Verifier Non-Locality in Interactive Proofs

Claude Crépeau[1] ⋆ and Nan Yang[2] ⋆⋆

[1] McGill University, Montréal, Québec, Canada. crepeau@cs.mcgill.ca
[2] Concordia University, Montreal, Quebec, Canada. na_yan@encs.concordia.ca

Abstract. In multi-prover interactive proofs, the verifier interrogates the provers and attempts to steal their knowledge. Other than that, the verifier's role has not been studied. Augmentation of the provers with non-local resources results in classes of languages that may not be **NEXP**. We have discovered that the verifier plays a much more important role than previously thought. Simply put, the verifier has the capability of providing non-local resources for the provers intrinsically. Therefore, standard MIPs may already contain protocols equivalent to one in which the prover is augmented non-locally. Existing MIPs' proofs of soundness implicitly depend on the fact that the verifier is not a non-local resource provider. The verifier's non-locality is a new unused tool and liability for protocol design and analysis. Great care should have been taken when claiming that **ZKMIP** = **MIP** and **MIP** = **NEXP**. For the former case, we show specific issues with existing protocols and revisit the proof of this statement. For the latter case, we exhibit doubts that we do not fully resolve. To do this, we define a new model of multi-prover interactive proofs which we call "correlational confinement form" (CCF-MIP).

1 Introduction

An *interactive proof* is a dialog between two parties: a polynomial-time *verifier* and an all-powerful *prover* [1, 2]. They agree ahead of time on some language L and a string x. The prover tries to convince the verifier that $x \in L$. If this is true, the prover should succeed almost all the time; if not, the prover should fail almost all the time. This is a generalization of the complexity class **NP**, except instead of simply being handed a polynomial-sized witness, you are allowed to quiz the hand that is feeding you. The set of languages that admit an interactive proof is called **IP**.

An interactive proof is *zero-knowledge* if the verifier learns nothing except the truth of "$x \in L$". This is usually defined by saying that a *distinguisher* is unable to tell apart a real conversation between the prover and the verifier and one which is generated by a lone polynomial-time *simulator*. The set of zero-knowledge interactive proofs [1] is called **ZKIP**.

One of the most important results regarding interactive proofs is that **IP** = **ZKIP** = **PSPACE**, which follows from seminal works of [3] and [4]. However, the only known way to achieve the **ZKIP** = **PSPACE** equality is to use some kind of commitment scheme which, in the single-prover model, is dependent on complexity assumptions.

The *multi-prover* model was introduced in [5]. This model consists of multiple, non-communicating provers talking solely to a single verifier. The inspiration for this model was that of a detective interrogating a bunch of suspects, each of whom is isolated in a separate room. The suspects may share a strategy before being separated, but once the interrogation begins they are no longer able to talk to one another. The main motivation for studying this model was to remove the complexity assumptions used in the commitment schemes. We will abbreviate "multi-prover interactive proof" as MIP, and the set of languages which can be accepted by MIPs as the boldface **MIP**.

⋆ Supported in part by FRQNT (INTRIQ) and NSERC (CryptoWorks21 and Discovery grant program).
⋆⋆ Supported in part by Professors Václav Chvátal, Jeremy Clark, Claude Crépeau, and David Ford.

An important consequence of having multiple provers is that the verifier can use one prover to check the consistency of other provers' answers. This gives the (weak) verifier more power over the (all-powerful) provers. Consequently, through the works of [5–7], it was shown that **MIP** = **ZKMIP** = **NEXP**. That is, any language in **NEXP** can be accepted by a MIP (optionally by a zero-knowledge MIP).

Our paper takes us all the way back to the foundations of **MIP**. We have identified a blind spot in what we call the "standard" MIP model (one verifier talking to a bunch of provers) that is not addressed in existing literature. As a lead-up to describing this blind spot, we invite the readers to consider the following ridiculous two-prover protocol:

Protocol 1. *(Ridiculous Protocol)*

1. Verifier sends Prover 1 a random string S.
2. Prover 1 replies with a string T.
3. Verifier sends Prover 2 the string T.
4. Prover 2 replies with a string S'.
5. Verifier accepts if $S = S'$.

Suppose that we claim the following ridiculous theorem:

Theorem 2. *(Ridiculous Theorem) The probability that the verifier accepts in the Ridiculous Protocol is exponentially small.*

Proof. By the definition of MIPs, the provers cannot communicate. If Prover 1 can output an S' that is the same as the uniformly random S that only Prover 2 knows, then they must have communicated. Contradiction. □

The reader is astute in pointing out that steps 2 and 3 of the Ridiculous Protocol clearly show that the verifier is helping the provers by relaying the very answer it is supposed to keep secret. The proof of the Ridiculous Theorem overlooked the blind spot that is the verifier's interactions. This is our point, exaggerated.

The blind spot we have discovered in the standard MIP model is what we shall call "correlational contamination" by the verifier. Put simply, a verifier talking to one prover *and then* talking to another prover risks unwittingly bridge the provers in some non-local capacity. At worst, this bridge could outright allow the provers to talk. If a MIP assumes that the provers are correlated in some way (i.e., local), then the verifier could undesirably change that correlation.

In existing MIP literature, the proofs of soundness do not account for this blind spot. It is easy to see the above verifier as clearly non-local (steps 2 and 3 *signals* for the provers). It is not so easy when the verifier is more complex. It is an even subtler point when we consider that the verifier could be helping the provers in a no-signaling, yet non-local manner. We believe that proofs within the standard model must be reconsidered in light of this discovery. We will further discuss this last point in Sections 2 and 7.

To clarify, we are not saying that any existing MIP protocol is unsound, only that their proofs of soundness all missed this blind spot, and therefore one could have drawn incorrect conclusions from those protocols. The standard MIP model cannot avoid the fundamental problem of correlational contamination. We believe that this blind spot is critical to the foundations of multi-prover interaction, and we wish to draw the community's attention to this issue and offer our solution – a multi-prover, multi-verifier model (Section 3). This is not a *replacement* of the single-verifier model, but rather a generalization of it. Studying this more general model has given us insight about the standard MIP model.

Our model is what we shall call the *correlational confinement form* of multi-prover interactive proofs. MIPs in this form have prover-verifier pairs who are talking, but no communication *between* any of the pairs. At the end of a correlational confinement form protocol, the verifiers get together and decide to accept or reject, at which point they no longer interact with the provers. This new model offers the following advantages:

1. If desired, the provers and verifiers are guaranteed to be local.
2. The non-local resources of provers and verifiers are made explicit.
3. It is easy to describe the provers' and verifiers' non-local resources in a much more fine-grained manner, rather than just "entangled" or "no-signaling."
4. It is possible to enforce "honest non-locality" on the provers by having the *verifier* provide them with non-local resources. Our model makes this explicit.
5. A new property of zero-knowledge emerges naturally as a result.

1.1 Our Contributions

In this work:

- We summarize issues with the standard (single-verifier) MIP model (Section 2).
- We describe our solution, a new model of multi-verifier MIP which we call "correlational confinement form" (CCF-MIP) and justify its definition by expanding on its advantages over the standard model. We argue that a special case of CCF-MIPs, one in which all provers and verifiers are local, is what the standard, single-verifier MIPs implicitly assumes. We make these assumptions explicit and show that there is a local-verifier, local-prover CCF-MIP which accepts oracle-3-SAT (Section 3).
- We describe a protocol which is local-verifier, local-prover and zero-knowledge which accepts oracle-3-SAT, achieving zero-knowledge without needing the provers to authenticate any messages, and prove its security (Section 4).
- We show that, in the CCF-MIP model, a new property of zero-knowledge naturally emerges that is a distinct stronger flavor of zero-knowledge (Section 5).
- We explore an advantage of our new model: the improved granularity by which we can define non-local correlations between the provers (and also between the verifiers), and begin to explore augmenting provers with fine-grained non-locality (Section 6).
- Finally, we explore another advantage of our new model: how to let the verifiers augment the provers with non-locality which malicious provers cannot exploit arbitrarily, which we call enforceably honest non-locality. We tentatively suggest that, since this new tool (enforceably honest non-locality) has yet to be studied by the research community, and may affect the set of languages accepted by standard MIPs, **NEXP** = **MIP** may not be fully settled (Section 7).

2 The Standard MIP Model

When [5] introduced multi-prover interactive proofs, it was formalized as follows:

Definition 1. *Let $P_1, \ldots, P_k$ be computationally unbounded Turing machines and let V be a probabilistic polynomial-time Turing machine. All machines have a read-only input tape, a private work tape and a random tape. The P_i's share a joint, infinitely long, read-only random tape. Each P_i has a write-only communication tape to V, and vice-versa. We call $(P_1, \ldots, P_k, V)$ a* k-prover interactive protocol.

Since then, there have been augmentations of the model by giving the *provers* various non-local resources, such as entanglement [8], or CHSH boxes [9, 10]. There also have been works studying the security of non-local-resistant protocols [11, 10], . However, the *verifier* remained constant throughout the literature.

The first work to point out a problem with the standard MIP model (our aforementioned blind spot), although it was not worded this explicitly, was [10]. In order to understand their point, we need to understand the following two-prover protocols.

Protocol 3. *(BGKW-type commitment protocol to b)*

P_1 and P_2 pre-share a random n-bit string w.

1. V sends a random n-bit strings r to P_2.
2. P_2 replies with $x \leftarrow b \times r \oplus w$.
3. P_1 announces to V a string w'.
4. V accepts iff $(w' \oplus x) \in \{0, r\}$.

This is a two-prover commitment protocol. Steps 1 and 2 commit, while steps 3 and 4 open. An intuitive proof of its binding condition is that, since the provers cannot signal, and they both need to know r in order to open the commitment in the way they want, therefore they cannot cheat. This intuition is incomplete, as was pointed out in [10], because breaking the binding condition *does not require signaling*. The following protocol, known as a CHSH-box, can be used to break binding without signaling.

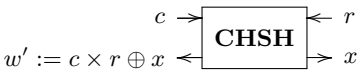

Fig. 1. a **CHSH**-box

By having P_1, P_2 obtain w', x via the CHSH-box, P_1 can open the commitment the way it wishes, c. This fact will become extremely important in Sections 4 and 5.

The punchline of [10] is that *the verifier* itself *can execute a CHSH-box for the provers without violating their no-signaling assumption.* This is not even the worst part. It is perhaps acceptable to look at protocol 3 and say that it is clearly not a CHSH-box, but consider the following:

1. Any security proof of protocol 3 must show that it does not contain a CHSH-box as a subroutine.
2. More generally, any security proof of a protocol must show that no subroutine within itself can be commandeered by the provers to achieve a non-local functionally (like the CHSH-box).
3. Composition of protocols, for instance between the committing and the opening of commitments, must be done in such a way that provably does not create a non-local box.

The solution proposed in [10] was that of *verifier isolation*. Practically speaking, this means that any message an "isolating" verifier sends to a set S of provers must be computed solely from messages that are received from S. The end result is that an isolating verifier can never accidentally implement a CHSH-box and, in general, it will always enforce the locality of the provers. In a sense, we can think of an isolating verifier as "local"; a non-isolating verifier can be thought of as "non-local". We will make this precise in the next section.

Let us look at another problem which inspired us to pose the question of verifier non-locality. Existing zero-knowledge **MIP** protocols such as [12] *require* that the verifier courier an authenticated message between the provers in order to obtain soundness while ensuring zero-knowledge. The gist of it goes like this:

1. V asks P_1 a bunch of questions.
2. V wants to check one of P_1's answers with P_2 for consistency.
3. In order for zero-knowledge to hold, V *must* ask P_2 a question it has already asked P_1.
4. P_1 authenticates a question with a key that was committed at the beginning of the protocol and sends it to V.
5. V sends the question and the authentication to P_2, who proceeds only if authentication succeeds.

Steps 4 and 5 consists of V sending a message from P_1 to P_2. It was *not* shown that this act cannot be used to signal. Moreover, as we have discussed above, it is not necessary to signal in order to break commitments, and therefore soundness. We do not know whether it is possible for the provers to actually succeed in commandeering the verifier in order to execute a non-local task; however, this needs to be proven, and the proof contained in [12] does not address this issue. Moreover, the zero-knowledge protocol of [12] *allows* P_1 to send an *arbitrary* message to P_2 (via the authentication key). Therefore one cannot even compose such a protocol in a *nested* fashion (as a subroutine call) since the inner instance would violate the assumption of the outer instance that no communication has occurred after commitment and before opening.

We quickly mention that existing simulators for a protocol such as those found in [12] needs to know how to break commitments in order to simulate. The simulator accomplishes this by acting as both provers, thereby receiving the secret strings r_0 and r_1 which are meant for one prover only. This standard model of zero-knowledge gives the simulator *unnecessary power*, in a sense. We will discuss this further in section 5. For more details on the problems of the standard **MIP** model, see [13].

2.1 Correlational Contamination as a Tool

There is a flip side to the fact that the verifier is capable of performing non-local tasks for the provers. In general, when provers become more correlated, for example through entanglement or other no-signaling distributions, it is not always clear whether the classes of languages accepted in these augmented models should shrink or expand (or change in other ways). This is because the increased correlations give both honest and dishonest provers more power, and the sum of their expanded capabilities could go either way.

With this in mind, recall that it was claimed that $\mathbf{MIP} = \mathbf{NEXP}$ [6, 7]. When provers are augmented with quantum entanglement a priori, this is a potentially different class of languages which is denoted by $\mathbf{MIP}^*$. At the time of this writing, we only know that $\mathbf{MIP}^* \supseteq \mathbf{NEXP}$ [8]. Equality has not been established. It is possible that there exists some language $L \in \mathbf{MIP}^*$ such that $L \notin \mathbf{NEXP}$. If this turns out to be the case, and that in a MIP for L, the quantum processing used by the provers can be simulated within classical polynomial-time, then we can ask a classical verifier to simulate entanglement for classical provers *in the protocol itself*, and the provers would not need to be quantum. This would falsify the result that $\mathbf{MIP} = \mathbf{NEXP}$. We have one of two possibilities:

1. The existing proof [6] that $\mathbf{MIP} \subseteq \mathbf{NEXP}$ overlooked the cases where V provides extra non-local resources to the provers.
2. All languages accepted by MIPs where P's are local with arbitrary extra non-local resources (in polynomial amount) are contained in $\mathbf{NEXP}$[3] (bringing no extra power at all).

If the first case turns out to be true, then the overall power of MIPs would increase. Otherwise, the second case being true would imply that we would only need to consider MIPs where the

[3] According to [14] this is already the case if V provides arbitrary non-locality achievable via polynomial amount of entanglement.

verifiers are also confined to local; this would simplify soundness proofs considerably. We do not know which of the two possibilities is correct, but they both seem worthy of further investigation.

We will discuss the idea of using the verifier to form enforceable and honest non-locality for the provers in section 7.

3 Correlational Confinement Form MIP

As discussed above, the standard MIP model of a single verifier talking to multiple provers could make the verifier a non-local bridge between the provers. If a protocol does not desire this – and all existing MIPs do not – it must be proven impossible.

We neutralize this problem by defining a model with multiple verifiers, each of which talks to a single prover; in turn, each prover talks to a single verifier. There are no communication tapes between the verifiers. There is a special verifier V_0 which *only reads* the outputs of the other verifiers; this is the verifier that will decide to accept or reject. We call this model "correlation-confined" since the provers and verifiers are explicitly local, and all of their non-local correlations (if any) are delegated to two machines called $\widehat{P}$ and $\widehat{V}$.

This model is a *generalization* of the standard model because the special setting where $\widehat{P}$ is empty and $\widehat{V}$ signals for the verifiers corresponds to the standard MIP model. In this section, we focus on a specific sub-case of the standard model where both $\widehat{P}$ and $\widehat{V}$ are empty, which we call "local." But first, let us define our model in full.

Definition 2. *Let $(\widehat{V}, V_0, V_1, \ldots, V_k, \widehat{P}, P_1, \ldots, P_k)$ be a tuple of interactive Turing machines, where the P's are computationally all-powerful and the V's are polynomial-time. Every machine has a read-only input tape, a local work tape, and a random tape initialized uniformly at random. For each i, there are two-way communication tapes between V_i and P_i, and that for all j, there is a two-way communication tape between $\widehat{V}$ and V_j and also between $\widehat{P}$ and P_j. In addition, for each ℓ, there is a read-only tape going from V_ℓ to V_0 (where V_0 reads). Then, this is said to be a multi-prover interactive proof in* correlational confinement form *(CCF-MIP).*

It is perhaps easier to understand our definition with the help of figure 2.

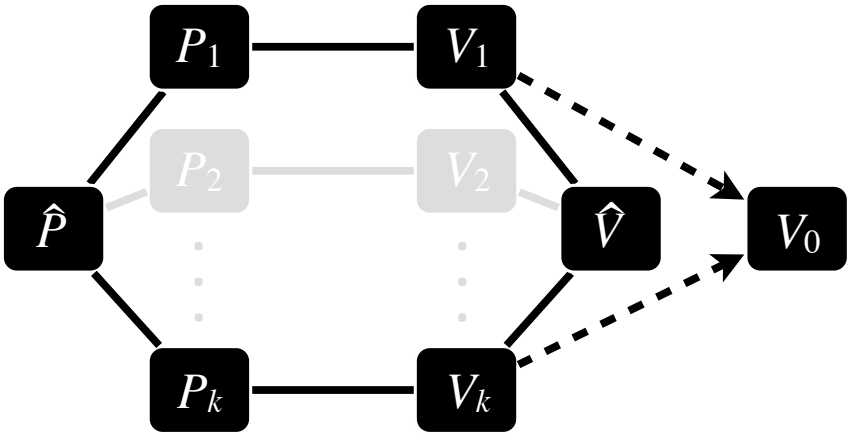

Fig. 2. MIP in Correlation Confinement Form

The solid lines are two-way communication tapes and the dotted arrows are read-only tapes, with the arrow indicating the direction of information flow. We call $\widehat{V}$ and $\widehat{P}$ "non-local tasks".

They will be used to generalize the model by explicitly giving the provers and verifiers non-local resources. For now, we are concerned only with what it means for the provers and verifiers to be local.

Definition 3. *A CCF-MIP is* local *if $\widehat{V} = \widehat{P} = \emptyset$ and all of the provers' (resp. verifiers) random tapes are initialized with an identical uniform random string.*

It is accepted that isolated parties sharing (classical) randomness is considered local, hence our need for identical random tapes in the above definition.

We can define that a CCF-MIP accepts a language L if the usual soundness and completeness conditions hold:

Definition 4. *A CCF-MIP $(\widehat{V}, V_0, V_1, \ldots, V_k, \widehat{P}, P_1, \ldots, P_k)$ accepts a language L if and only if*

- *(completeness)* $\forall x \in L, \mathbf{Pr}[V_0(x, t_1, \ldots, t_k) = \mathsf{accept}] > 2/3$,
- *(soundness)* $\forall x \notin L, \forall P'_1, \ldots, P'_k, \mathbf{Pr}[V_0(x, t_1, \ldots, t_k) = \mathsf{accept}] < 1/3$,

where t_i is the read-only tape from V_i to V_0 at the end of the interaction of V_i with P_i (or P'_i) on input x.

Note that we do not quantify over $\widehat{P}$ (nor $\widehat{V}$), as soundness would be impossible for all languages outside **PSPACE** since one possible $\widehat{P}$ is simply a signaling task for the provers. We want to use $\widehat{P}$ and $\widehat{V}$ not as (possibly malicious) participants to the protocol, but as a description of non-local resources available to the provers and verifiers.

With CCF-MIPs, the special verifier V_0 decides to accept or reject. This verifier cannot communicate with anyone else, avoiding the aforementioned correlational contamination. This is in contrast with standard, single-verifier MIPs where the verifier talks to all the provers and then decides to accept or not.

Local-prover CCF-MIPs form a subclass of standard MIPs. They are, by design, more restricted in what you can make the verifier do. An immediate question is whether this is *too* restrictive. Perhaps, in all interesting cases, it is necessary for a single verifier to go back-and-fourth between provers, using previous discussions to generate new questions.

The answer is that, surprisingly, of all the literature we have surveyed, almost all protocols can be re-written in such a way that the single verifier is "split" in our sense without any loss of functionality. We have explicitly demonstrated this in [15] for the multi-prover protocol for oracle-3-SAT in [7]. We briefly discuss how below.

The protocol details can be found in [7], section 3. For our purposes, we only need the form of the protocol, which is as follows:

Protocol 4. *(BFL Classic)*

1. V asks P_1 some questions non-adaptively.
2. V chooses a question Q from the pool of questions which was asked to P_1.
3. V asks Q to P_2.
4. V accepts if the interaction with P_1 was successful, and the answer from P_2 is consistent with those of P_1.

The crucial observation is that V does not *adaptively* ask questions to P_1. Therefore, the questions asked on that entire side of the conversation might as well have been selected in advance; and if they can be selected in advance, they can be *shared* in advance with a second verifier. We can therefore naturally rewrite the [7] protocol as a local CCF-MIP in the following way. We ask that the reader fills in the details from [7], section 3.

Protocol 5. *(BFL in correlational confinement form)*

1. V_1 prepares the questions which it will ask P_1.
2. V_1 chooses a question Q from the above list and shares it with V_2.
3. CCF-MIP begins. All parties are confined as per definitions.
4. V_1 asks the questions to P_1.
5. V_2 asks Q to P_2.
6. V_0, reading the responses, decides to accept or reject, based on the same criteria as in protocol 4.

The [7] protocol is for oracle-3-SAT, which is **NEXP**-complete. Thus we can conclusively say that **NEXP** $\subseteq$ **MIP**, at least. Why inclusion and not equality? We have partially discussed that in section 2.1, and we will discuss it further in section 7.

The non-trivial question is whether there exists a local *zero-knowledge* CCF-MIP for **NEXP**. The existing technique for achieving zero-knowledge in MIP [5, 12] requires the (single) verifier to courier an authenticated message between provers. This is not possible with local CCF-MIPs. In the next section, we show that there is a way around that constraint.

4 A Local ZKMIP in Correlational Confinement Form for NEXP

We describe here a CCF-MIP with the following properties (in addition to completeness and soundness):

- Local, as defined in the previous section.
- Zero-knowledge, in the traditional sense as well as an expanded sense discussed in the next section.

As discussed above, the oracle-3-SAT protocol from [7] is local in the sense that the interaction with P_2 involves only a single question, sampled from the pool of questions the verifier has asked P_1. This can be "localized" by a pair of verifiers by asking them to share a random tape before the protocol begins.

The existing, generic way of turning an interactive proof into a zero-knowledge one is by running it in committed form. That is, instead of answering a question directly, P_1 commits his answers, then at the end convinces V that the answers are correct. V then asks P_2 a question, and receives a committed answer. The provers would then show that the answers are equal, without opening the commitments. The problem is that during the P_2 phase, the verifier *must* ask a question that it has asked P_1 in order to ensure zero-knowledge.

This is addressed in [5] and [12] by asking the first prover to compute the verifier's question in committed form, add a message authentication tag using a secret key that the provers shared, and finally asks the verifier to send the message to P_2, who then checks whether the authentication is valid before executing its part of the protocol. This type of protocol cannot be made into local, correlational confinement form, since the verifiers of a CCF-MIP cannot talk to each other.

We present a local zero-knowledge CCF-MIP which addresses this problem. Our solution basically asks the provers to encrypt an answer with a key that is based on the verifier's question. That is, we force V_2 to behave honestly (to ask a question that V_1 has asked), without having the provers to sign off on the question, by saying, "If the verifiers ask the provers the same question, they will receive the same random string. Otherwise, they will receive two unrelated random strings." Since these questions can be chosen by the verifiers in advance, there is no need for

the provers to authenticate the questions before answering, nor for the verifier to courier said question between provers.

The rest of the protocol consists of P_1 and V_1 evaluating the verifier's circuit from the "BFL classic" protocol from section 3 of [7], which we call Λ, in committed form. We refer the reader to that reference for the detailed description of their protocol.

The gist of our local zero-knowledge CCF-MIP (protocol 7) is the prover saying to the verifier, "You filled in the randomness for the BFL verifier, and I filled in the answers. If I were to open the commitments, you would see that the BFL verifier would accept."

To begin, we will first need the CHSH commitment (protocol 6), which is secure in the local setting as previously proved in [16].

4.1 The Protocols

The following is a CHSH-type commitment that is perfectly concealing and statistically binding. In general, we use the commitment-box notation "$\boxed{b}$" as the name of a commitment to bit b in the next two protocols.

Protocol 6. *A statistically binding, perfectly concealing commitment protocol.*

All parties agree on a security parameter 1^k.
P_1 and P_2 partition their private random tape into two k-bit strings w_1, w_2.

Pre-computation phase:

- V_1 samples two k-bit strings z_1, z_2 independently and uniformly, and provides them to V_2.
- V_1 sends z_1 to P_1 and V_2 sends z_2 to P_2.

Commit phase:

- P_1 commits b to V_1 as $\boxed{b} = (b \times z_1) \oplus w_1$, where $b \times z_1$ is a multiplication in $\mathbb{F}_{2^n}$.
- P_2 sends V_2: $d = (w_1 \times z_2) \oplus w_2$.

Opening phase:

- P_1 sends w_1, w_2 to V_1.
- V_1 computes $b = 1$ if $\boxed{b} \oplus w_1 = z_1$, or $b = 0$ if $\boxed{b} = w_1$.
- V_0 **rejects** if $\boxed{b} \oplus w_1$ is anything but z_1 or 0, or if $d \oplus w_2 \neq z_1 \times z_2$ and **accepts** b otherwise.

Below is the local zero-knowledge CCF-MIP for oracle-3-SAT, Protocol 7. The basis of Protocol 7 is the (non-zero-knowledge) MIP for oracle-3-SAT found in [7], the notations of which we borrowed[4]. Following it we will give a proof sketch of its completeness, soundness, zero-knowledge and locality.

Protocol 7. *A local zero-knowledge CCF-MIP for oracle-3-SAT*

[4] For this extended abstract, we assume the reader is already familiar with the detailed protocol of [7]. If you are not, chances are, you will not be able to make sense of Protocol 7. We also use statements such as "P_1 proves to V_1 that $\boxed{\Omega_1}$ was computed correctly". The reader is expected familiarity with zero-knowledge computations on committed circuits as put forward by [17–19, 12].

Let x, an instance of oracle-3-SAT, be the common input, let $|x| = k$, and let Λ be the verifier's program in the BFL classic protocol [7].

1. **Pre-computation:**
   (a) All parties agree on a family of strongly-universal-2 hash functions $\{H_i\}$ indexed by k-bit keys.
   (b) V_1 samples two k-bit strings z_1, z_2 independently and uniformly, and provides them to V_2.
   (c) V_1 selects k random K-bit strings $r_1, ..., r_k$ and evaluates the circuit of Λ on input r_i, resulting in questions $Q_1, ..., Q_k$, and provides them to V_2
   (d) V_1 randomly chooses i, $1 \leq i \leq k+3$, the index of an oracle query that will be made to both P_1 and P_2. V_1 provides i to V_2.
   (e) V_1 sends z_1 to P_1 and V_2 sends z_2 to P_2.
   (f) P_1 and P_2 agree on a k-bit string γ.
   (g) P_1 commits $\boxed{\gamma}$ to V_1.
2. **Multilinearity test:** Let k be the number of oracle queries in this phase.
   For $1 \leq i \leq k$:
   (a) V_1 sends Q_i to P_1, the question that would be asked by Λ with coins r_i.
   (b) P_1 commits his answer as $\boxed{A(Q_i)}$.
   (c) After committing all of his answers, P_1 and V_1 evaluate a circuit description of Λ in committed form with inputs $\boxed{A(Q_1)}, \ldots, \boxed{A(Q_k)}$. P_1 opens the circuit's output. If it rejects, V_1 instructs V_0 to reject.
3. **Sumcheck with oracle:**
   – Let $g(z, Q_{k+1}, Q_{k+2}, Q_{k+3}, A(Q_{k+1}), A(Q_{k+2}), A(Q_{k+3}))$
   be the arithmetization obtained by section 3.2 of [7], let z be a string of length r and $Q_{k+1}, Q_{k+2}, Q_{k+3}$ be strings of length s, as appropriate. V_1 and P_1 execute the protocol of section 3.3 of [7] in committed form, using coins selected by P_1 whenever randomness is required. At the end of this phase, P_1 shows that the committed final value is equal to

$$g\Big(z, Q_{k+1}, Q_{k+2}, Q_{k+3}, \boxed{A(Q_{k+1})}, \boxed{A(Q_{k+2})}, \boxed{A(Q_{k+3})}\Big),$$

   an evaluation in committed form of g using the committed random bits that were used during the protocol's loop. If this fails, V_1 instructs V_0 to reject.
4. **Consistency test:**
   (a) V_1 sends i to P_1.
   (b) P_1 computes $\boxed{\Omega_1} = \boxed{A(Q_i)} \oplus H_{\boxed{\gamma}}(Q_i)$ and sends $\boxed{\Omega_1}$ to V_1.
   (c) P_1 proves to V_1 that $\boxed{\Omega_1}$ was computed correctly, from the existing commitments.
   (d) P_1 opens $\boxed{\Omega_1}$ for V_1, who gets Ω_1.
   (e) V_2 sends Q_i to P_2 (recall that this was pre-agreed in step 1.(c))
   (f) P_2 responds to V_2 with $\Omega_2 = A(Q_i) \oplus H_\gamma(Q_i)$.
   (g) V_0 accepts if and only if all of the following conditions are met:
      – $\Omega_1 = \Omega_2$
      – All commitments which have been opened are valid.
      – V_1 did not reject in the two previous cases

4.2 Proof Sketches of Security

Locality

In both the commitment protocol (protocol 6) and the CCF-MIP (protocol 7), the provers do not interact with each other, making them trivially local as per definition 3; V_0 does not interact with the remaining verifiers *once the protocol begins*; V_1 and V_2 do not interact with the provers *once the protocol ends*; therefore, since neither the provers nor the verifiers share any additional resources, the verifiers are local.

Completeness

Completeness follows from the completeness of the underlying protocol [7], and the fact that the commitment protocol (protocol 6) is well-defined for honest provers (who will never send a commitment that they cannot open).

Soundness

The scaffolding for our soundness proof will be the following. Suppose that there is a trusted third party handling commitments that are perfectly binding, and that this third party sends the receiver a unique identifying receipt for every commitment. If we were to use this commitment to execute a protocol in committed form (as we do in protocol 7), then to each committed transcript of an execution there would be a unique real transcript (of the BFL protocol in our case). In this case, a committed transcript shows that it ends in *accept* if and only if the associated real transcript, which one would obtain by running the base protocol, also ends in *accept*. Thus, if the base protocol has a soundness error of, say, 1/3, then so does this committed version.

We can remove the scaffolding by replacing the trusted third-party with protocol 6. This does not change the argument, except with exponentially small change to the soundness error. To begin with, local probabilistic provers are equivalent to local deterministic provers. This is because the success probability α of randomized provers is an average over the randomized provers' random tapes. Each instance of a random tape represents a deterministic strategy. Therefore there is a deterministic strategy which succeeds with probability at least α. Therefore, we only need to consider local deterministic provers.

Let us analyze our commitment protocol. Since P_1 is deterministic, we may unambiguously consider what happens if we were to "rewind" it. Suppose that at some point P_1 opens a particular commitment c to 0, and that by rewinding P_1 and have V_1 make different choices, P_1 then opens the *same* commitment c to 1 (an attempt to break binding). Because of locality, P_1's behavior is independent of what P_2 receives (namely z_2). Therefore, there is only *one* such z_2 which V_0 will ultimately accept as a valid opening of c in both ways (recall that our commitment is statistically binding).

In the worst case, for every commitment there exists a sequence of interaction between V_1 and P_1 such that P_1 will attempt to break the binding of that commitment. Each such commitment-breaking corresponds to at most one string z_2 that will actually work.

Let us denote the set of such binding-breaking strings by B. If $z_2 \notin B$, then the provers *will not break binding*, and the soundness error is reduced to that of the underlying protocol (at most 1/3). On the other hand, since $|B| < \mathbf{poly}(k)$, the probability that $z_2 \in B$ is at most $\mathbf{poly}(k)/2^k$.

Therefore, the soundness error of our protocol is at most

$$Pr[z_2 \notin B \text{ and underlying protocol accepts}] + Pr[z_2 \in B] \leq \frac{1}{3} + \frac{\mathbf{poly}(k)}{2^k}.$$

Zero-Knowledge

Our use of the multi-verifier setup as a guarantee of locality leads us to an unforeseen problem. The participants of our protocol are two pairs of prover-verifiers and a "decision" verifier. In attempting to prove zero-knowledge in this new picture of MIPs, we must rethink about the purpose of the (sole) simulator. Before we do that, however, let us describe how to simulate in the standard case of a single simulator.

The simulation will be divided in two parts. In the first part, the simulator produces a transcript of the *pre-computation*, *multilinearity test* and *sumcheck with oracle* parts, which involves only interactions with V_1. In the second part, the simulator will fake a valid *consistency test*.

We begin with the simple observation that *if the provers can signal, then they can break the commitment of protocol 6.* This is the basis of our simulation strategy, because if the provers can break the binding condition of commitments, then it does not matter what was actually committed. Put another way, if the simulator knows everything that both verifiers tell their respective provers, then the simulator can satisfy any cut-and-choose challenge by V_1 and convince it of anything, even though the simulator would have been committing random answers.

Since the simulator speaks to both verifiers, it does indeed know how to break binding. Therefore it can behave as if it were an honest P_1, except the simulator's answers are commitments of independent and uniformly random strings. The simulator then breaks binding whenever V_1 challenges it to open a commitment. This is the simulator's strategy for everything except the consistency check question.

For the consistency check question, which we had denoted as Q_i, the behavior of the simulator should be:

- V_1 asks Q_i and V_2 asks Q'_i.
- The simulator computes Ω_1 honestly, from the random gibberish it has committed.
- If the verifiers are honest, then $Q_i = Q'_i$ and the simulator outputs $\Omega_1 = \Omega_2$.
- Otherwise, the simulator outputs Ω_1 and an independent and uniformly random Ω_2.

But since the simulator knows which question was chosen (Q_i) and which was sent to the second prover (Q'_i), it can behave as expected.

The hiding strength of our commitment is perfect, therefore the transcripts for V_1 and V_2 are identically distributed to real transcripts. The simulator sends them to V_0 for acceptance or rejection (remember that V_0 can behave arbitrarily).

This is how zero-knowledge is handled in the standard MIP model. In the CCF-MIP model, a new, natural property of zero-knowledge emerges. Namely, if there are multiple provers and verifiers which are correlation-confined (whether they are local or not), should there not also be multiple simulators?

For protocol 7, it turns out that having a single simulator is unnecessary. It is possible to produce the same transcript with multiple, confined simulators, each interacting with a verifier. The trick is that they might need to be less confined than the verifiers – that is, they may need more non-local resources. We will discuss this in the next section.

5 Zero-Knowledge with Simulators of Minimal Correlation

For single-prover interactive proofs and standard MIPs, we typically discuss the "flavor" of zero-knowledge. Broadly speaking, there are *computational*, *statistical* and *perfect* zero-knowledge interactive proofs, corresponding to an increasing indistinguishability of the simulated transcript. For almost all known languages, the flavor of an interactive proof's zero-knowledge corresponds

to the hiding property of the commitment used. A detailed discussion about this can be found in [20].

If we directly adapt the existing definition of zero-knowledge for CCF-MIPs, we end up with something like, "For every set S of local verifiers, there exists a simulator M such that M can output a transcript indistinguishable from that of S interacting with real provers." As mentioned in protocol 7's proof of zero-knowledge, by giving the simulator the ability to converse with both verifiers, it is trivial to break the binding condition of the commitment scheme. The simulator can satisfy every cut-and-choose challenge issued by the verifiers. Therefore it is easy to simulate a real conversation since the simulator can convince the verifiers of anything.

By moving to a multi-verifier model, we were able to make the non-local resources of the provers and verifiers explicit. For protocol 7, no one involved has any non-local resources; specifically, the verifiers do not. It stands to reason that there should be as many simulators as verifiers, and that the simulators should have as little additional non-local resources as possible (ideally, it would have the same as those of the verifiers). This leads to the following definition:

Definition 5. *Let $\mathcal{M} = (\widehat{M}, M_1, \ldots, M_k)$ be a tuple of polynomial-time interactive Turing machines. Each machine has a random tape, and every random tape is initialized with the same random bits. For $1 \leq i \leq k$, there is a two-way communication tape between $\widehat{M}$ and M_i. There are no communication tapes between any of the M_i's. Then this is called a set of* correlation-confined simulators *and $\widehat{M}$ is the* confinement class *of $\mathcal{M}$.*

Definition 6. *Let $\mathcal{T} = (\widehat{V}, V_0, V_1, \ldots, V_k, \widehat{P}, P_1, \ldots, P_k)$ be a CCF-MIP. If there exists an $\widehat{M}$ such that for all verifiers $(V_0', V_1', \ldots, V_k')$, there exists $(M_1, \ldots, M_k)$, such that the transcripts of conversations between*

$$(\widehat{V}', V_0', V_1', \ldots, V_k', \widehat{P}, P_1, \ldots, P_k)$$

and

$$(\widehat{V}', V_0', V_1', \ldots, V_k', \widehat{M}, M_1, \ldots, M_k)$$

are identically distributed, where $(\widehat{M}, M_1, \ldots, M_k)$ is a set of correlation-confined simulators, then we say that $\mathcal{T}$ is a perfectly indistinguishable, $\widehat{M}$-confined zero-knowledge CCF-MIP.

To summarize, the zero-knowledge of CCF-MIPs has two attributes:

- *Strength of indistinguishability*: how indistinguishable is the simulated transcript from a real one.
- *Simulator's confinement class*: how much extra non-local resources do the simulators need to produce the transcript.

The single-simulator zero-knowledge of standard model MIPs is equivalent to a signaling set of simulators; that is, those whose $\widehat{M}$ is a signaling box. This is what protocol 7's proof of zero-knowledge actually described. However, we can do better.

Recall that the single simulator had two parts, the first part consists of passing V_1's challenges, and the second consists of passing the consistency test. We will begin by describing the first part (pre-computation, multilinearity test and sumcheck with oracle) for multiple simulators. Given two arbitrary verifiers V_1 and V_2:

Protocol 8. *(Perfectly Indistinguishable, CHSH-Confined Simulator for Protocol 7, Part 1)*

The setup:

- Let $(\widehat{M}, M_1, M_2)$ be a set of correlation-confined simulators.

– M_1 and M_2 can send $\widehat{M}$ an index along with a bit.
– $\widehat{M}$ completes the indexed CHSH box (protocol 3) for both simulators.

The simulation strategy:

1. The simulators agree on unique indices for every commitment used in the protocol.
2. M_1 interacts with V_1 the way P_1 would. Whenever P_1 should commit, M_1 commits to random bits, just like the single-simulator from section 4.
3. For each commitment, V_2 sends M_2 a string s. M_2 sends to $\widehat{M}$ the index of the commitment and s.
4. $\widehat{M}$ runs the CHSH box (protocol 3) and replies with V_2's half of the output.
5. Whenever M_1 needs to open a commitment, it can be opened in the way M_1 desires by sending the corresponding index and bit to $\widehat{M}$.
6. $\widehat{M}$ completes the corresponding CHSH box which outputs t. $\widehat{M}$ sends t to M_1.
7. M_1 sends t to V_1.

The gist of the first part of the simulation is simple. CHSH boxes can break the commitment scheme used in protocol 7 without giving the parties the ability to signal. Make $\widehat{M}$ compute CHSH boxes for the simulators, who then have the ability to break commitments without signaling. The rest of part one of the simulation is exactly as in the single-simulator case.

The second part (the consistency test) can be done by having the simulators ignore the question:

Protocol 9. *(Perfectly Indistinguishable, CHSH-Confined Simulator for Protocol 7, Part 2)*

1. V_1 sends i to M_1.
2. M_1 computes $\boxed{\Omega_1} = H_{\boxed{\gamma}}(Q_i)$.
3. Using $\widehat{M}$ to break binding, M_1 convinces V_1 that $\boxed{\Omega_1}$ is actually $\boxed{A(Q_i)} \oplus H_{\boxed{\gamma}}(Q_i)$.
4. M_1 opens $\boxed{\Omega_1}$ for V_1, who gets $\Omega_1 = H_{\boxed{\gamma}}(Q_i)$.
5. V_2 sends Q'_i to M_2.
6. M_2 responds with $\Omega_2 = H_{\boxed{\gamma}}(Q'_i)$.

By the properties of the strongly-universal-2 hash H, if $Q_i = Q'_i$ then $\Omega_1 = \Omega_2$. Otherwise $\Omega_1 \neq \Omega_2$ with exponentially high probability. This produces the result as desired. The simulators then feed the transcripts to V_0, and terminates simulation.

We close this section with two open questions. First, although protocol 7 is a *local* CCF-MIP, the only known ways of simulating the transcript are to give the simulators some kind of non-local resource such as a CHSH box (or a fully signaling box, but that is not necessary). We do not know whether it is possible to simulate protocol 7 with *local* simulators, but we are unable to show this to be impossible.

Second, the simulators' confinement class is not unique. We can modify the behavior of a CHSH box (by adding useless steps, for example) without reducing its non-localness. Therefore, it is not clear whether we can speak of a "maximal" confinement class for a given CCF-MIP. We have only shown, in our case, that the simulators can be *relatively* more confined (less non-locally correlated) than it was believed.

6 Non-Local Resources via $\widehat{P}$

The $\widehat{M}$-confined simulators of the previous section hint at how we would like to utilize our correlational confinement form MIPs in general. In existing literature, non-local resources of provers are typically given as entanglement (for quantum provers) [9], a no-signaling probability distribution from which the provers can sample [10], or arbitrary no-signaling resources [21]. There is a lack of granularity to them.

This is where $\widehat{P}$, what we call a non-local task, comes in. $\widehat{P}$, being a Turing machine, can be defined operationally. It is much more flexible than just letting provers sample from some distribution. First, a couple of definitions:

Definition 7. *Let $\mathcal{T} = (\widehat{V}, V_0, V_1, \ldots, V_k, \widehat{P}, P_1, \ldots, P_k)$ be a CCF-MIP. Then $\mathcal{T}$ has $\widehat{V}$-*confined verifiers *and $\widehat{P}$-*confined provers*. We call $\mathcal{T}$ a $(\widehat{V}, \widehat{P})$-confined MIP.*

In this section, we will focus on what we can do with $\widehat{P}$ and not $\widehat{V}$. We begin with the observation that when the provers' non-local resources change, it changes as a tool (for honest provers) and as a liability (for dishonest provers), and it is not always clear how the resulting accepted class of languages should change as a result.

For example, with local verifier/provers, we have **MIP** = **NEXP**. Increase the provers' non-local resources to arbitrary entanglement (**MIP***), and we get **MIP**$^* \supseteq$ **NEXP** [8], a possible *increase* of accepted languages. Increase the provers' non-local resources again to arbitrary no-signaling distributions (**MIP**^{ns}), however, and we end up with **MIP**^{ns} = **EXP** [21], a possible *decrease* of accepted languages (**EXP** vs. **NEXP** being open at the time of this writing).

The behavior of $\widehat{P}$ can be more complex than those correlations studied in existing literature, as $\widehat{P}$ is computationally unbounded. We believe that the consequences of this most general form of correlations between provers has not been explored. We end this section with the following question, which, sadly, we are unable to answer:

Question 1. Does there exist a $(\widehat{V}, \widehat{P})$-confined MIP which accepts a language not in **NEXP**?

More questions arise when a $\widehat{P}$ in question is computable in polynomial-time, since this means that whatever non-local resources $\widehat{P}$ is providing the provers can actually be simulated by a set of signaling verifiers (or a single verifier, in the standard MIP model). We discuss this in the next section.

7 Non-Local Resources via the Verifiers

The **MIP** = **NEXP** result is the combination of two works. First, [6] showed that **MIP** $\subseteq$ **NEXP**. Second, [7] showed that **MIP** $\supseteq$ **NEXP**. In this work, we began by describing how the second inclusion's proof of soundness overlooked a blind spot (correlational contamination). We then suggested a new, more restrictive model and presented a protocol for **NEXP** in this form, with the guarantee that the verifiers are now local. We believe that what the existing results really show is that "$(\emptyset, \emptyset)$-confined MIPs accept **NEXP**."

We make the argument below that the full generality of *local-prover* MIPs has not been explored, and that it is still possible that **MIP** $\supsetneq$ **NEXP**. We state upfront that, unfortunately, we do not have a specific example to back up our generic argument below.

So far in our discussion, verifier non-locality has been undesirable. This follows from the fact that verifier non-locality leads to prover non-locality, and the latter breaks many protocols. For example, [9] showed that certain MIPs for 3-SAT break down if the provers are entangled; [10] showed that certain multi-prover commitment schemes are no longer binding if the provers have a CHSH box; more recently [21] showed that if the provers are allowed to sample *arbitrary* no-signaling distributions, then the set of languages accepted decreases to **EXP**.

However, properly utilized, we believe that verifier non-locality is a hitherto unused tool for protocol design. Consider this generic argument: Suppose that there exists T, a $(\emptyset, \widehat{P})$-confined MIP which accepts a language $L \notin \mathbf{NEXP}$, and $\widehat{P}$ is polynomial-time computable ($T \notin \mathbf{MIP}$ since the provers are not local). Then, we can construct a new $(\widehat{V}, \emptyset)$-confined MIP T' in which we run T, except $\widehat{V}$ of T' identical to $\widehat{P}$ of T. When running T', whenever $\widehat{P}$ is needed by the provers, they will ask the verifiers to compute it for them by calling $\widehat{V}$. If this series of miracles occurs, then $\mathbf{MIP} \supsetneq \mathbf{NEXP}$, because $(\widehat{V}, \emptyset)$-confined MIPs are a subset of standard, single-verifier MIPs (and therefore $T' \in \mathbf{MIP}$). We agree that this may seem too many miracles to ask for, however they have not yet shown to be impossible.

We need to make clear that giving the provers a (polynomial-time) non-local task $\widehat{P}$ *is not the same* as having the verifiers compute $\widehat{P}$ for the provers by calling $\widehat{V}$. Giving the provers non-local resources means that malicious provers can exploit them in arbitrary ways. However, if we make the provers local, and let the verifiers provide the provers the same non-local resources, then malicious provers can no longer exploit them arbitrarily. For instance, the verifiers can control *the timing* and *the amount* of the non-local resources are used, as dictated by the protocol. If the provers deviate, then the verifiers abort. This is an example of how verifiers (and in the case of standard MIPs, the lone verifier) can enforce *honest* non-locality on the provers.

We leave as an open problem the following question:

Question 2. Does there exists a $(\widehat{V}, \emptyset)$-confined MIP which accepts a language not in **NEXP**?

Acknowledgements

We would like to thank Gilles Brassard, André Chailloux, Serge Fehr, Max Fillinger, Sophie Laplante, Justin Li, Anthony Leverrier, Arnaud Massenet, Samuel Ranellucci, Louis Salvail, and Christian Schaffner for various discussions about earlier versions of this work. We would also like to thank Jeremy Clark for his insightful comments. Finally, we are grateful to Raphael Phan and Moti Yung for inviting us to publish a preliminary version of this work as an *Insight Paper* at MyCrypt 2016. Special regards to two STOC-2018 reviewers who suggested changes to improve appropriateness of the wording in this paper.

References

1. S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," *SIAM. J. Computing*, vol. 18, pp. 186–208, Feb. 1989.
2. L. Babai, "Trading group theory for randomness," in *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pp. 421–429, May 1985.
3. A. Shamir, "IP = PSPACE," *J. ACM*, vol. 39, pp. 869–877, Oct. 1992.
4. M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway, "Everything provable is provable in zero-knowledge," in *Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '88, (London, UK, UK), pp. 37–56, Springer-Verlag, 1990.
5. M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson, "Multi-prover interactive proofs: How to remove intractability assumptions," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, (New York, NY, USA), pp. 113–131, ACM, 1988.
6. L. Fortnow, J. Rompel, and M. Sipser, "On the power of multi-prover interactive protocols," *Theor. Comput. Sci.*, vol. 134, pp. 545–557, Nov. 1994.
7. L. Babai, L. Fortnow, and C. Lund, "Non-deterministic exponential time has two-prover interactive protocols," *Comput. Complex.*, vol. 2, pp. 374–374, Dec. 1992.
8. T. Ito and T. Vidick, "A multi-prover interactive proof for nexp sound against entangled provers," in *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, (Washington, DC, USA), pp. 243–252, IEEE Computer Society, 2012.

9. R. Cleve, P. Hoyer, B. Toner, and J. Watrous, "Consequences and limits of nonlocal strategies," in *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, CCC '04, (Washington, DC, USA), pp. 236–249, IEEE Computer Society, 2004.
10. C. Crepeau, L. Salvail, J.-R. Simard, and A. Tapp, *Two Provers in Isolation*, pp. 407–430. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
11. S. Fehr and M. Fillinger, *Multi-prover Commitments Against Non-signaling Attacks*, pp. 403–421. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
12. J. Kilian, *Uses of randomness in algorithms and protocols*. MIT Press, 1990.
13. C. Crepeau and N. Yang, *Multi-prover Interactive Proofs: Unsound Foundations*, pp. 485–493. Cham: Springer International Publishing, 2017.
14. H. Kobayashi and K. Matsumoto, "Quantum multi-prover interactive proof systems with limited prior entanglement," *Journal of Computer and System Sciences*, vol. 66, no. 3, pp. 429 – 450, 2003.
15. N. Yang, "Zero-knowledge multi-prover interactive proofs," Master's thesis, Concordia University, 2013.
16. T. Lunghi, J. Kaniewski, F. Bussières, R. Houlmann, M. Tomamichel, S. Wehner, and H. Zbinden, "Practical relativistic bit commitment," *Phys. Rev. Lett.*, vol. 115, p. 030502, Jul 2015.
17. G. Brassard and C. Crépeau, "Zero-knowledge simulation of boolean circuits (extended abstract)," in *Advances in Cryptology: Proceedings of Crypto '86* (A. M. Odlyzko, ed.), vol. 263, pp. 223–233, Springer-Verlag, 1987.
18. G. Brassard and C. Crépeau, "Non-transitive transfer of confidence: A perfect zero-knowledge interactive protocol for SAT and beyond," in 27^{th} *Symp. of Found. of Computer Sci.*, pp. 188–195, IEEE, 1986.
19. R. Impagliazzo and M. Yung, "Direct minimum-knowledge computations," in *Advances in Cryptology: Proceedings of Crypto '87* (C. Pomerance, ed.), vol. 293, pp. 40–51, Springer-Verlag, 1988.
20. O. Goldreich, *Foundations of Cryptography: Volume 1*. New York, NY, USA: Cambridge University Press, 2006.
21. Y. T. Kalai, R. Raz, and R. D. Rothblum, "How to delegate computations: The power of no-signaling proofs," in *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, (New York, NY, USA), pp. 485–494, ACM, 2014.