened is one that consists of queries and responses to the real receiver. Hence T does not abort with non-negligible probability. Furthermore, it holds that $(\langle T, \mathsf{Rec}\rangle \mid b, \mathsf{NoAbort}_T) \approx_c (\mathsf{Sim}_k^{\mathsf{Rec}} \mid b)$. The distribution of the single bit b is the same as described at the end of the proof of Theorem 4.8. ■

B.2.2 Equivalence of PAR-SB and SH

Claim 4.7 (Restated)**.** *The receiver in* $(\mathsf{Send}^k, \mathsf{Rec}^k)$ *has real min-entropy at least* $km(1 - 1/k^{1/3})$*. There is a black-box reduction that uses the hiding property of* $(\mathsf{Send}^k, \mathsf{Rec}^k)$ *to show that* Rec^k *has context-independent accessible max-entropy* $\leq km - k/4$*.*

Proof. The first part of the claim concerning real min-entropy follows from the definitions and amplification by parallel repetition (Proposition 3.8 in [HRVW09]). For the second part, we give a black-box construction that uses any A^* (with corresponding predicate $\mathsf{success}$) that samples high context-independent max-entropy to construct a $(k, 2^{-k})$-simulator embedding T such that a distinguisher between $(\langle T, \mathsf{A}^*\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ and $(\langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^*\rangle$ can be used to break the hiding of $(\mathsf{Send}^k, \mathsf{Rec}^k)$. Using lemmas from [HRVW09], we also show that indeed such a distinguisher exists, therefore A^* cannot exist.

The simulator embedding is identical to the one given in the body of the paper. Proving that it works requires the following two lemmas:

Lemma B.3 ([HRVW09], Lemma 6.10)**.**

$$\Pr_{v=\langle \mathsf{Send}^k, A^*\rangle}[\mathsf{AccH}_{\mathsf{Rec}^k,\mathsf{A}^*}(v) > km - k/4 \textit{ and } v \textit{ is rejecting}] \leq n^{-\omega(1)}$$

By the definition of $\mathsf{success}(v)$, this lemma implies

$$\Pr_{v=\langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^*\rangle}[\mathsf{success}(v) \text{ and } v \text{ is accepting}] \geq 1/\mathrm{poly}(n) - n^{-\omega(1)} \geq 1/\mathrm{poly}(n) \tag{2.1}$$

Lemma B.4 ([HRVW09], Lemma 6.13)**.**

$$\Pr_{v=(\langle T, A^*\rangle \mid \underline{b})}[\mathsf{AccH}_{\mathsf{Rec}^k,\mathsf{A}^*}(v) > km - k/4 \textit{ and } v \textit{ is accepting}] \leq n^{-\omega(1)}$$

Similarly, this lemma implies

$$\Pr_{v=(\langle T, \mathsf{A}^*\rangle \mid \underline{b})}[\mathsf{success}(v) \text{ and } v \text{ is accepting}] \leq n^{-\omega(1)} \tag{2.2}$$

We claim that the efficiently computable predicate "$\mathsf{success}(v)$ and v is accepting" can distinguish between $\langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^* \rangle$ and $(\langle T, \mathsf{A}^* \rangle \mid \underline{b}, \mathsf{NoAbort}_T)$. Notice that by definition of Rec_h and T, $\langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^* \rangle = \langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}_h \rangle$ and $(\langle T, \mathsf{A}^* \rangle \mid \underline{b}, \mathsf{NoAbort}_T) = (\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b})$, therefore this is equivalent to breaking the hiding property for the cheating verifier Rec_h.

By definition, $\langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^* \rangle = \langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}_h \rangle$, therefore Inequality 2.1 implies that

$$\Pr_{v=\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}_h \rangle)} [\mathsf{success}(v) \text{ and } v \text{ is accepting}] \geq 1/\mathrm{poly}(n)$$

Again by definition, $(\langle T, \mathsf{A}^* \rangle \mid \underline{b}, \mathsf{NoAbort}_T) = (\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b})$. Therefore Inequality 2.2 and the observation from [HRVW09, GK90] that T does not abort with probability $1/p(n)^{O(1)}$ where $p(n)$ is the running time of Sim_k implies

$$\Pr_{v=(\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b})} [\mathsf{success}(v) \text{ and } v \text{ is accepting}] \leq n^{-\omega(1)} p(n)^{O(1)} \leq n^{-\omega(1)}$$

Therefore, the efficiently computable predicate "$\mathsf{success}(v)$ and v is accepting" breaks hiding for the cheating verifier Rec_h. ■

B.2.3 Concurrent composition

Theorem 4.8 (Restated). *For $\mathcal{I} = 2^{[k]}$ and any $\underline{\mathcal{B}}$ such that $\mathcal{I}, \underline{\mathcal{B}}$ satisfy the constraints of Remark 2.5, and relative to any oracle, no $o(\log n / \log\log n)$-round commitment is CC-CBCH secure for $\mathcal{I}, \underline{\mathcal{B}}$.*

We clarify the definitions we left out of the sketch in the body.

Building the schedule Σ: the message schedule we use is exactly that of [CKPR03], which we call Σ and is defined on k^2 sessions. The k^2 sessions are numbered $1, \ldots, k^2$ and divided into k blocks of k sessions, which are scheduled recursively. Formally, starting with $k' = k^2$, we defer the explicit schedule recursively as follows:

1. If $k' \leq k$, then execute sessions $1, \ldots, k$ sequentially.
2. If $k' > k$, then for $j = 1, \ldots, t$:
   - (Message exchange) Send two messages (one from sender to receiver and one from receiver to sender) in each of the first k sessions.
   - (Recursive schedule) If $j < t$, apply the schedule recursively to the next $\lfloor \frac{k'-k}{t-1} \rfloor$ sessions.

A *recursive block* is the the set of k sessions whose messages are exchanged together in a message exchange phase. It is convenient to identify a session $i \in [k^2]$ with $(i_b, i_s) \in [k]^2$ where i_b is its recursive block and i_s is its position within that block.

A simulator query q consists of sender queries and receiver responses, possibly from many different concurrent sessions and in an arbitrary order. Suppose that the last sender message of q (which is what the receiver should respond to) belongs to block j, and belongs to the i'th message exchange in block j. The *block prefix* of q, denoted $\mathsf{bl-prefix}(q)$, is the set of all messages in q that occur before the first message in the block j. The *iteration prefix* of q, denoted $\mathsf{it-prefix}(q)$, is the set of all messages in q that occur before and including the $i-1$'th message exchange in block j. Note that the iteration prefix is only defined if $i > 1$.

Using Σ to build simulator embedding: Let $p(n) = \text{poly}(n)$ be an upper bound on the running time of Sim_k.

1. T_{com} picks one session j at random, and picks g, h at random from families of $p(n)$-wise independent hash functions.

2. T_{com} runs Sim_k with schedule Σ, responding to the queries q as follows:

   (a) T_{com} computes the iteration-prefix of $\mathsf{it\text{-}prefix}(q)$ and checks if $g(\mathsf{it\text{-}prefix}(q)) = 0$, and if so responds to the simulator with a "receiver abort" message (note this does not mean T aborts, only that the receiver it is emulating aborts).

   (b) Otherwise, T_{com} checks to see if the query q corresponds to the j'th session that should be forwarded to the real receiver: if so it forwards it to the real receiver and responds to the simulator with the real receiver's response. If the simulator tries to rewind the receiver in the j'th session, T_{com} aborts and halts.

   (c) For queries q to sessions besides session j, T_{com} computes the block-prefix $\mathsf{bl\text{-}prefix}(q)$ and answers as Rec_h, which is defined as $\mathsf{Rec}_h(q) = \mathsf{Rec}(q; h(\mathsf{bl\text{-}prefix}(q)))$.[5]

3. When either the real receiver or Rec_h asks a session i to be opened, then T_{decom} is given b_i, and computes the opening as computed by Sim_{k^2}.

Constraints on $\mathcal{I}$: the main constraint on $\mathcal{I}$ that we impose is that it be a product of many sets $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2 \times \ldots \times \mathcal{I}_{\omega(\log k)}$. In this way, each session will be associated with many bits (say the bits in $\mathcal{I}_1$), which will be executed in parallel. This product structure is needed because we want each session to be able to make a decision about what subset of bits to be opened as soon as the commit-phase for that session is over, without affecting what subsets other sessions are allowed to ask to be opened.

[5] In fact it is also required that Sim_k be modified to never cause too many "receiver abort" messages, but we leave out the details. The reader is referred to [CKPR03, Ros06] for details.