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

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

March 9, 2010

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 (resp. 5) rounds are necessary and sufficient to build computationally (resp. statistically) binding and computationally hiding commitments. Also, there are no perfectly binding 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.

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 [14, 4]. 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.* [18]) 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 for every cheating receiver strategy that outputs a commit-phase transcript, a set $I \subset [k]$, and decommitments to $(b_i)_{i \in I}$ that is indistinguishable from the output of the cheating 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 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 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 via fully black-box reductions:*

1. *One-way permutations imply* 4*-round* PAR-CBCH *commitments exist.*
2. t*-round stand-alone* SH *commitments imply* $(t+3)$*-round* PAR-SB *commitments exist.*
3. t*-round stand-alone* SH *commitments imply* $\omega(t \log n)$*-round* CC-SB *commitments exist.*

In particular, Item 2 implies that collision-resistant hash functions (or even just 2-round statistically hiding commitments) suffice to construct 5-round PAR-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 *commitment.*
2. *There is no* 4*-round* PAR-SB *commitment.*
3. *There is a black-box reduction that uses a* $O(1)$*-round* PAR-SB *commitment to build a* $O(1)$*-round statistically hiding commitment.*

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

We stress that besides the constraint that the simulator be black-box, these results are otherwise *unconditional.* Namely, Theorem 1.2 implies that no such commitments exist in the plain model (without oracles), but also implies 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 impossibility results in the plain model.

Combining the second item of Theorem 1.2 with the main theorem of [15], which proves that there is no black-box reduction building a $o(n/\log n)$-round statistically hiding commitment from one-way permutations, 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.*

Wee [23] independently proved via different techniques a theorem similar to Corollary 1.3 for the very closely related case of trapdoor commitments.

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-CBCH *commitments.*

1.2 Comparison to previous constructions

Notions related to security against selective opening attacks have previously been studied in the literature. Security against selective opening is closely related to chameleon blobs [5, 6], trapdoor commitments [11], and equivocable commitments [2, 9, 8]. Roughly speaking, these notions all allow a simulator that can generate commit-phase transcripts that can be opened in many ways. Indeed, our constructions will be based on the equivocable commitment of [8].

Security against selective opening may be weaker than the notions above, and was directly studied in [10, 3]. Bellare *et al.* [3] 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. [3] already noted that stand-alone SH commitments satisfy a notion of PAR-SH security based on indistinguishability (this notion is different from ours). Independent of our work, Zhang *et al.* [24] 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 (under our definition of selective opening security). Their construction is an extension of a recent trapdoor commitment of Pass and Wee [19].

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 [15].

1.3 Comparison to previous lower bounds

Bellare *et al.* [3] 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- or 4-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 [3] 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 [3] 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**. [3] 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 Definition 2.5). In particular, the results of [3] also use the assumptions in Definition 2.5.

Ways in which our lower bounds are weaker: our results are weaker because they only apply to constructions with black-box simulators, *i.e.* we require that there exists a single simulator that works given black-box access to any cheating receiver. The results of [3] 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 parallel composition are essentially the equivocable commitment scheme of [8], while the case for concurrent composition follows in a straight-forward way by combining the commitment of [8] with the preamble from the concurrent zero knowledge proof of [21].

Our lower bounds are proven by observing that most known lower bounds for zero knowledge (*e.g.* [13, 17, 7, 16, 20]) 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 [13], etc.), then $L \in \mathbf{BPP}$.

As was observed by [10, 3], plugging a t-round $\mathsf{PAR\text{-}CBCH}$ 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

[1] Because it still requires that the crypto primitive be treated as an oracle, [3] do *not* rule out techniques such as Barak's simulator for constant-round public-coin zero-knowledge [1], 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.

without increasing round complexity. Furthermore, the resulting protocol has $t + 2$ rounds, and has a black-box simulator if the commitment had a black-box simulator. This immediately implies the following:

Proposition 1.6 ([13], weak impossibility of PAR-CBCH, informal)**.** *In the plain model, 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 [13], this implies that **NP** = **BPP**. But this contradicts the existence of a PAR-CBCH commitment scheme, since by the Cook-Levin reduction we can use an algorithm solving **NP** to break any commitment.

Our results improve upon Proposition 1.6 as they apply to broader categories of commitments (*e.g.* 3-round vs. non-interactive). In addition, Proposition 1.6 uses the Cook-Levin reduction and therefore does not apply when considering schemes that might use random oracles. In contrast, Theorem 1.2 does hold relative to any oracle, and in the case of Item 3 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 3 of Theorem 1.2 with the black-box lower bound of Haitner *et al.*. This is only possible if Item 3 of Theorem 1.2 is a black-box reduction, which would not be true using the approach of the weak impossibility result Proposition 1.6.

To prove Theorem 1.2, we construct what we call "blindfolded senders": senders that run the commit phase without knowing the bits that must be revealed. We show that the existence of such blindfolded senders implies that binding can be broken. We then construct blindfolded senders for various kinds of protocols by applying the proof strategy for zero knowledge lower bounds originally outlined by Goldreich and Krawczyk [13]. By arguing directly, we avoid the Cook-Levin 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. We let U_k denote the uniform distribution over $\{0,1\}^k$. For a set S, we let $x \leftarrow_{\mathrm{R}} S$ denote a uniform element of S. Let 2^S denote the set of all subsets of S. All security definitions in this paper are with respect to non-uniform circuits. We say that an event occurs with overwhelming probability if it occurs with probability $1 - n^{-\omega(1)}$, and that it occurs with negligible probability if it occurs with probability $n^{-\omega(1)}$. 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 $\mathrm{poly}(n)$ 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.

We now formally define commitments for single-bit messages; since we will be concerned with commitments that are composable, 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. Exactly t messages are exchanged in the commit phase t.
2. The *decommit phase* transcript consists of Send sending (b, open) to Rec. $\mathsf{Rec}(\tau, b, \mathsf{open}) = 1$ if open is a valid opening, and outputs 0 otherwise.

Notation and variable definitions: We assume that a commitment scheme is put in a canonical form, where each party alternates speaking. We assume the number of rounds is even and the receiver speaks first. If the number of rounds is $2t$, then we label the sender's messages $\alpha_1, \dots, \alpha_t$ and the receiver's messages $\beta_1, \dots, \beta_t$, and we let $\alpha_{[i]} = (\alpha_1, \dots, \alpha_i)$ 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 $\beta_{[i]} = \mathsf{Rec}(\alpha_{[i-1]}; \omega)$ where $\alpha_{[i-1]}$ are the first $i-1$ sender messages, and ω are the receiver's random coins. We let $\mathsf{Rec}(\perp; \omega) = \beta_1$ denote the first receiver message.

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 $\mathsf{open}, \mathsf{open}'$ such $\mathsf{Rec}(\tau, 0, \mathsf{open}) = 1$ and $\mathsf{Rec}(\tau, 1, \mathsf{open}') = 1$. 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.

2.1.2 Hiding under selective opening attacks

We only study the case of computational hiding (see Remark 1.5). In the following, $\mathcal{I} \subseteq 2^{[k]}$ 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 (Hiding under selective opening: k-fold parallel composition security game)**.** Sender input: $\underline{b} \in \{0,1\}^k$. Let Rec' be the (possibly cheating) sender.

1. $\mathsf{Send}^k, \mathsf{Rec}'$ run k executions of the commit phase in parallel using independent random coins, obtaining k commit-phase transcripts $\tau^k = (\tau_1, \ldots, \tau_k)$.
2. Rec' chooses a set $I \leftarrow_{\mathrm{R}} \mathcal{I}$ and sends it to Send^k.
3. Send^k sends (b_i, ω_i) for all $i \in I$, where ω_i is an opening of the i'th commitment.

In Item 2, the honest receiver is defined to pick $I \in \mathcal{I}$ uniformly, while a malicious receiver may pick I adversarially.

Definition 2.4 (Hiding under selective opening, parallel composition)**.** Let $\mathcal{I} \subseteq 2^{[k]}$ be a family of subsets and $\underline{\mathcal{B}}$ be a family of message distributions over $\{0,1\}^k$ for all k. Let $(\mathsf{Send}, \mathsf{Rec})$ be a commitment and Sim_k be a simulator. We say that $(\mathsf{Send}, \mathsf{Rec})$ is secure against selective opening attacks for $(\mathcal{I}, \underline{\mathcal{B}})$ if for all k:

- 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' (without knowledge of $\underline{b}$) 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 $\underline{b} \leftarrow_{\mathrm{R}} \underline{\mathcal{B}}$.

Definition 2.5. We say that $(\mathcal{I}, \underline{\mathcal{B}})$ is *non-trivial* if (the uniform distribution over) $\mathcal{I}, \underline{\mathcal{B}}$ are efficiently samplable, $|\mathcal{I}| = k^{\omega(1)}$ and also $\Pr_{I \leftarrow_{\mathrm{R}} \mathcal{I}}[H_\infty(\underline{\mathcal{B}}_I) \geq 1/\mathrm{poly}(n)] \geq 1/\mathrm{poly}(n)$.

Here $\underline{\mathcal{B}}_I$ is the joint distribution of bits $\underline{\mathcal{B}}_i$ for $i \in I$. 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 noticeable probability over the choice of I, there is non-negligible entropy in the bits revealed. This is very natural as otherwise any receiver is trivially simulable since it always sees the same constant bits. This non-triviality condition suffices for all our lower bounds except Item 3 and Item 4 of Theorem 1.2; see their respective sections for further discussion.

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 simultaneously for all $\mathcal{I}, \underline{\mathcal{B}}$. For such a notion, we prepend STR to the name of the security property (*e.g.* $\mathsf{STR\text{-}PAR\text{-}SB}$).

Definition 2.6 (Security game 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 [3]. We do not work with this weakening because it makes the definition not truly concurrent: forcing all the openings to occur simultaneously "synchronizes" the sessions.

2.2 Inaccessible entropy

All our definitions here are taken from [16], 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 [16] denote "smoothed" versions of entropy that take into account A, B that can abort; for simplicity we define our notions without this subtlety.

Definition 2.8. 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 [16].

Definition 2.9. 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:

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

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

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

3 Constructions

Di Crescenzo and Ostrovsky [8] (see also [9]) showed how to build an *equivocable* commitment scheme. Equivocable means that for every cheating receiver Rec', there exists a simulator that generates a commit-phase transcript that is computationally indistinguishable from a real transcript, but which the simulator can decommit to both 0 and 1. Equivocation seems even stronger than STR-PAR-CBCH security, except that STR-PAR-CBCH explicitly requires security to hold against selective opening attacks. Although it is not clear how to generically convert any stand-alone equivocable commitment to an equivocable commitment that is composable in parallel/concurrently, the particular construction of Di Crescenzo and Ostrovsky can be composed by using a suitable preamble.

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 [18] 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 ([8, 9, 18])**.** 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 $(\sigma \wedge b)_i = \sigma_i \wedge b$).
> **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 ([8])**.** 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. ([8]) *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 sketch of Theorem 3.3. We include a proof sketch for the sake of completeness, and refer the reader to [18, 12, 8] for full proofs.

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 [12], 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 [12] 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). ■

Protocol 3.4 ([21])**.** Preamble for STR-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.

Theorem 3.5 ([21, 22])**.** *Protocol 3.1 using the preamble of Protocol 3.4 gives a STR-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.* [21]. They prove the following:

Theorem 3.6 (Theorem 5.2 of [21], informal)**.** *There is black-box simulator strategy that, given access to any efficient receiver for Protocol 3.4 with any concurrent scheduling, outputs with high probability in every session a string α* before *Step* 3 *such that the receiver opens to α in step* 5*.*

Namely, [21] 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, one can apply the simulator strategy of [12, 8], as in the proof sketch of Theorem 3.3. ■

4 Optimality of constructions

We now define our main tool for proving lower bounds, *blindfolded senders*. Intuitively, a blindfolded sender must run its commit phase without knowing what it is committing to, so if it can cause the receiver to accept with non-negligible probability, then it must be able to open its commitments in many ways.

4.1 Blindfolded senders

For a pair of algorithms $T = (T_{com}, T_{decom})$, define the following game:

1. $(\tau^k, I, \mathsf{state}_{com}) = \langle T_{com}, \mathsf{Rec}^k \rangle$. Here, state_{com} is the internal state of T_{com} to be transmitted to T_{decom}. I is the set Rec^k asks to be opened. Notice T_{com} runs without knowledge of $\underline{b}$, hence T is "blindfolded" during the commit phase.

2. $T_{decom}(\underline{b}, \tau^k, I, \mathsf{state}_{com}) = \{(b_i, \mathsf{open}_i)\}_{i \in I}$.

The overall transcript (conditioned on not aborting) is $(\langle T, \mathsf{Rec}^k \rangle \mid \underline{b}, \mathsf{NoAbort}_T) = (\tau^k, I, \{(b_i, \mathsf{open}_i)\}_{i \in I})$, where $\mathsf{NoAbort}_T$ denotes the event that T does not abort. Say that $(\tau^k, I, \mathsf{state}_{com})$ is δ*-openable* if with probability at least δ over the choice of $\underline{b}$, Rec^k accepts $(\tau^k, I, \{(b_i, \mathsf{open}_i)\}_{i \in I})$, where $\{(b_i, \mathsf{open}_i)\}_{i \in I} = T_{decom}(\underline{b}, \tau^k, I, \mathsf{state}_{com})$..

Definition 4.1 (Blindfolded sender)**.** We say that $T = (T_{com}, T_{decom})$ form a (k, ε, δ)-blindfolded sender for $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$ if it holds that

$$\Pr[(\tau^k, I, \mathsf{state}_{com}) = \langle T_{com}, \mathsf{Rec}^k \rangle \text{ is } \delta\text{-openable } \wedge \ \mathsf{NoAbort}_T] \geq \varepsilon$$

We say T is a k-blindfolded sender if it is a $(k, 1/\mathrm{poly}(n), 1 - n^{-\omega(1)})$-blindfolded sender.

Using blindfolded senders to break binding. Here we show that secure commitments cannot admit blindfolded senders. In the next few sections, we will show that certain kinds of commitments (*e.g.* 3-round) must admit blindfolded senders, which, combined with the following theorem, imply that those kinds of commitments cannot be secure. All of these theorems are proven via black-box reductions.

Theorem 4.2. *Fix any non-trivial* $(\mathcal{I}, \underline{\mathcal{B}})$ *and* k*-fold repeated commitment scheme* $(\mathsf{Send}^k, \mathsf{Rec}^k)$ *with a simulator* Sim_k *that proves computational hiding. If this commitment has a* k*-blindfolded sender* $T = (T_{com}, T_{decom})$ *for any* k*, then this commitment cannot be statistically binding. If furthermore* T *is efficient, then this commitment cannot be computationally binding.*

Proof. The idea is to convert a k-blindfolded T sender into a sender Send' that breaks binding in a single execution of the commitment, Send' emulates T internally and chooses one of the k parallel instances to insert its interaction with the real receiver Rec. By the non-triviality of $(\mathcal{I}, \underline{\mathcal{B}})$, with high probability over $I \leftarrow_{\mathrm{R}} \mathcal{I}$ the coordinates in I have significant min-entropy, and in particular some coordinate must have significant min-entropy. Therefore if Send' picks this coordinate, then since T is able to open its commitment with non-trivial probability for $I \leftarrow_{\mathrm{R}} \mathcal{I}$ and $\underline{b} \leftarrow_{\mathrm{R}} \underline{\mathcal{B}}$, it follows that Send' can open its commitment to both 0 and 1 with non-negligible probability.

We now proceed formally by constructing a malicious sender Send' and proving that this sender breaks binding.

Algorithm 4.3.
Malicious sender Send', interacting with a single honest receiver Rec:

1. Pick a random j. For each $j' \neq j$, sample random coins $\omega^{(j')}$ to run an honest receiver.

2. Respond to the i'th message β_i from Rec as follows.

   (a) If $i > 1$, let $(\alpha^{(1)}_{[i-1]}, \ldots, \alpha^{(k)}_{[i-1]})$ be T_{com}'s response from previous queries.

(b) For $j' \neq j$, compute $\beta_i^{(j')} = \mathsf{Rec}(\alpha_{[i-1]}^{(j')}; \omega^{(j')})$. Set $\beta_i^{(j)} = \beta_i$.

(c) Feed $(\beta_i^{(1)}, \ldots, \beta_i^{(k)})$ to T_{com} to obtain response $(\alpha_{[i]}^{(1)}, \ldots, \alpha_{[i]}^{(k)})$ (assuming T_{com} does not abort).

(d) Forward $\alpha_i^{(j)}$ back to Rec.

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

4. If $j \notin I$, Send' 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 Send': By hypothesis, T is a $(k, \varepsilon, 1 - n^{-\omega(1)})$-blindfolded server for some $\varepsilon = 1/\mathrm{poly}(n)$. This implies that with probability at least ε, $\langle T_{com}, \mathsf{Rec}^k \rangle$ produces an $(1 - n^{-\omega(1)})$-openable $(\tau^k, I, \mathsf{state}_{com})$. Therefore, since the probability of producing an accepting opening for a random $\underline{b}$ at least $(1 - n^{-\omega(1)})$, it holds with probability at least $\varepsilon(1 - n^{-\omega(1)})^2$ that Rec^k accepts both openings $T_{decom}(\underline{b}, \tau^k, I, \mathsf{state}_{com})$ and $T_{decom}(\underline{b}', \tau^k, I, \mathsf{state}_{com})$.

Since $(\mathcal{I}, \underline{\mathcal{B}})$ is non-trivial, a straightforward calculation implies that $\Pr_{\underline{b}, \underline{b}', I}[\forall i \in I,\ b_i = b'_i] \leq n^{-\omega(1)}$. Therefore with probability $\varepsilon(1 - n^{-\omega(1)})^2 - n^{-\omega(1)}$, T produces accepting openings for $\underline{b}$ and $\underline{b}'$ and furthermore there exists i such that $\underline{b}_i \neq \underline{b}'_i$. Since the sender picked at random the coordinate j that contains the real interaction, with probability $1/k$ it chooses $j = i$ and therefore with non-negligible probability produces decommitments for both 0 and 1 in an interaction with the real receiver, breaking binding. ■

4.1.1 Strong non-triviality

Some of our results require the following stronger notion of non-triviality.

Definition 4.4. $(\mathcal{I}, \underline{\mathcal{B}})$ is strong non-trivial if:

1. $\mathcal{I}$ is a product of $\sqrt{k}$ large sets: formally, there exists some partition $\Pi = (\Pi_1, \ldots, \Pi_{\sqrt{k}})$ of $[k]$ into $\sqrt{k}$ subsets, and $\mathcal{I} = \mathcal{I}_1 \times \ldots \mathcal{I}_{\sqrt{k}}$ and for each i, it holds that $\mathcal{I}_i \subseteq 2^{\Pi_i}$ and $|\mathcal{I}_i| = n^{\omega(1)}$.

2. For each $i \in [\sqrt{k}]$, let I_i be the projection of I onto the coordinates in Π_i. It holds that

$$\Pr_{I \leftarrow_{\mathrm{R}} \mathcal{I}}[\forall i,\ H_\infty(\underline{\mathcal{B}}_{I_i}) \geq \omega(\log n)] \geq 1/\mathrm{poly}(n)$$

This definition strengthens the non-triviality condition on $(\mathcal{I}, \underline{\mathcal{B}})$ in two ways: first we require that $\mathcal{I}$ be a product of $\sqrt{k}$ sets, each of which is large. (Here, $\sqrt{k}$ is arbitrary, any n^ε would be equivalent for our purposes.) Second, we require the amount of entropy in $\underline{\mathcal{B}}_{I_i}$ to be large ($\omega(\log n)$ rather than just $1/\mathrm{poly}(n)$) simultaneously for all i. Notice that it is still satisfied by natural $(\mathcal{I}, \underline{\mathcal{B}})$, for instance $\mathcal{I} = 2^{[k]}$ the set of all subsets of $[k]$, and $\underline{\mathcal{B}} = U_k$ the uniform distribution over $\{0,1\}^k$.

Theorem 4.5. *Fix any strong non-trivial* $(\mathcal{I}, \underline{\mathcal{B}})$ *and* k*-fold repeated commitment scheme* $(\mathsf{Send}^k, \mathsf{Rec}^k)$ *with a simulator* Sim_k *that proves computational hiding. If this commitment has a* $(k, 1/\mathrm{poly}(n), 1/\mathrm{poly}(n))$*-blindfolded sender* $T = (T_{com}, T_{decom})$ *for any* $k = \omega(\log n)$*, then this commitment cannot be statistically binding. If furthermore* T *is efficient, then this commitment cannot be computationally binding.*

Proof sketch. The proof is identical to Theorem 4.5, the only additional observation is that because T only guarantees with noticeable probability that the commit-phase $(\tau^k, I, \mathsf{state}_{com})$ is $1/\mathrm{poly}(n)$-openable (rather than $(1 - n^{-\omega(1)})$-openable), we need the stronger non-trivial guarantee to say that even sampling only from the $1/\mathrm{poly}(n)$ fraction of the message distribution $\underline{\mathcal{B}}$ that causes Rec^k to accept, still we will find $\underline{b}, \underline{b}'$ that differ on the subset I of bits to be opened. ∎

4.2 Impossibility results for parallel composition

We construct blindfolded senders using the strategy of Goldreich and Krawczyk [13]. Intuitively, the idea is to construct a sender T whose output distribution is the same as $\mathsf{Sim}_k^{\mathsf{Rec}_h}$. Here, Rec_h is intuitively a cheating receiver that, for each sender message, uses its hash function h to generate a response that looks completely random, and therefore Sim_k gains no advantage by rewinding Rec_h. From this cheating property, we will be able to conclude that T satisfies Definition 4.1

Goldreich and Krawczyk [13] observe that we can make the following simplifying assumptions *w.l.o.g.*: **(1)** Sim_k makes exactly $p(n) = \mathrm{poly}(n)$ queries to its receiver black box, **(2)** all queries made by Sim_k are distinct, and **(3)** Sim_k always outputs a transcript τ^k that consists of queries it made to the receiver and the corresponding receiver responses.

The following lemma from [13] says that simply by guessing uniformly at random, one can pick with some probability the queries/responses that the simulator outputs as its final transcript.

Lemma 4.6 ([13])**.** *Fix a black-box simulator Sim_k for a protocol with t sender messages, and suppose Sim_k makes $p(n)$ queries. Draw $u_1, \ldots, u_t \leftarrow_R [p(n)]$, then with probability $\geq 1/p(n)^t$, the final transcript output by Sim_k consists of the $u_1, \ldots, u_t$'th queries (along with the corresponding receiver responses).*

4.2.1 3-round commitments

Theorem 4.7. *For all non-trivial $(\mathcal{I}, \underline{\mathcal{B}})$ and relative to any oracle, there exists no 3-round PAR-CBCH commitment protocol secure for $(\mathcal{I}, \underline{\mathcal{B}})$.*

Proof. We construct a polynomial-time k-blindfolded sender for $(\mathsf{Send}, \mathsf{Rec})$ for $k = n$. By Theorem 4.2, this contradicts the binding property of the commitment.

Algorithm 4.8.
Blindfolded sender $T = (T_{com}, T_{decom})$ for 3-round commitments:

1. T_{com} picks $u_1, u_2 \leftarrow_R [p(n)]$.
2. T_{com} internally runs Sim_k, answering its queries as follows:
   - For the u_1, u_2'th queries, if the u_1'th query is a first sender message α_1 and the u_2'th query is a second sender message $\alpha_{[2]}$ that extends α_1, then T_{com} forwards them to the real receiver and forwards the receiver's responses to the simulator. Otherwise, T_{com} aborts.
   - For all other queries: if the query is α_1, then T_{com} returns $\mathsf{Rec}^k(\alpha_1; \omega)$ for uniform ω. If the query is $\alpha_{[2]}$ then T returns a random $I \leftarrow_R \mathcal{I}$.
3. When Sim_k requests that a subset I of bits be revealed, T_{com} checks to see if I was the set that the real receiver asked to be opened. If not, T_{com} aborts.

4. In the opening phase, T_{decom} receives $\underline{b}$ and feeds $(b_i)_{i\in I}$ to the simulator and obtains $(\tau^k, I, (b_i, \mathsf{open}_i)_{i\in I})$. T_{decom} checks that τ^k and I consists of queries to/from the real receiver, and if not aborts. Otherwise it outputs these openings.

Analyzing blindfolded sender T. It is clear that T runs in polynomial time.

Lemma 4.6 implies that with probability $1/p(n)^2$, Sim_k picks the set to be revealed I using the guessed queries u_1, u_2.

Claim 4.9. *The probability that Sim_k makes two queries $\alpha_{[2]}, \alpha'_{[2]}$ that are both answered with the same I is negligible*

This claim holds because $|\mathcal{I}| = n^{\omega(1)}$ and Sim_k makes at most $p(n) = \mathrm{poly}(n)$ queries. Claim 4.9 implies that when T emulates Sim_k, Sim_k cannot pick I using the real receiver's messages but then find a different commit-phase transcript that leads to the same set I. Therefore the probability that T does not abort and outputs the queries to and responses from the real receiver is at least $1/p(n)^2 - n^{-\omega(1)} \geq 1/\mathrm{poly}(n)$.

Claim 4.10. *Rec^k accepts $(\langle T, \mathsf{Rec}^k\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ with overwhelming probability.*

This claim combined with the above assertion that T does not abort with non-negligible probability implies that T satisfies Definition 4.1.

We now prove Claim 4.10 by comparing the output of T to $(\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b})$ where Rec_h is defined as follows: h is a $p(n)$-wise indepenent hash function, it responds to first sender queries α_1 by computing $\beta_1 = \mathsf{Rec}(\alpha_1; h(\alpha_1))$ and to second sender queries $\alpha_{[2]}$ by sampling uniform $I \leftarrow_{\mathrm{R}} \mathcal{I}$ using $h(\alpha_{[2]})$ as random coins.[2]

As observed by [13], $(\langle T, \mathsf{Rec}\rangle \mid \underline{b}, \mathsf{NoAbort}_T) = (\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b})$ for a uniform choice of h. Since Rec_h is efficient, by the hiding property this is indistinguishable from $\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}_h\rangle$, which in turn by definition is equal to $\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}^k\rangle$. Since Rec^k always accepts a real interaction, therefore Rec^k accepts $(\langle T, \mathsf{Rec}\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ with overwhelming probability. ■

4.2.2 4-round commitments

Theorem 4.11. *For all $\mathcal{I}, \underline{\mathcal{B}}$ and relative to any oracle, there exists no 4-round PAR-SB commitment protocol secure for $(\mathcal{I}, \underline{\mathcal{B}})$.*

Proof. As before, it suffices to construct a k-blindfolded sender for $k = n$.

Algorithm 4.12.
Blindfolded sender $T = (T_{com}, T_{decom})$ for 4-round PAR-SB commitments:

1. T_{com} picks $u_1, u_2 \leftarrow_{\mathrm{R}} [p(n)]$.

2. T_{com} receives the first message β_1 from the receiver.

3. T_{com} internally runs Sim_k, answering its queries as follows:

[2] The message β_1 and the set I are independent, so there is no consistency constraint to ensure between β_1 and I. This is why we can handle 3 rounds and not just non-interactive commitments as a naive application of [13] might suggest.

- For the simulator's u_1, u_2'th queries, if the u_1'th query is a first sender message α_1 and the u_2'th query is a second sender message $\alpha_{[2]}$ that extends α_1, then T_{com} forwards them to the real receiver and forwards the receiver's responses to the simulator. Otherwise, T_{com} aborts.
- For all other queries: if the query is α_1 then T_{com} samples a random $\omega' \leftarrow_{\mathrm{R}} \{\omega \mid \mathsf{Rec}(\perp;\omega) = \beta_1\}$ and returns $\beta_2 = \mathsf{Rec}(\beta_1, \alpha_1; \omega')$ to the simulator. If the query is $\alpha_{[2]}$ then the simulator picks a random $I \leftarrow_{\mathrm{R}} \mathcal{I}$ and returns it to the simulator.

4. When Sim_k requests that a subset I of bits be revealed, T_{com} checks to see if I was the set that the real receiver asked to be opened. If not, T_{com} aborts.
5. In the opening phase, T_{decom} receives $\underline{b}$ and feeds $(b_i)_{i \in I}$ to the simulator and obtains $(\tau^k, I, (b_i, \mathsf{open}_i)_{i\in I})$. T_{decom} checks that τ^k and I consists of queries to/from the real receiver, and if not aborts. Otherwise it outputs the openings.

Analyzing blindfolded sender T. T may not run in polynomial time because sampling $\omega' \leftarrow_{\mathrm{R}} \{\omega \mid \beta_1 = \mathsf{Rec}(\perp;\omega)\}$ may be inefficient. This implies the sender breaking binding given by Theorem 4.2 may be inefficient, which is why we can only handle PAR-SB commitments.

Applying Lemma 4.6, T does not abort with probability $\geq 1/p(n)^2$. Claim 4.9 applies here for the same reason as in the proof of Theorem 4.7, therefore it holds with probability $1/p(n)^2 - n^{-\omega(1)} \geq 1/\mathrm{poly}(n)$ that T's messages to/from the receiver are exactly those in the output of its emulation of Sim_k.

We claim that Claim 4.10 holds in this case as well, which would imply that T satisfies Definition 4.1. We prove Claim 4.10 in this setting by comparing the output of T to $(\mathsf{Sim}_k^{\mathsf{Rec}_h^{\omega_1,\ldots,\omega_s}} \mid \underline{b})$, where we use the cheating receiver strategy $\mathsf{Rec}_h^{\omega_1,\ldots,\omega_s}$ defined by Katz [17]: s will be set below, and the ω_i are random coins for the honest receiver algorithm such that $\mathsf{Rec}(\perp;\omega_i) = \mathsf{Rec}(\perp;\omega_j)$ for all $i, j \in [s]$, and h is a $p(n)$-wise independent hash function with output range $[s]$. The first message of $\mathsf{Rec}_h^{\omega_1,\ldots,\omega_s}$ is $\beta_1 = \mathsf{Rec}(\perp;\omega_1)$ and given sender message α_1, the second message is $\beta_2 = \mathsf{Rec}(\beta_1, \alpha_1; \omega_{h(\beta_1,\alpha_1)})$. Given sender messages $\alpha_{[2]}$, the set I to be opened is sampled using $\omega_{h(\beta_{[2]},\alpha_{[2]})}$ as random coins.

As observed in [17], for $s = 50p(n)^2/\delta$ it holds that the statistical distance between $(\langle T, \mathsf{Rec}^k\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ and $(\mathsf{Sim}_k^{\mathsf{Rec}_h^{\omega_1,\ldots,\omega_s}} \mid \underline{b})$ is at most δ, where the randomness is over uniform $p(n)$-wise independent h, uniform ω_1 and uniform $\omega_2, \ldots, \omega_s$ conditioned on $\mathsf{Rec}(\perp;\omega_j) = \mathsf{Rec}(\perp;\omega_1)$ for all $j \in [s]$. By the commitment's hiding property this is indistinguishable from $\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}_h^{\omega_1,\ldots,\omega_s}\rangle$, which in turn is equal to $\langle \mathsf{Send}^k(\underline{b}), \mathsf{Rec}^k\rangle$ by the definition of $\mathsf{Rec}_h^{\omega_1,\ldots,\omega_s}$. Finally, since Rec^k always accepts a real interaction, therefore it accepts $(\langle T, \mathsf{Rec}^k\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ with probability $1 - \delta - n^{-\omega(1)}$.

We can apply the above argument for any $\delta \geq 1/\mathrm{poly}(n)$ to conclude that Rec^k accepts $(\langle T, \mathsf{Rec}^k\rangle \mid \underline{b}, \mathsf{NoAbort}_T)$ with probability $1 - \delta - n^{-\omega(1)}$ for all $\delta \geq 1/\mathrm{poly}(n)$. Therefore Rec^k must accept with probability $1 - n^{-\omega(1)}$ and so T satisfies Definition 4.1. ■

4.2.3 Perfectly binding commitments

Theorem 4.13. *For all non-trivial $(\mathcal{I}, \underline{\mathcal{B}})$ 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 $(k, 2^{-mkt}, 1)$-blindfolded sender for $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$. This suffices to prove the theorem: although Theorem 4.2 is for the case of statistically binding, looking at its proof the reduction employed in fact shows that one can use a $(k, 2^{-mk6}, 1)$-blindfolded receiver to build a sender strategy that breaks binding with non-zero probability, contradicting perfect binding. Suppose without loss of generality that Rec sends its random coins as the very last message in the commit phase.

Building blindfolded sender T: Let $p(n)$ denote the maximum number of queries made by Sim_k. Let t be the number of rounds in the commitment.

1. T_{com} guesse random coins ω of the real receiver, and also picks a random subset $U \subseteq [p(n)]$ of size t, let $u_1 < u_2 < \ldots < u_t$ be its elements.
2. T_{com} internally executes Sim_k, answering its queries as follows:
   - For the u_j'th query, T_{com} forwards the query to the real receiver and forwards the response back to Sim_k.
   - For other queries, Sim_k computes responses using the coins ω that the sender guessed.
3. At the end of the commit-phase Rec^k sends all its random coins. 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. In the opening phase, T_{decom} receives $\underline{b}$ and feeds $(b_i)_{i \in I}$ to the simulator and obtains $(\tau^k, I, (b_i, \mathsf{open}_i)_{i \in I})$. T_{decom} checks that τ^k and I consists of queries to/from the real receiver, and if not aborts. Otherwise it outputs the openings.

Analyzing blindfolded sender T: with probability 2^{-mk}, T_{com} correctly guesses the receiver's random coins. By Lemma 4.6, 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 $(\mathsf{Sim}_k^{\mathsf{Rec}^k} \mid \underline{b})$, so T satisfies Definition 4.1.

■

4.2.4 Public-coin commitments

Theorem 4.14. *For all strong non-trivial $(\mathcal{I}, \underline{\mathcal{B}})$ 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 $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$ for a strong non-trivial $\mathcal{I}$, we construct a $(\omega(\log n), 1/\mathrm{poly}(n), 1/\mathrm{poly}(n))$-blindfolded sender, which is implicit in [20]. Combined with Theorem 4.5 this implies that $(\mathsf{Send}, \mathsf{Rec}, \mathsf{Sim}_k)$ is not PAR-CBCH secure.

Building the blindfolded sender T: following [20], our blindfolded sender will require $k = \text{poly}(t)$ parallel sessions. Look at the partition of $[k]$ into subsets $\Pi = (\Pi_1, \dots, \Pi_{\sqrt{k}})$. Because $\mathcal{I}_i \subseteq 2^{\Pi_i}$ and $|\mathcal{I}_i| = n^{\omega(1)}$, therefore it holds that $|\Pi_i| = \omega(\log n)$.

We consider the coordinates in a single subset of the partition to belong to one session. T_{com} internally execute Sim_k by randomly choosing one $j \in [\sqrt{k}]$ of the sessions to forward to the real receiver, while the rest are internally simulated. [20] 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 real receiver'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 [20] 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 blindfolded sender T for computational binding: [20] prove that the blindfolded sender causes the receiver to accept with non-negligible probability, say $\geq \varepsilon$. Then by a standard averaging argument, with probability $\geq \varepsilon/2$, the $\langle T_{com}, \mathsf{Rec}\rangle$ produces an $(\varepsilon/2)$-openable commit-phase transcript. Therefore T is a $(\omega(\log n), 1/\text{poly}(n), 1/\text{poly}(n))$-blindfolded server. ∎

4.3 PAR-SB commitments imply (stand-alone) SH commitments

To prove Item 2 of Theorem 1.2, we show that PAR-SB commitments can be used to generate a gap between real and accessible entropy [16]. Then we apply the transformation of [16] that converts an entropy gap into a statistically hiding commitment.

Theorem 4.15. *For strong non-trivial $(\mathcal{I}, \underline{\mathcal{B}})$, if there exists $O(1)$-round $(\mathsf{Send}, \mathsf{Rec})$ that is PAR-SB secure for $(\mathcal{I}, \underline{\mathcal{B}})$, then there exists $O(1)$-round statistically hiding commitments.*

Proof of Theorem 4.15. 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.

Lemma 4.16. Rec^k *has real min-entropy at least $km(1 - 1/k^{1/3})$ and has context-independent accessible max-entropy $\leq km - k/4$.*

Let Π be the partition such that $\mathcal{I} = \mathcal{I}_1 \times \ldots \times \mathcal{I}_{\sqrt{k}}$ and $\mathcal{I}_i \subseteq 2^{\Pi_i}$. For sufficiently large k, Lemma 4.16 implies there is an entropy gap for the coordinates in Π_i, and by the entropy gap amplification lemma (Lemma 3.8) of [16] implies that the entropy gap sums over all of the coordinates. Therefore for large enough k the gap is sufficient to apply the black-box construction of statistically hiding commitments from entropy gaps given by Lemmas 6.7, 4.7, and 4.18 of [16]. ∎

Proof of Lemma 4.16. The real min-entropy part of the claim follows from the definitions and amplification by parallel repetition (Proposition 3.8 in [16]). For the accessible entropy part, we use the following:

Lemma 4.17. *If there exists efficient* A^* *(and efficient predicate* $\mathsf{success}$*, see Definition 2.9) sampling high context-independent max-entropy for* Rec^k*, then there exists a* $(k, 1/\mathrm{poly}(n), 1/\mathrm{poly}(n))$*-blindfolded sender.*

By Theorem 4.5 this contradicts the binding property of the commitment and so A^* cannot exist. ∎

Proof of Lemma 4.17. This lemma holds intuitively because we can use A^* to perform the same role as Rec_h and $\mathsf{Rec}_h^{\omega_1,\ldots,\omega_s}$ in the analysis of the blindfolded senders in Theorem 4.7 and Theorem 4.11. The fact that A^* can access high accessible entropy essentially means that it can sample the i'th message conditioned on a partial transcript of first $i-1$ messages. Applying Theorem 4.2 implies that such a blindfolded sender T would break binding property of the commitment, and therefore such A^* cannot exist.

We now proceed formally.

Algorithm 4.18.
Blindfolded sender $T = (T_{com}, T_{decom})$ for $\mathsf{PAR\text{-}SB}$ commitments.

1. T_{com} picks a random subset $U \subseteq [p(n)]$ of size t, let $u_1 < u_2 < \ldots < u_t$ be its elements. T_{com} stores a table (initially empty) that associates 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).

   - For $j = u_l$'th, T_{com} checks the query $\alpha_{[i]}$ satisfies $i = l$ and $\alpha_{[l-1]}$ was the u_{l-1}'th query. If not, T_{com} aborts. Otherwise, it forwards the query $\alpha_{[i]}$ to the real receiver and gets as 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 may be inefficient, and therefore T_{com} may be inefficient.)
   - For $j \notin U$, T_{com} samples uniform s_i, computes $\mathsf{A}^*(\alpha_{[i]}; s_{[i]})$, letting (β_i, ω_i) denote its last output.

   Then, T_{com} returns β_i to Sim_k and adds an entry into its table associating $s_{[i]}$ with $\alpha_{[i]}$.

3. When Sim_k requests that a subset I of bits be revealed, T_{com} checks to see if I was the set that the real receiver asked to be opened. If not, T_{com} aborts.

4. In the opening phase, T_{decom} receives $\underline{b}$ and feeds $(b_i)_{i\in I}$ to the simulator and obtains $(\tau^k, I, (b_i, \mathsf{open}_i)_{i\in I})$. T_{decom} checks that τ^k and I consists of queries to/from the real receiver, and if not aborts. Otherwise it outputs these openings.

Analyzing T: we require the following lemmas:

Lemma 4.19 ([16], Lemma 6.10)**.**

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

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

$$\Pr_{v=\langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^*\rangle}[\mathsf{success}(v) \text{ and } v \text{ is accepting}] \ge 1/\mathrm{poly}(n) - n^{-\omega(1)} \ge 1/\mathrm{poly}(n) \qquad (4.1)$$

Also, as observed in [16], T is essentially answering queries $j \notin U$ according to the following cheating receiver strategy Rec_h, where h is a uniformly chosen $p(n)$-wise independent hash function:

Algorithm 4.20.
Cheating receiver Rec_h:

1. Generate a first receiver message β_1 by computing $s_0 = h(0)$ and $\mathsf{A}^*(\perp; s_0) = (\beta_1, \omega_1)$.
2. On sender message $\alpha_{[i]}$, generate a response β_i by computing $s_i = h(\alpha_{[i]})$ and $\mathsf{A}^*(\alpha_{[i]}; s_0, \dots, s_i) = (\beta_i, \omega_i)$.

It is clear from the definitions that

$$(\langle T, \mathsf{A}^* \rangle \mid \underline{b}, \mathsf{NoAbort}_T) = (\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b}) \tag{4.2}$$

From Equation 4.1 and the the commitment's hiding property which says that $(\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b}) \approx_c \langle \mathsf{Send}^k(\underline{b}), \mathsf{A}^* \rangle$, we deduce

$$\Pr_{v = (\mathsf{Sim}_k^{\mathsf{Rec}_h} \mid \underline{b})}[\mathsf{success}(v) \text{ and } v \text{ is accepting}] \geq 1/\mathrm{poly}(n)$$

By Equation 4.2 it follows that

$$\Pr_{v = (\langle T, \mathsf{A}^* \rangle \mid \underline{b}, \mathsf{NoAbort}_T)}[\mathsf{success}(v) \text{ and } v \text{ is accepting}] \geq 1/\mathrm{poly}(n) \stackrel{def}{=} \delta$$

But $\mathsf{success}(v)$ and v is accepting means precisely that Rec^k accepts v as a valid transcript. Also, Lemma 4.6 implies that $\Pr[\mathsf{NoAbort}_T] \geq 1/p(n)^t$. Therefore, T is a $(k, 1/p(n)^t, \delta)$-blindfolded sender.

■

4.4 Impossibility results for concurrent composition

Our theorem for the concurrent setting also holds for strong non-trivial $(\mathcal{I}, \underline{\mathcal{B}})$. For the product set $\mathcal{I} = \mathcal{I}_1 \times \ldots \times \mathcal{I}_{\sqrt{k}}$ with partition $\Pi = (\Pi_1, \dots, \Pi_{\sqrt{k}})$, we run $\sqrt{k}$ concurrent sessions, one for each Π_i. In the i'th session we run commitments for the coordinates in Π_i *in parallel*, but the commitments for different Π_i, Π_j can be interleaved concurrently. Let us define an honest receiver Rec^k to sample $I \leftarrow_{\mathrm{R}} \mathcal{I}$ by having the i'th session of Rec choose the subset of $I_i \in \mathcal{I}_i$ to be opened *as soon as the i'th session of the commitment terminates.*

Theorem 4.21. *For strong non-trivial $(\mathcal{I}, \underline{\mathcal{B}})$, 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. Let $\ell = k^{1/4}$. Notice that for non-trivial $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_{\sqrt{k}}$, $\ell^2 = \sqrt{k}$ is the number of sessions.

Building the schedule Σ: the message schedule we use is exactly that of [7], which we call Σ and is defined on ℓ^2 sessions. The ℓ^2 sessions are numbered $1, \dots, \ell^2$ and divided into ℓ block of ℓ sessions, which are scheduled recursively. Each session consists of executing commitments in parallel for the bits corresponding to the coordinates given by a set in the partition Π associated with $\mathcal{I}$. Formally, starting with $\ell' = \ell^2$, we defer the explicit schedule recursively as follows:

1. If $\ell' \leq \ell$, then execute sessions $1, \dots, \ell$ sequentially.

2. If $\ell' > \ell$, then for $j = 1, \dots, t$:

   - (Message exchange) Send two messages (one from sender to receiver and one from receiver to sender) in each of the first ℓ sessions.
   - (Recursive schedule) If $j < t$, apply the schedule recursively to the next $\lfloor \frac{\ell'-\ell}{t-1} \rfloor$ sessions.

A *recursive block* is the the set of ℓ sessions whose messages are exchanged together in a message exchange phase. It is convenient to identify a session $i \in [\ell^2]$ with $(i_b, i_s) \in [\ell]^2$ 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 $\mathsf{bl\text{-}prefix}(q)$, is the set of all messages in q that occur before the first message in the block j. The *iteration prefix* of q, denoted $\mathsf{it\text{-}prefix}(q)$, 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 blindfolded sender: Let $p(n) = \mathrm{poly}(n)$ 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 $\mathsf{Rec}_h(q) = \mathsf{Rec}(q; h(\mathsf{bl\text{-}prefix}(q)))$.[3]

3. When either the real receiver or Rec_h asks a session i to be opened, then T_{decom} is given b_{I_i}, and computes the opening as computed by Sim_k.

Analyzing blindfolded sender T: [7] prove the following lemma:

Lemma 4.22 ([7], informal)**.** *It holds with non-negligible probability that there exists a "good session" in the execution of* $\mathsf{Sim}_k^{\mathsf{Rec}_\Sigma^k}$, i.e. *a session where* Sim_k *does not rewind* Rec_Σ^k.

[3] 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 [7, 22] for details.

The only place where T may abort is if in its emulation of Sim_k, the simulator tries to rewind the receiver in session j. Therefore, with probability $1/k$, T inserts the real receiver into the good session that is guaranteed to exist by Lemma 4.22 with non-negligible probability. Furthermore, since the k concurrent simulation is indistinguishable from a real interaction, it follows that $\mathsf{Rec}_{\Sigma}^{k}$ accepts $(\langle T, \mathsf{Rec}_{\Sigma}^{k}\rangle \mid b, \mathsf{NoAbort}_T)$ with overwhelming probability.

Because $\mathcal{I}_i \subseteq 2^{\Pi_i}$ and $|\mathcal{I}_i| = n^{\omega(1)}$, therefore it holds for all i that $|\Pi_i| = \omega(\log n)$. Therefore T is a $(\omega(\log n), 1/\mathrm{poly}(n), 1 - n^{-\omega(1)})$-blindfolded sender, and we may apply Theorem 4.5 to conclude that this contradicts the binding property of the commitment.

■

5 Acknowledgements

The author would like to thank Dennis Hofheinz and Salil Vadhan for helpful conversations.

References

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

[2] 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. ISBN 0-89791-785-5. doi: http://doi.acm.org/10.1145/237814.238014.

[3] 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. ISBN 978-3-642-01000-2.

[4] 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. ISBN 0-387-18047-8.

[5] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, Oct. 1988.

[6] 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.

[7] 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. ISSN 0097-5397. doi: http://dx.doi.org/10.1137/S0097539701392949.

[8] G. Di Crescenzo and R. Ostrovsky. On concurrent zero-knowledge with pre-processing. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 485–502. Springer, 1999. ISBN 3-540-66347-9.

[9] 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. ISBN 0-89791-962-9. doi: http://doi.acm.org/10.1145/276698.276722.

[10] 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. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/950620.950623.

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

[12] O. Goldreich and A. Kahan. How to construct constant-round zero-knowledge proof systems for NP. 9(3):167–189, Summer 1996.

[13] O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. of Com.*, 25(1):169–192, Feb. 1996. ISSN 0097-5397 (print), 1095-7111 (electronic). URL http://epubs.siam.org/sam-bin/dbq/article/22068. Preliminary version appeared in ICALP' 90.

[14] 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.

[15] 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.

[16] 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. ISBN 978-1-60558-506-2. doi: http://doi.acm.org/10.1145/1536414.1536497.

[17] J. Katz. Which languages have 4-round zero-knowledge proofs? In *TCC*, pages 73–88, 2008.

[18] M. Naor. Bit commitment using pseudorandomness. 4(2):151–158, 1991. Preliminary version in CRYPTO' 89.

[19] 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. ISBN 978-3-642-00456-8.

[20] 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. ISBN 978-3-642-03355-1.

[21] M. Prabhakaran, A. Rosen, and A. Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *Proc. 43rd FOCS*. IEEE, 2002.

[22] A. Rosen. *Concurrent Zero-Knowledge - With Additional Background by Oded Goldreich*. Information Security and Cryptography. Springer, 2006. ISBN 978-3-540-32938-1.

[23] H. Wee. On statistically binding trapdoor commitments from one-way functions, 2008. Manuscript.

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