Electronic Colloquium on Computational Complexity, Report No. 77 (2023)

Batch Proofs are Statistically Hiding

Nir Bitansky[1], Chethan Kamath[1], Omer Paneth[1], Ron D. Rothblum[2], and Prashant Nalini Vasudevan[3]

[1]Tel Aviv University,
nirbitan@tau.ac.il,ckamath@protonmail.com,omerpa@tauex.tau.ac.il
[2]Technion, rothblum@cs.technion.ac.il
[3]National University of Singapore, prashant@comp.nus.edu.sg

May 25, 2023

Abstract

Batch proofs are proof systems that convince a verifier that $x_1, \ldots, x_t \in \mathcal{L}$, for some NP language $\mathcal{L}$, with communication that is much shorter than sending the t witnesses. In the case of *statistical soundness* (where the cheating prover is unbounded but honest prover is efficient), interactive batch proofs are known for UP, the class of *unique witness* NP languages. In the case of computational soundness (aka arguments, where both honest and dishonest provers are efficient), *non-interactive* solutions are now known for all of NP, assuming standard cryptographic assumptions. We study the necessary conditions for the existence of batch proofs in these two settings. Our main results are as follows.

Statistical Soundness: the existence of a statistically-sound batch proof for $\mathcal{L}$ implies that $\mathcal{L}$ has a *statistically witness indistinguishable (*SWI*) proof*, with inverse polynomial SWI error, and a non-uniform honest prover. The implication is unconditional for public-coin protocols and relies on one-way functions in the private-coin case.

This poses a barrier for achieving batch proofs beyond UP (where witness indistinguishability is trivial). In particular, assuming that NP does not have SWI proofs, batch proofs for all of NP do not exist. This motivates further study of the complexity class SWI, which, in contrast to the related class SZK, has been largely left unexplored.

Computational Soundness: the existence of batch arguments (BARGs) for NP, together with one-way functions, implies the existence of statistical zero-knowledge (SZK) arguments for NP with roughly the same number of rounds, an inverse polynomial zero-knowledge error, and non-uniform honest prover.

Thus, constant-round interactive BARGs from one-way functions would yield constant-round SZK arguments from one-way functions. This would be surprising as SZK arguments are currently only known assuming constant-round statistically-hiding commitments (which in turn are unlikely to follow from one-way functions).

Non-interactive: the existence of non-interactive BARGs for NP and one-way functions, implies non-interactive statistical zero-knowledge arguments (NISZKA) for NP, with negligible soundness error, inverse polynomial zero-knowledge error, and non-uniform honest prover. Assuming also *lossy public-key encryption*, the statistical zero-knowledge error can be made negligible. We further show that BARGs satisfying a notion of *honest somewhere extractability* imply lossy public key encryption.

All of our results stem from a common framework showing how to transform a batch protocol for a language $\mathcal{L}$ into an SWI protocol for $\mathcal{L}$.

ISSN 1433-8092

Contents

1 Introduction

Batch proofs are interactive proof-systems that enable a prover to convince a verifier that input statements $x_1, \ldots, x_t$ all belong to a language $\mathcal{L} \in \mathsf{NP}$, with communication that is much shorter than sending the t witnesses. Batch proofs have been studied recently in two main threads: depending on whether the soundness property is required to hold against arbitrary cheating prover strategies, or only against computationally bounded ones.

The Statistical Setting. In the *statistical setting*, we require that even a computationally unbounded prover cannot convince the verifier to accept a false statement (other than with some bounded probability). On the other hand, we require that there is an *efficient honest prover* strategy (given the witnesses as an auxiliary input) for convincing the verifier of true statements. Such proofs systems are known as *doubly efficient interactive proofs* (see [Gol18] for a recent survey).

A recent sequence of works by Reingold *et al.* [RRR21, RRR18, RR20] construct doubly-efficient batch proofs for any language in the class UP (consisting of NP languages in which YES instances have a *unique* accepting witness). In particular, Rothblum and Rothblum [RR20] give such a protocol with communication $\mathsf{poly}(m, \log(t))$, where m is the length of a single witness and poly is a polynomial that depends only on the UP language. Doubly-efficient batch proofs beyond UP remain unknown, leading to a natural question [RRR21]:

> *Does every language $\mathcal{L} \in \mathsf{NP}$ have a statistically sound doubly-efficient batch proof? Do there exist other subclasses of NP (beyond UP) that have such proofs?*

If we waive the restriction that the honest prover is efficient, there is a simple answer to this question. Specifically, there is a space $\mathsf{poly}(n, m) + \log(t)$ algorithm for deciding whether $x_1, \ldots, x_t \in \mathcal{L}$, where n is the instance length and m is the witness length. Thus, via the $\mathsf{IP} = \mathsf{PSPACE}$ theorem [LFKN92, Sha92], there is an interactive proof for this problem with communication $\mathsf{poly}(n, m, \log(t))$. However, this protocol is entirely impractical as the honest prover runs in time $2^{\Omega(n)}$.

The Computational Setting. A natural relaxation of the statistical soundness condition is to only require *computational soundness*, which means that soundness is guaranteed only against *efficient* cheating provers. Such proof systems are commonly called *argument systems*. The seminal work of Kilian [Kil92] gives general-purpose succinct arguments for all of NP, assuming the existence of collision-resistant hash functions (CRH). In more detail, Kilian's protocol is a four-message argument-system with communication $\mathsf{poly}(\lambda, \log(n))$, where λ is the security parameter, for any language $\mathcal{L} \in \mathsf{NP}$. In particular, for any $\mathcal{L} \in \mathsf{NP}$, we can apply Killian's result to the related NP language

$$\mathcal{L}^{\otimes t} = \left\{(x_1, \ldots, x_t) \in (\{0,1\}^n)^t : x_1, \ldots, x_t \in \mathcal{L}\right\}$$

and obtain a *batch argument* (BARG) for $\mathcal{L}$ with communication $\mathsf{poly}(\lambda, \log(n), \log(t))$.

Kilian's protocol relies on collision-resistant hash functions (or certain relaxations thereof [BKP18, KNY18]). However, it is unclear whether such hash functions are also necessary. This gives rise to the following basic question:

> *What are the minimal cryptographic assumptions needed for succinct arguments for NP? In particular, can BARGs be constructed based solely on the existence of one-way functions?*

We remark that it is not clear that the existence of one-way functions is even *necessary* for general purpose succinct arguments for NP. The only result that we are aware of is by Wee [Wee05], who showed that 2-message succinct arguments imply the existence of a hard on average search problem in NP.

The Non-Interactive (Computational) Setting. As noted above, Kilian's protocol requires 4 messages. Reducing the number of messages in succinct arguments is a major open question in the field. Restricting to the case of BARGs though, we have a much better understanding due to recent breakthrough works. In particular, a sequence of works [BHK17, CJJ21, CJJ22, WW22, HJKS22, PP22, KLVW22] construct BARGs consisting of a single message, given a common reference string (equivalently, 2-message publicly verifiable arguments in the plain model), assuming specific cryptographic assumptions such as LWE or assumptions related to discrete log.

Still, so far all constructions of non-interactive BARGs rely on specific cryptographic assumptions. This raises the question of whether one can make do with a general assumption as in Kilian's protocol. More ambitiously:

Can non-interactive BARG*s be constructed from collision-resistant hash functions?*

1.1 Our Results

In this work we study, and give partial answers, to all of the above questions. Our key idea is a new transformation that compiles a batch protocol[1] Π, for verifying that $x_1, \ldots, x_t \in \mathcal{L}$, into a protocol Π', for a single instance, which has a hiding property. Here and below, when we say that $\mathcal{L}$ has a batch protocol, we mean that the communication for proving that $x_1, \ldots, x_t \in \mathcal{L}$ is $t^{1-\epsilon} \cdot \mathsf{poly}(m)$, for some $\epsilon > 0$, where m is the length of a single witness and poly is some polynomial which may depend on $\mathcal{L}$ but does not depend on t.

More specifically, we transform a batch protocol Π into a protocol Π' for a single instance satisfying a form of *statistical witness indistinguishability* (SWI). Recall that a protocol for an NP relation $\mathcal{R}$ is ϵ-SWI, if for every input x and witnesses $w_1, w_2 \in \{w : \mathcal{R}(x, w) = 1\}$, the view of the verifier when the prover uses w_1 is ϵ-close, in statistical distance, to its view in an interaction in which the prover uses w_2. We say that the protocol is *honest verifier* SWI if the SWI property only holds in an honest execution of the protocol (but the default notion applies to *malicious verifiers*).

Our main step transforms a batch protocol Π into an *honest-verifier* SWI protocol Π', where the SWI-error ϵ can be any inverse polynomial. The transformation also preserves the soundness of the original protocol. In other words, if Π is computationally (resp., statistically) sound then the resulting protocol Π' is computationally (resp., statistically) sound. If Π has r-rounds then Π' has $r+1$ rounds. However, the efficient honest prover strategy of Π' is non-uniform, where the non-uniform advice depends on the specification of the protocol Π.

We use this basic step in the different settings described above to reduce batch protocols into protocols satisfying hiding properties, as described next.

The Statistical Setting. Our first application of the above framework is in the statistical setting. In this setting we obtain SWI against malicious verifiers, in which the SWI error is inverse polynomial. In case we start off with a public-coin BARG the result is unconditional. Otherwise we need to assume a one-way function.[2]

Theorem 1 (Informally Stated, see Theorem 3.1 and Corollaries 3.13, 3.14)**.** *Suppose that $\mathcal{L} \in$ NP has a statistically sound r-round public-coin batch proof. Then, for any polynomial p, the language $\mathcal{L}$ has an $O(r)$-round* SWI *proof with $\frac{1}{p}$-*SWI *error and a non-uniform honest prover.*

Furthermore, for general (i.e., private-coin) statistically sound batch proofs we achieve the weaker conclusion of honest-verifier SWI*, or, assuming the existence of a one-way function, malicious verifier* SWI*.*

It is worth pointing out that Theorem 1 is also applicable to languages in UP (for which batch proofs are known), but there the conclusion is meaningless since UP has a trivial SWI proof - just send the witness!

[1] We use the terminology of "protocol" where we want to be intentionally vague as to whether soundness is computational or statistical.

[2] Note that the Goldwasser and Sipser [GS89] transformation from private-coin to public-coin protocols is inapplicable, since it results in an inefficient honest prover.

In contrast though, for general NP languages, the existence of an SWI proof seems extremely surprising. In particular, it is known that NP does not have proofs satisfying the stronger property of *statistical zero-knowledge* (SZK) (assuming the polynomial hierarchy does not collapse [For89, AH91]).[3] As it seems that the notion of SWI is closely related to that of SZK (modulo the trivial cases arising from unique witnesses) it seems reasonable to expect that NP does not have such proofs. Thus, we derive the following immediate corollary:

Corollary 2 (Informally Stated)**.** *Assume that there exists some $\mathcal{L} \in$ NP that does* not *have an* SWI *proof as in Theorem 1. Then* NP *does not have statistically sound batch proofs.*

We emphasize that we do not take for granted the fact that NP does not have SWI proofs, and we find this to be an intriguing open question. Indeed, while we have a very deep understanding of the structure of SZK (see [Vad99]), the structure of the class of languages having SWI proofs has, to the best of our knowledge, not been explored. Theorem 1 provides concrete motivation for a similar study of the class SWI, which we leave to future work.

The Computational Setting. We also apply our basic framework in the computational setting. Here though, assuming that one-way functions exist, we are able to derive the stronger hiding property of *statistical zero-knowledge*.

Theorem 3 (Informally Stated, See Theorem 3.1 and Corollary 3.14)**.** *Assume the existence of a one-way function. Suppose that every $\mathcal{L} \in$ NP has an r-round* BARG*. Then, for every polynomial p, every $\mathcal{L} \in$ NP has an $O(r)$-round statistical zero-knowledge argument-system (*SZKA*) with $\frac{1}{p}$-zero-knowledge error and a non-uniform honest prover.*

Recall that constant-round SZKA for NP are only known to exist assuming constant-round statistically-hiding commitments, and the latter seem stronger than one-way functions (and there is a blackbox separation [HHRS15]). Thus, Theorem 3 shows that the existence of constant-round BARGs for NP suffices to "lift" one-way functions to a primitive which is only known based on collision-resistant hash functions (or multi-collision resistant hash functions [BKP18, BDRV18]).

We remark that a related positive result was obtained recently by Amit and Rothblum [AR23], who constructed constant-round succinct arguments for deterministic languages (specifically for the class NC) from one-way functions. Thus, a negative interpretation of Theorem 3 is that extending the [AR23] result from succinct arguments for deterministic languages to BARGs for NP seems unlikely as it would have unexpected implications. Alternatively, a positive perspective is that Theorem 3 presents a concrete direction for constructing constant-round SZKA for NP from one-way functions.

The Non-Interactive (Computational) Setting. We apply the basic framework for the third time in the context of *non-interactive* BARG*s*. Here we face a difficulty, in that our basic framework increases the round complexity of the protocol by one round. We are able to overcome this challenge by relying on BARGs satisfying a weak form of adaptive soundness called *somewhere soundness*, a relaxation of *somewhere extractability* [CJJ22], which is achieved by recent BARG constructions. We obtain the following result:

Theorem 4 (Informally Stated, See Theorem 3.1 and Corollary 3.16)**.** *Assume the existence of one-way functions and that* NP *has somewhere-sound non-interactive* BARG*s. Then, for any polynomial p,* NP *has non-interactive statistical zero-knowledge arguments (*NISZKA*), with a negligible soundness error, $\frac{1}{p}$-zero-knowledge error, and a non-uniform honest prover.*

Like non-interactive BARGs, NISZKA are currently only known to exist based on specific cryptographic assumptions (or in the random oracle model). Theorem 4 shows that a construction from a "relatively weak"

[3] Recall that *statistical zero-knowledge* (SZK) requires that for every efficient verifier strategy there is an efficient simulator that generates a view that is statistically close to that in the actual interaction (for instances in the language). SWI can be thought of as a relaxation of SZK in which the simulator can be unbounded.

assumption, such as collision-resistant hash functions, would yield a similar result for NISZKA - which would constitute major progress in the field of zero-knowledge.

Theorem 4 yields an inverse polynomial statistical zero-knowledge error. We prove that assuming *lossy public key encryption*, which exist from a variety of assumptions (c.f. [PW08]), we can reduce this error to negligible. The resulting NISZK is non-adaptively sound.

Theorem 5 (Informally Stated, See Theorem 4.4)**.** *Assume the existence of lossy public-key encryption. Any* NISZKA *for* NP *with an inverse polynomial zero-knowledge error and negligible adaptive soundness error can be turned into one with a negligible zero knowledge error and negligible non-adaptive soundness error.*

The proof of Theorem 5 is similar in spirit to the amplification of non-interactive *computational* zero-knowledge by Goyal, Jain, and Sahai [GJS19]. Their transformation requires subexponential public-key encryption, whereas we require (polynomial) lossy public-key encryption to maintain *statistical* zero-knowledge.

We also observe that lossy public-key encryption follows from BARGs satisfying a variant of somewhere extractable BARGs, which guarantees that it is possible to extract the specific witness that was used in some predefined index in an honest proof. This is in contrast to the standard notion of somewhere extractability guaranteeing that *some* witness can be extracted (even from maliciously generated accepting proofs). As a result of independent interest, we also show that the standard notion of somewhere extractable BARGs imply *private information retrieval* and thus also *statistically sender-private oblivious transfer* and lossy public-key encryption. However the lossy public-key encryption obtained has (negligible) decryption errors (which is not sufficient for our amplification theorem). See further details in Appendix B.

Remark 1 (Hiding for Batch Protocols)**.** *All of the results listed above start with a batch protocol for a language $\mathcal{L}$ and derive a protocol with hiding properties (i.e., either* SWI *or* SZK*) for a single instance of $\mathcal{L}$. We note that all of the results can be used to obtain similar hiding properties also for a* batch *protocol for $\mathcal{L}$ via the following simple observation: rather than applying the basic result to $\mathcal{L}$, we can apply it to $\mathcal{L}^{\otimes t'}$ for any $t' \ll t$.*

Remark 2 (On the Possibility of Weak Batching)**.** *All of our results assume a batch protocol for t instances, with communication $t^{1-\epsilon}$.*[4] *Thus, our results are inapplicable to very weakly compressing batch protocols that have slightly non-trivial communication such as say, $t \cdot \sqrt{m} + \mathsf{poly}(m)$, where m is the witness length. Such weak batch protocols can nevertheless be quite powerful (see [RRR21]) and we leave the study of this setting as an interesting open problem.*

1.2 Additional Related Works

The study of communication in statistically sound interactive proofs, focusing on the prover to verifier communication, was initiated in [GH98, GVW02]. In particular, Goldreich *et al.* [GVW02] transform interactive proofs with a *single bit* of communication to be SZK. We emphasize that the results in [GH98, GVW02] are inapplicable in the setting of batch proofs. For example, the main result in [GH98] says that proofs with short communication can be emulated in time that is exponential in the communication, but this merely indicates that the communication in batch proofs for NP needs to be $\Omega(m + \log t)$, where m is the witness length.

Kaslasi *et al.* [KRR+20, KRV21] consider batch verification of protocols that are *a priori* statistical zero-knowledge, while retaining the zero-knowledge property. The constructions of [KRR+20, KRV21] are *not* doubly-efficient and so our results are inapplicable in their context.

Batch verification is also related to the problem of *AND instance compression* [HN10, FS08]. In AND instance compression, the goal is, given formulas $\phi_1, \ldots, \phi_k$, to generate in polynomial time a new formula ϕ that is satisfiable if and only if $\phi_1, \ldots, \phi_k$ are all satisfiable, and so that the length of ϕ is less than k. Batch verification considers the dual problem of compressing the *witnesses*. We note that strong infeasibility

[4] We remark that Kalai *et al.* [KLVW22] show how to amplify weak non-interactive BARGs into BARGs with very good compression but they assume the existence of rate-1 OT, whereas we are seeking transformations that rely only on the existence of the weak BARG.

results for AND instance compression were shown by Drucker [Dru15]. Despite the differences, a main technical lemma used by Drucker (and a subsequent simplification by Dell [Del16]) is a key inspiration for our analysis.[5] We note that this lemma has previously been used for identifying sufficient conditions for obtaining cryptographic primitives from average-case hardness [BBD+20].

Lastly, we mention a recent result of Kitagawa *et al.* [KMY20], who show how to transform any SNARG (a much stronger notion than non-interactive BARG, and not known based on standard assumptions) into a NIZK, assuming one-way functions. The resulting NIZK argument is only computational zero-knowledge. In contrast, Corollary 3.16 assumes the weaker notion of non-interactive BARG and constructs the stronger notion of statistical zero-knowledge. Still, the results are incomparable as we rely on a non-uniform honest prover, and have an inverse polynomial zero knowledge error (or alternatively rely also on lossy public-key encryption).

1.3 Technical Overview

Let $\mathcal{R}$ be an NP relation, and let

$$\mathcal{R}^{\otimes t} = \left\{ \Big((x_1, \ldots, x_t), (w_1, \ldots, w_t) \Big) : |x_1| = \cdots = |x_t| \text{ and } \forall i \in [t],\ (x_i, w_i) \in \mathcal{R} \right\}$$

be the corresponding batch relation. We start by assuming a batch protocol for $\mathcal{R}^{\otimes t}$ (without specifying yet whether soundness is statistical or computational). For simplicity, let us assume that $\mathcal{R}^{\otimes t}$ has an entirely non-interactive protocol - that is, a single message sent from the prover to the verifier. We view the prover message in this case as a "compression function" f that takes as input $(x_1, \ldots, x_t, w_1, \ldots, w_t)$ and outputs a short proof string π that convinces the verifier. Note that f is an efficiently computable function, since we assume the honest prover strategy is efficient (given also the witnesses).

Since f outputs a short string, of length less than t, its output cannot contain all of the witnesses. Thus, intuitively at least, a large portion of the information about the witnesses must be lost. This leads us to the following natural idea for a protocol, for a single[6] instance of $\mathcal{R}$, that has hiding properties.

$\underline{P(x, w)}$: (where x is an input and w is a corresponding witness)

1. Choose a random index $i^* \in [t]$.
2. Select input/witness pairs $(x_i, w_i) \in \mathcal{R}$ for all $i \in [t] \backslash \{i^*\}$, in some yet-to-be-specified way.
3. Generate $\pi = f(x_1, \ldots, x_t, w_1, \ldots, w_t)$, where we fix $x_{i^*} = x$ and $w_{i^*} = w$.
4. Send $(x_1, \ldots, x_t, i^*, \pi)$ to the verifier.

The verifier V accepts if (1) $x_{i^*} = x$ and (2) the batch verifier accepts the input $(x_1, \ldots, x_t)$ with the proof π. Completeness and soundness of this protocol follow immediately from the completeness and soundness of the batch protocol (notice that for soundness, it suffices that x is a NO instance for $\mathcal{R}$ to make $(x_1, \ldots, x_t)$ a NO instance for $\mathcal{R}^{\otimes t}$.

The key question is how to choose the instance-witness pairs in Step 2 in such a way that π hides w_{i^*}. This choice is crucial. To see this, consider a contrived compression function whose goal is to be maximally non-hiding for some specific input x^*. For example, the compression function, in addition to outputting a convincing proof, might check if one of the t inputs is equal to x^*. If so, it also outputs the corresponding witness as part of the proof. Notice that this strategy is still highly compressing. While this is clearly a contrived strategy, since we seek to give a general result, that compiles *any* batch proof, we have to consider such strategies as well.

The above contrived strategy is a major concern for SWI as there exists a specific input, namely x^*, for which the prover always reveals the witness. A natural approach to circumvent this attack is to consider a

[5] We note that a closely related lemma was established earlier in the context of constructing an oblivious transfer protocol from any private information retrieval scheme [DMO00].

[6] By this we mean for an instance corresponding to $\mathcal{L}(\mathcal{R})$, the language corresponding to the relation $\mathcal{R}$.

distributional notion of SWI. That is, consider some efficiently sampleable distribution $\mathcal{D}$ supported on triples (x, w_0, w_1), where $(x, w_0), (x, w_1) \in \mathcal{R}$. Suppose we only want SWI to hold for random instance/witness pairs sampled from $\mathcal{D}$. In such a case, P can choose each (x_i, w_i) from $\mathcal{D}$ independently. Now, for inputs (x, w_0, w_1) that are also generated from $\mathcal{D}$, by symmetry, the function f will be unable to discover whether w_0 or w_1 was guessed (other than with inverse polynomial probability). Intuitively, and this can be formalized, this leads to a distributional-SWI protocol (with an SWI error that decreases polynomially with t).

While the distributional approach described above works, it is weaker than what we aim to achieve in two ways. First, it is restricted to NP languages that have a solved instance generator (recall that if the language is also hard wrt to this distribution then the sampler is a one-way function). Second, the SWI property is distributional - it only holds wrt instance-witness pairs sampled from $\mathcal{D}$ (rather than the usual worst-case guarantee).

At this point we face a problem. If we aim to get a worst-case SWI guarantee, the contrived compression function f that targets some specific x^* seems like a non-starter. Indeed, using f as a blackbox, it is hopeless to try to discover x^*. Still, if we happened to know that the compression function is precisely the contrived one described above, we could fix the same x^* as part of prover P and then use x^* (with corresponding random witnesses that are also hardwired) in *all* of the coordinates of f. Doing so would hide the specific witness that P uses in the i-th coordinate. But what about a general compression function f? Can we somehow fix specific instance/witness pairs that are specifically good for fooling f? Somewhat surprisingly the answer turns out to be yes.

How to find instance-witness pairs. Our main technical result shows that for every compression function f there exists a polynomial-size multiset $S \subseteq \mathcal{R}^{\otimes t}$ (i.e. a polynomial number of instance-witness t-tuples), so that if the tuple $((x_1, w_1), \ldots, (x_t, w_t))$ used in the above protocol is sampled uniformly from S, then the resulting protocol is SWI (with error that depends on how compressing f is).

Central to our approach is a lemma of Dell [Del16] (building on work by Drucker [Dru15] and related to a result of [DMO00]) about information lost by compressing functions. Consider a function $g : \{0,1\}^t \to \{0,1\}^{\rho t}$ for some $\rho < 1$. Intuitively, as the function is compressing, it must be losing information about some of its input bits. Dell formalized this by showing that the output distribution of g when its input bits are chosen uniformly at random is not affected much by arbitrarily fixing the bit at a randomly chosen location. Let B be the uniform distribution over $\{0,1\}^t$, and denote by $B|_{j \leftarrow b}$ the variable corresponding to sampling B and setting the j^{th} co-ordinate to b. Dell showed that in terms of statistical distance:

$$(j, g(B|_{j \leftarrow 0})) \approx_{\sqrt{\rho}} (j, g(B|_{j \leftarrow 1}))\,.$$

Suppose g is a function parameterized by triples (x_i, w_i^0, w_i^1), where $(x_i, w_i^0), (x_i, w_i^1) \in \mathcal{R}$, and uses its input bits b_i to select witness $w_i^{b_i}$, and outputs f computed with these instance-witness pairs $(x_i, w_i^{b_i})$. The above lemma would then say that picking a random $j \in [t]$ and fixing the witness used for x_j to be either of w_j^0 or w_j^1 would not make much of a difference to the output distribution of f. Denoting $(x_1, \ldots, x_t)$ by $\boldsymbol{x}$ and $(w_1, \ldots, w_t)$ by $\boldsymbol{w}$, with $j \leftarrow [t]$ and each w_i sampled uniformly from $\{w_i^0, w_i^1\}$, this implies that:

$$\left(j, \boldsymbol{x}, f(\boldsymbol{x}, \boldsymbol{w}|_{j \leftarrow w_j^0})\right) \approx \left(j, \boldsymbol{x}, f(\boldsymbol{x}, \boldsymbol{w}|_{j \leftarrow w_j^1})\right)$$

This is already reminiscent of witness-indistinguishability, though the property here only holds for a randomly chosen instance among a set of t instances. We can, in fact, use this to get the distributional version of SWI discussed above. Consider any distribution D over (x, w_0, w_1) such that $(x, w_0), (x, w_1) \in \mathcal{R}$. Now, with (x, w_0, w_1) and all the (x_i, w_i^0, w_i^1) sampled from $\mathcal{D}$, we have:

$$(j, \boldsymbol{x}|_{j \leftarrow x}, f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow w_0})) \approx (j, \boldsymbol{x}|_{j \leftarrow x}, f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow w_1}))$$

Note that in the protocol above, when the prover inserts the given (x, w) at location j and uses instances x_i and witnesses w_i in the remaining locations, the view of the verifier is precisely $(j, \boldsymbol{x}|_{j \leftarrow x}, f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow w}))$. So the above implies that the *expected* SWI *error* for the protocol when everything is sampled as specified is small.

In other words, for every distribution D over (x, w_0, w_1), there is a distribution over $\left((x_i, w_i^0, w_i^1)\right)$ such that with samples from these, the expected SWI error in our protocol is small. We can view this process as a 2-player zero-sum game: the row player chooses (x, w_0, w_1) and the column player chooses a distribution D over all such tuples. The payoff is the expected SWI error in our protocol. The above argument shows that for every strategy D for the column player there is a mixed strategy for the row player (specifically, the strategy D), for which we can bound the expected payoff. The minimax theorem now implies that there is a *single* distribution D' over tuples $\left((x_i, w_i^0, w_i^1)\right)$ such that *for every* (x, w_0, w_1), if the prover uses a sample from D' to populate the other inputs to f, the SWI error is small. Using a sparse minimax theorem [LY94] now implies the existence of a polynomial-sized multiset of $\left((x_i, w_i^0, w_i^1)\right)$'s such that sampling from this leads to almost the same SWI error. This implies the existence of the set we want, which we hard-code into the prover's algorithm as a non-uniform advice.[7]

Remark 3. *The $O(\sqrt{\rho})$ error in our analysis is tight for some functions (e.g., if g is the majority function). However, "natural" compression functions might be not exhibit such a behavior and and could potentially give rise to a negligible* SWI *error.*

Handling Multi-round Protocols. To handle multi-round protocols we follow the same basic strategy, running the underlying batch protocol using tailor-made instance/witness pairs. While we are unable to show that this approach satisfies *malicious-verifier* SWI, we manage to show that it is *honest-verifier* SWI. We do so by first extending the above analysis to 2-message protocols (i.e. a verifier message followed by a prover message). To handle protocols with more messages, we observe that when analyzing *honest-verifier* SWI, we can imagine that the verifier sends to the prover all of its randomness in advance and reduce back to the 2-message case.

Augmenting the Basic Result. At this point we have a transformation from any batch protocol into an honest-verifier SWI protocol with inverse polynomial SWI error. We can improve this state of affairs in the different settings as follows:

1. In the case of statistical soundness, if the batch proof is *public-coin*, we can rely on an information-theoretic coin-flipping protocol due to Goldreich *et al.* [GSV98] which leads to malicious verifier SWI.[8] For the case of private-coin protocols, following an approach of [BMO90, OVY93, Oka96], we show that assuming the existence of a one-way function, we can transform any honest-verifier SWI protocol to be malicious verifier SWI. We emphasize that despite the usage of a one-way function, both soundness and hiding properties are *statistical*.

2. In the case of computational soundness, assuming the existence of a one-way function, we can rely on the celebrated "FLS trick" of Feige *et al.* [FLS90] to bootstrap the honest-verifier SWI argument to an honest-verifier SZK argument.[9] Then, using the [GMW86] compiler from honest-verifier to malicious verifier we obtain a full-fledged malicious verifier zero-knowledge argument (using the [FS90] constant-round private-coin argument-system as the underlying zero-knowledge proof).

3. In the non-interactive setting: recall that in this setting the prover sends a single message, that may depend on a previously chosen common random string (CRS). One challenge that we have to deal with is that in the basic protocol, the prover needs to send its choice of $(x_1, \ldots, x_t)$ before starting the batch protocol (i.e., before the CRS is chosen), whereas in the protocol we construct this happens after the

[7] It seems tempting to try to use a uniform minmax theorem, as in [VZ13], to obtain a uniform honest prover. A key bottleneck however is that our payoff function does not seem to be efficiently computable. See also Remark 4.

[8] Note that we cannot use the honest-to-malicious transformation of Hubácek *et al.* [HRV18] (which works also in the private-coin setting) because that result relies on the connection of SZK to instance dependent commitments. Thus, it is not clear how to apply their result in the setting of SWI.

[9] In a nutshell, the verifier sends to the prover $z = G(s)$, where G is a PRG and s is a random seed, and the prover then proves that either $x \in \mathcal{L}$ or z is in the image of the PRG. Computational soundness can be argued by switching to a truly random z, and SWI by having the simulator use s as the witness.

CRS is chosen. As mentioned earlier, we handle this reversing by relying on somewhere soundness, a weak form of adaptive soundness for BARGs.

Given the resulting non-interactive SWI argument, we can use the FLS trick in a similar way to obtain a NISZKA protocol with inverse polynomial error. As our last step, we show a statistical zero-knowledge amplification theorem similar to the one by [GJS19] for computational zero knowledge. Like their transformation, we construct a combiner based on MPC-in-the-head (in our case, an information-theoretic one, such as BGW). Lossy public-key encryption is used as a dual-mode commitment — for computationally indistinguishable public keys we get either statistical hiding or statistical binding. Finally, we show based on a coupling proof, similar to the one in [LM20] that the combiner is in fact also an amplifier.

2 Definitions

We rely on the standard computational concepts and notation:

- A PPT is a probabilistic polynomial-time algorithm.

  We follow the common practice of modelling any efficient adversary strategy as a family of polynomial-size circuits. For an adversary A corresponding to a family of polynomial-size circuits $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$, we often omit the subscript λ, when it is clear from the context. We also say that such an A runs in *non-uniform polynomial time*.

- We say that a function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if for all constants $c > 0$, there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$. We sometimes denote negligible functions by negl. We say that a function $f : \mathbb{N} \to \mathbb{R}$ is *overwhelming* if $1 - f$ is negligible.

- We say that a function $f : \mathbb{N} \to \mathbb{R}$ is noticeable if there exists a constant $c > 0$ and $N \in \mathbb{N}$ such that for all $n > N$, $f(n) \geqslant n^{-c}$.

- We denote statistical distance by SD. For two random variables X, Y and $\varepsilon \in [0, 1]$, we write $X \approx_\varepsilon Y$ to denote the fact that $\mathsf{SD}(X, Y) \leqslant \varepsilon$.

- We say that an ensemble of distributions $\mathcal{D} = \{D_\lambda\}$ is efficiently sampleable if there is a polynomial p and a family of circuits $\mathsf{S} = \{\mathsf{S}_\lambda\}$ where $|\mathsf{S}_\lambda| \leqslant p(\lambda)$, and the distribution of the outputs of S_λ, on input a uniformly random string, is D_λ.

- For two ensembles $\mathcal{X} = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ and function ε, we write $\mathcal{X} \approx_\varepsilon \mathcal{Y}$ if for all large enough λ, $X_\lambda \approx_{\varepsilon(\lambda)} Y_\lambda$.

- For a (polynomially-balanced) relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, we use $\mathcal{L}(\mathcal{R})$ to denote the language defined by $\mathcal{R}$, i.e., $\{x \in \{0,1\}^* : \exists w \in \{0,1\}^* \text{ s.t. } (x, w) \in \mathcal{R}\}$. We sometimes abuse notation and write $x \in \mathcal{R}$ to mean there exists some w such that $(x, w) \in \mathcal{R}$; analogously, we write $x \notin \mathcal{R}$ to mean there is no such w.

- For a distribution X over a set Ω and $x \in \Omega$, we use $x \leftarrow X$ to denote the result of sampling according to X. For a random variable X over Ω and $x \in \Omega$, we use $X(x)$ to denote the probability that the value of the random variable is x.

2.1 Proof Systems: Soundness, Privacy, Batching

In what follows, we denote by $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ a protocol between two parties P and V. For input w for P, and common input x, we denote by $\langle \mathsf{P}(w) \leftrightarrows \mathsf{V} \rangle(x)$ the view of V in the protocol, including all received messages and random coins (if V is randomized). We abuse notation and write $\langle \mathsf{P}(w) \leftrightarrows \mathsf{V} \rangle(x) = 1$ to denote the fact that V accepts.

We next define the relevant notions of completeness, soundness, privacy, and batching. In the following definitions $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ is a protocol for an NP relation $\mathcal{R}$.

Definition 2.1 (Completeness)**.** *The protocol* $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *is* complete *with (completeness) error* $\varepsilon = \varepsilon(\lambda)$ *if for every* $(x, w) \in \mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*)$

$$\Pr\left[\langle \mathsf{P}(w) \leftrightarrows \mathsf{V} \rangle(x) = 1\right] \geqslant 1 - \varepsilon(\lambda) \ .$$

Definition 2.2 (Statistical Soundness)**.** *The protocol* $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *is* statistically sound *with (soundness) error* $\varepsilon = \varepsilon(\lambda)$ *if for every (unbounded) prover* P^* *and every large enough* $\lambda \in \mathbb{N}$, $x \in \{0,1\}^\lambda \backslash \mathcal{L}(\mathcal{R})$,

$$\Pr\left[\langle \mathsf{P}^* \leftrightarrows \mathsf{V} \rangle(x) = 1\right] \leqslant \varepsilon(\lambda) \ .$$

A statistically sound protocol is also called a proof.

Definition 2.3 (Computational Soundness)**.** *The protocol* $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *is* computationally sound *if for every polynomial-size circuit family of provers* $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_{\lambda \in \mathbb{N}}$, *there exists a negligible function* μ, *such that for all* $\lambda \in \mathbb{N}$, $x \in \{0,1\}^\lambda \backslash \mathcal{L}(\mathcal{R})$,

$$\Pr\left[\langle \mathsf{P}^*_\lambda \leftrightarrows \mathsf{V} \rangle(x) = 1\right] \leqslant \mu(\lambda) \ .$$

A computationally sound protocol is also called an argument.

Definition 2.4 (Statistical Witness Indistinguishability)**.** *The protocol* $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *is* statistically witness-indistinguishable *with error* ε *if for every polynomial-size circuit family* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda \in \mathbb{N}}$,

$$\{\langle \mathsf{P}(w_0) \leftrightarrows \mathsf{V}^*_\lambda \rangle(x)\}_{\substack{(x,w_0,w_1) \in \mathcal{R} \\ |x| = \lambda}} \approx_\varepsilon \{\langle \mathsf{P}(w_1) \leftrightarrows \mathsf{V}^*_\lambda \rangle(x)\}_{\substack{(x,w_0,w_1) \in \mathcal{R} \\ |x| = \lambda}} \ ,$$

where $(x, w_0, w_1) \in \mathcal{R}$ *is an abuse of notation to be interpreted as* $(x, w_0), (x, w_1) \in \mathcal{R}$. *If the above indistinguishability is only guaranteed for the honest verifier* V, *then* $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *is* honest-verifier *statistically witness-indistinguishable.*

Definition 2.5 (Statistical Zero Knowledge)**.** *The protocol* $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *is* statistically zero-knowledge *with error* ε *if there exists an expected PPT simulator* S *such that for every polynomial-size circuit family* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda \in \mathbb{N}}$,

$$\{\langle \mathsf{P}(w) \leftrightarrows \mathsf{V}^*_\lambda \rangle(x)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} \approx_\varepsilon \{\mathsf{S}(x, \mathsf{V}^*_\lambda)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} \ .$$

The protocol is honest-verifier statistical zero-knowledge *if the above is only guaranteed for the honest verifier* V.

Definition 2.6 (Computational Zero Knowledge)**.** *The protocol* $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *is* computationally zero-knowledge *if there exists an expected PPT simulator* S *such that for every polynomial-size circuit family* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda \in \mathbb{N}}$,

$$\{\langle \mathsf{P}(w) \leftrightarrows \mathsf{V}^*_\lambda \rangle(x)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} \approx_c \{\mathsf{S}(x, \mathsf{V}^*_\lambda)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} \ .$$

The protocol is honest-verifier computational zero-knowledge *if the above is only guaranteed for the honest verifier* V.

Definition 2.7 (Interactive Batch Protocol)**.** *A* batch protocol *for* $\mathcal{R}$ *is a protocol for* $\bigcup_{t \in \mathbb{N}} \mathcal{R}^{\otimes t}$, *where:*

$$\mathcal{R}^{\otimes t} := \left\{(x_1, \ldots, x_t), (w_1, \ldots, w_t) : |x_1| = \cdots = |x_t|, (x_1, w_1), \ldots, (x_t, w_t) \in \mathcal{R}\right\} \ . \tag{1}$$

- *The protocol's completeness and soundness errors* ($\delta(\lambda, t)$ *and* $\epsilon(\lambda, t)$) *are defined to be its largest completeness and soundness errors, respectively, on any* t *instances (and any of their witnesses) of size* λ
- *The protocol has compression rate* $\rho = \rho(\lambda, t)$, *for instance length* λ *and number of instances* t, *if maximum total length of prover messages (over all such sets of instances) is* ρt

Non-Interactive Protocols. We now define some stronger notions of soundness that we need when working with non-interactive batch protocols. A non-interactive protocol $\langle \mathsf{P} \to \mathsf{V} \rangle$ (in the CRS model) is described by a set of three algorithms $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ as follows:

- $\mathsf{Gen}(1^\lambda)$: Given the instance size λ, outputs a CRS crs
- $\mathsf{P}(crs, x, w)$: Given CRS crs, instance x, and witness w, outputs a proof π
- $\mathsf{V}(crs, x, \pi)$: Given CRS crs, instance x, and proof π, either accepts or rejects

Definition 2.8 (Completeness for Non-Interactive Protocols)**.** *$\langle \mathsf{P} \to \mathsf{V} \rangle$ has completeness error δ if for every large enough $\lambda \in \mathbb{N}$ and $(x, w) \in \mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*)$,*

$$\Pr_{\substack{crs \leftarrow \mathsf{Gen}(1^\lambda) \\ \pi \leftarrow \mathsf{P}(crs, x, w)}} [\mathsf{V}(crs, x, \pi) = 1] \geqslant 1 - \delta(\lambda).$$

For soundness, we will need the following notions.

Definition 2.9 (Non-Adaptive Computational Soundness)**.** *A non-interactive protocol $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ for a relation $\mathcal{R}$ is* non-adaptively computationally sound *if, for every $x \in \{0,1\}^\lambda \backslash \mathcal{R}$, for every polynomial-size circuit family of provers $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_{\lambda \in \mathbb{N}}$, there is a negligible function μ, such that for all $\lambda \in \mathbb{N}$:*

$$\Pr_{\substack{crs \leftarrow \mathsf{Gen}(1^\lambda) \\ \pi \leftarrow \mathsf{P}^*_\lambda(crs)}} [\mathsf{V}(crs, x, \pi) \text{ accepts}] \leqslant \mu(\lambda).$$

Definition 2.10 (Adaptive Computational Soundness)**.** *A non-interactive protocol $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ for a relation $\mathcal{R}$ is* adaptively computationally sound *if, for every polynomial-size circuit family of provers $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_{\lambda \in \mathbb{N}}$, there is a negligible function μ, such that for all $\lambda \in \mathbb{N}$:*

$$\Pr_{\substack{crs \leftarrow \mathsf{Gen}(1^\lambda) \\ (x, \pi) \leftarrow \mathsf{P}^*_\lambda(crs)}} \left[x \in (\{0,1\}^\lambda \backslash \mathcal{R}) \wedge \mathsf{V}(crs, x, \pi) \text{ accepts}\right] \leqslant \mu(\lambda).$$

Definition 2.11 (SWI for Non-Interactive Protocols)**.** *$\langle \mathsf{P} \to \mathsf{V} \rangle$ is* statistically witness-indistinguishable *with error ε if, for all large enough λ and every $(x, w_0), (x, w_1) \in \mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*)$, we have:*

$$(crs, \pi_0) \approx_\varepsilon (crs, \pi_1)$$

where $crs \leftarrow \mathsf{Gen}(1^\lambda)$, $\pi_0 \leftarrow \mathsf{P}(crs, x, w_0)$, and $\pi_1 \leftarrow \mathsf{P}(crs, x, w_1)$.

Definition 2.12 (SZK for Non-Interactive Protocols)**.** *$\langle \mathsf{P} \to \mathsf{V} \rangle$ is* statistically zero-knowledge *with error ε if there exists an expected PPT simulator S such that:*

$$\{(crs, \pi)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} \approx_\varepsilon \{\mathsf{S}(x)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} .$$

where $crs \leftarrow \mathsf{Gen}(1^\lambda)$, $\pi \leftarrow \mathsf{P}(crs, x, w)$.

Definition 2.13 (CZK for Non-Interactive Protocols)**.** *$\langle \mathsf{P} \to \mathsf{V} \rangle$ is* computationally zero-knowledge *if there exists an expected PPT simulator S such that:*

$$\{(crs, \pi)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} \approx_c \{\mathsf{S}(x)\}_{\substack{(x,w) \in \mathcal{R} \\ |x| = \lambda}} .$$

where $crs \leftarrow \mathsf{Gen}(1^\lambda)$, $\pi \leftarrow \mathsf{P}(crs, x, w)$.

Definition 2.14 (Non-Interactive Batch Protocol)**.** *We describe a non-interactive* batch *protocol by a set of PPT algorithms as follows:*

- $\mathsf{Gen}(1^\lambda, 1^t)$*: Given the instance size* λ *and the number of instances* t*, outputs a CRS* crs
- $\mathsf{TGen}(1^\lambda, 1^t, i^*)$*: Given in addition an index* $i^* \in [t]$*, outputs a CRS* crs^* *together with a trapdoor* td
- $\mathsf{P}(crs, (x_1, \ldots, x_t), (w_1, \ldots, w_t))$*: Given CRS* crs*, instances* x_i*, and witnesses* w_i*, outputs a proof* π
- $\mathsf{V}(crs, (x_1, \ldots, x_t), \pi)$*: Given CRS* crs*, instances* x_i*, and proof* π*, either accepts or rejects*

Here, the prover's communication is just the proof π*, and the compression rate is defined with respect to this.*

The following definitions of soundness properties are adapted from [CJJ22], though they have been simplified and slightly weakened as this is sufficient for our purposes.

Definition 2.15 (CRS Indistinguishability)**.** *A batch protocol* $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ *is* CRS-indistinguishable *if for every polynomial* t *and every* $i(\lambda) \in [t(\lambda)]$*, the distributions of* $\mathsf{Gen}(1^\lambda, 1^{t(\lambda)})$ *and* crs^* *sampled from* $\mathsf{TGen}(1^\lambda, 1^{t(\lambda)}, i(\lambda))$ *are computationally indistinguishable.*

Definition 2.16 (Somewhere Soundness)**.** *A batch protocol* $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ *for a relation* $\mathcal{R}$ *is* somewhere computationally sound *if it satisfies CRS indistinguishability, and for every polynomial* t *and polynomial-size circuit family of provers* $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_{\lambda \in \mathbb{N}}$*, there is a negligible function* μ *such that for all* $\lambda \in \mathbb{N}$*, letting* $t = t(\lambda)$*, and for every* $i^* \in [t]$*:*

$$\Pr_{\substack{crs, \pi \\ x_1, \ldots, x_t}} [x_{i^*} \notin \mathcal{R} \wedge \mathsf{V}(crs, (x_1, \ldots, x_t), \pi) \text{ accepts}] \leqslant \mu(\lambda),$$

where $(crs, td) \leftarrow \mathsf{TGen}(1^\lambda, 1^t, i^*)$*, and* $((x_1, \ldots, x_t), \pi) \leftarrow \mathsf{P}^*_\lambda(crs, i^*)$*.*

3 Statistical Witness Indistinguishability from Batching

In this section, we prove that a sufficiently shrinking batch protocol for a relation can be used to construct an honest-verifier statistically witness-indistinguishable protocol for it with the same soundness properties. This is captured by the following theorem. In Section 3.2, we prove a related theorem that preserves non-interactivity and stronger notions of computational soundness, which is required for our results for non-interactive BARGs. Recall that for a relation $\mathcal{R}$ and polynomial t, $\mathcal{R}^{\otimes t}$ denotes the product relation (as in Definition 2.7).

Theorem 3.1. *Consider an* NP *relation* $\mathcal{R}$*. Suppose it has a batch protocol* $\Pi = \langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ *that, when run on some polynomial* $t = t(\lambda)$ *instances of size* λ*, has compression rate* $\rho = \rho(\lambda) < 1$*. Then,* $\mathcal{R}$ *has a protocol* $\Pi_{\mathsf{WI}} = \langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ *with the following properties (on instances of size* λ*):*

- Π_{WI} *is* HVSWI *with error* $O\left(\sqrt{\rho}\right)$*.*
- Π_{WI} *has the same completeness error as* Π *run on* t *instances.*
- *If* Π *is statistically sound, then so is* Π_{WI}*, with the same soundness error as* Π *run on* t *instances.*
- *If* Π *is computationally sound, then so is* Π_{WI}*.*
- *If* P *is computed by a family of polynomial-sized circuits, then so is* P_{WI}*; and* V_{WI} *runs in uniform polynomial-time given blackbox access to* V*.*
- *The communication and round complexity in* Π_{WI} *are the same as those of* Π*, plus an additional message sent by* P_{WI} *at the start that is* $(\lambda \cdot t + \log t)$ *bits long.*

Fix some relation $\mathcal{R}$ for which there is a batch protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ with compression rate ρ as hypothesized. We will show how to construct from this a protocol $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ for $\mathcal{R}$ that inherits its soundness properties and is, in addition, HVSWI. This protocol follows the template in Fig. 1, which is parameterized by an ensemble of distributions $\mathcal{D}$ and a function t, which we will instantiate later.

Given a batch protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$, a function $t : \mathbb{N} \to \mathbb{N}$, and an ensemble of distributions $\mathcal{D} = \{D_\lambda\}$, where the support of D_λ is contained in $(\{0,1\}^\lambda \times \{0,1\}^*)^{t(\lambda)}$, the protocol $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle^{(\mathcal{D},t)}$ works as follows given an instance $x \in \{0,1\}^\lambda$ and a witness $w \in \{0,1\}^*$:

1. P_{WI} generates a sample $\{(x_i, w_i)\}_{i \in [t(\lambda)]}$ from D_λ, and samples $j \leftarrow [t(\lambda)]$
2. P_{WI} sends all the x_i's and j to V_{WI}
3. P_{WI} and V_{WI} run the protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ on the input $(x_1, \ldots, x_{j-1}, x, x_{j+1}, \ldots, x_{t(\lambda)})$, with P_{WI} using $(w_1, \ldots, w_{j-1}, w, w_{j+1}, \ldots, w_{t(\lambda)})$ as the witnesses
4. V_{WI} accepts iff the verifier V in the above execution accepts

Figure 1: Template for constructing HVSWI protocols from batch protocols

We next state lemmas capturing the properties of this protocol, and use them to prove Theorem 3.1. The proof of Lemma 3.2 is included below, and Lemma 3.3 is proven in Section 3.1.

Lemma 3.2 (Completeness and Soundness)**.** *Suppose $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ is a batch protocol for a relation $\mathcal{R}$. Let t be any polynomial and $\mathcal{D} = \{D_\lambda\}$ be such that the support of D_λ is contained within $(\mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*))^{t(\lambda)}$. Then, the protocol $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ in Fig. 1, when instantiated with $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$, $\mathcal{D}$ and t, is a protocol for $\mathcal{R}$ that satisfies the following:*

1. *If $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ has completeness error $\delta(\lambda)$ when run with $t(\lambda)$ instances of size λ, then $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle^{(\mathcal{D},t)}$ has completeness error $\delta(\lambda)$.*
2. *If $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ has statistical soundness error $\epsilon(\lambda)$ when run with $t(\lambda)$ instances of size λ, then $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle^{(\mathcal{D},t)}$ has statistical soundness error $\epsilon(\lambda)$.*
3. *If $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ is computationally sound, then $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle^{(\mathcal{D},t)}$ is also computationally sound.*

Proof. Fix any x such that $|x| = \lambda$, and denote $t(\lambda)$ by t. As all the (x_i, w_i)'s sampled from D_λ are contained in $\mathcal{R}$, the input $(x_1, \ldots, x_{j-1}, x, x_{j+1}, \ldots, x_t)$ is contained in $\mathcal{R}^{\otimes t}$ if and only if there is some w such that $(x, w) \in \mathcal{R}$. The completeness and statistical soundness errors of $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ thus carry over immediately to $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ as stated in the theorem.

For computational soundness, suppose there is a malicious prover $\mathsf{P}^*_{\mathsf{WI}}$ that can make V_{WI} accept with probability μ given an $x \notin \mathcal{R}$. Then, without loss of generality, there exists a $j \in [t]$ and $(x_1, \ldots, x_t)$ such that $\mathsf{P}^*_{\mathsf{WI}}$ can make V_{WI} accept with probability μ with the first message being $(x_1, \ldots, x_{j-1}, x, x_{j+1}, \ldots, x_t)$ and j. As V_{WI} is just emulating the verifier V, this means there is a P^* that emulates $\mathsf{P}^*_{\mathsf{WI}}$ and makes V accept on this input with probability μ. Further, if $\mathsf{P}^*_{\mathsf{WI}}$ is polynomial-time, so is P^*, as t is a polynomial. If $\mu(\lambda)$ is non-negligible, this breaks computational soundness of $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$, proving the theorem. □

Lemma 3.3 (Witness Indistinguishability)**.** *Consider a batch protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ for a relation $\mathcal{R}$ that has polynomial-sized witnesses. For a polynomial t, when the protocol is run with $t(\lambda)$ instances of size λ, suppose the total communication from the prover is at most $\rho(\lambda)t(\lambda)$ bits for some function ρ. Then, there is an efficiently sampleable ensemble of distributions $\mathcal{D} = \{D_\lambda\}$, where D_λ is supported in $(\mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*))^{t(\lambda)}$, such that the protocol $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ in Fig. 1, when instantiated with $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$, $\mathcal{D}$, and t, is* HVSWI *with error $O\left(\sqrt{\rho(\lambda)}\right)$.*

Using Lemmas 3.2 and 3.3 (the latter is proved in Section 3.1), we are ready to prove Theorem 3.1.

Proof of Theorem 3.1. Consider a relation $\mathcal{R}$ with polynomial-sized witnesses and a batch protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ that, for some polynomial t, when run on $t(\lambda)$ instances of size λ, has completeness error $\delta(\lambda)$, statistical soundness error $\epsilon(\lambda)$, and at most $\rho(\lambda)t(\lambda)$ bits of communication from the prover for some function ρ. Let $\mathcal{D}$ be the ensemble guaranteed by Lemma 3.3, and consider the protocol $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ as described in Fig. 1 instantiated with this $\mathcal{D}$ and t. This protocol has the following properties:

- Lemma 3.3 implies that this protocol is HVSWI with WI error $O\left(\sqrt{\rho(\lambda)}\right)$.
- Lemma 3.2 implies that its completeness and statistical soundness errors are $\delta(\lambda)$ and $\epsilon(\lambda)$, respectively.
- Lemma 3.2 implies that if $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ is computationally sound, then so is $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$.
- All V_{WI} does is run V on an input provided by P_{WI} and accept iff it accepts. P_{WI} also simply runs P on an input and witnesses, and in addition computes samples from D_λ and $[t(\lambda)]$, which can be done in non-uniform polynomial time since $\mathcal{D}$ is efficiently sampleable.
- In addition to the messages of $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$, the only additional communication in $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ is the initial prover message consisting of $t(\lambda)$ instances and an element of $[t(\lambda)]$.

The above arguments prove the respective properties of the protocol promised by the theorem. □

3.1 Witness Indistinguishability

In this section, we prove Lemma 3.3 about the witness indistinguishability of the protocol from Fig. 1. We will first come up with an ensemble of distributions $\mathcal{D}$ that, when used to instantiate this protocol, will make the protocol witness-indistinguishable. Fix any batch protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ for a relation $\mathcal{R}$, an instance length λ, witness length m, and the number of batch instances t. Suppose that when $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ is run on t instances of length λ, each with witness of length m, the total prover communication is at most ρt, where the compression rate ρ is less than 1.

Compressing Functions. We will use the fact that compressing functions necessarily lose information to make such a prover lose information about the witness we want to hide. This property of compression is captured by the following lemma by Dell, building on the work of Drucker [Dru15]. Similar consequences of compression have been used in the context of cryptography in the past, for instance to construct Oblivious Transfer from Private Information Retrieval protocols [DMO00, Lemma 1].

Lemma 3.4 ([Del16, Lemma 9])**.** *Let $t \in \mathbb{N}$, $\rho \in [0,1)$, and B be the uniform distribution over $\{0,1\}^t$. For any randomized mapping $f : \{0,1\}^t \to \{0,1\}^{\rho t}$, with $j \leftarrow [t]$, we have:*

$$\mathop{\mathbb{E}}_{j \leftarrow [t]} \left[\mathsf{SD}\left(f\left(B|_{j \leftarrow 0}\right), f\left(B|_{j \leftarrow 1}\right) \right) \right] \leqslant \sqrt{2 \ln 2 \cdot \rho},$$

where $B|_{j \leftarrow b}$ is the result of drawing a sample $(b_1, \ldots, b_t) \leftarrow B$ and then replacing b_j with b.

We now define a function that captures the knowledge gained by the honest verifier by interacting with the honest prover in the protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$. Its input consists of t instances $x_1, \ldots, x_t \in \{0,1\}^\lambda$, potential witnesses $w_1, \ldots, w_t \in \{0,1\}^m$, and potential random string r of V. We use $\boldsymbol{x}$ to denote $(x_1, \ldots, x_t)$ for brevity.

$f((x_1, \ldots, x_t), (w_1, \ldots, w_t), r)$:

1. Run $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ with input $(x_1, \ldots, x_t)$, using r as randomness for V, and with $(w_1, \ldots, w_t)$ as the witnesses provided to P
2. Output the sequence of prover messages in the above execution

In addition, for any pair of tuples of t potential witnesses $y_1, \ldots, y_t \in \{0,1\}^m$ and $z_1, \ldots, z_t \in \{0,1\}^m$, we define the following function on bits b_i.

$g_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},r}(b_1, \ldots, b_t)$:

1. For each $i \in [t]$, set $w_i = y_i$ if $b_i = 0$, and $w_i = z_i$ if $b_i = 1$
2. Output $f(\boldsymbol{x}, \boldsymbol{w}, r)$

The proposition below follows immediately from Lemma 3.4 and the compression of the protocol.

Proposition 3.5. *For any tuple of $x_i \in \{0,1\}^\lambda$, $y_i, z_i \in \{0,1\}^m$, and any r of the appropriate length, letting B be the uniform distribution over $\{0,1\}^t$,*

$$\mathop{\mathbb{E}}_{j \leftarrow [t]} \left[\mathsf{SD} \left(g_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},r} \left(B|_{j \leftarrow 0} \right), g_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},r} \left(B|_{j \leftarrow 1} \right) \right) \right] \leqslant \sqrt{2 \ln 2 \cdot \rho}.$$

Interpreting the function g in terms of the function f then gives the following.

Proposition 3.6. *Consider any t-tuple of $x_i \in \{0,1\}^\lambda$, $y_i, z_i \in \{0,1\}^m$, and any r of the appropriate length. For $i \in [t]$, let W_i be set to y_i or z_i uniformly at random. Then,*

$$\mathop{\mathbb{E}}_{j \leftarrow [t]} \left[\mathsf{SD} \left(f \left(\boldsymbol{x}, \boldsymbol{W}|_{j \leftarrow y_j}, r \right), f \left(\boldsymbol{x}, \boldsymbol{W}|_{j \leftarrow z_j}, r \right) \right) \right] \leqslant \sqrt{2 \ln 2 \cdot \rho}.$$

Two-Player Zero-Sum Games. Consider a two-player zero-sum game $G = (R, C, p)$, where R is the set of pure strategies for the "row" player, C the same for the "column" player, and $p : R \times C \to \mathbb{R}$ is the payoff function. Let ρ and κ denote mixed strategies for the two players, which are distributions over R and C, respectively. The *value* of this game is defined as:

$$val(G) = \min_{\rho} \max_{\kappa} \mathop{\mathbb{E}}_{\substack{r \leftarrow \rho \\ c \leftarrow \kappa}} [p(r,c)] = \max_{\kappa} \min_{\rho} \mathop{\mathbb{E}}_{\substack{r \leftarrow \rho \\ c \leftarrow \kappa}} [p(r,c)] .$$

where the equality follows from von Neumann's minimax theorem [vN28]. Lipton and Young prove the following sparse minimax theorem that will be useful for us to infer sampleable mixed strategies.

Lemma 3.7 ([LY94])**.** *Consider any two-player zero-sum game $G = (R, C, p)$ such that $p(r,c) \in [0,1]$ for any (r,c). For any $\epsilon > 0$, there is multiset $S \subseteq R$ of size $\Theta(\log |C| / \epsilon^2)$ such that for every $c \in C$:*

$$\mathop{\mathbb{E}}_{r \leftarrow S} [p(r,c)] \leqslant val(G) + \epsilon.$$

That is, there is a sparse mixed strategy that is almost as good as the optimal strategy over R. We will now define a game that captures the witness indistinguishability of the protocol described in Fig. 1, and use the above lemma to find a distribution $\mathcal{D}_\lambda$ with which to instantiate the protocol. Note that this is the first point in the proof where we involve the relation $\mathcal{R}$ that the protocols are for.

The game $G_W = (R, C, p)$ is defined with the following sets of pure strategies:

- $R = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\}$, where each vector is of length t, $x_i \in \{0,1\}^\lambda$, $y_i, z_i \in \{0,1\}^m$, and $(x_i, y_i), (x_i, z_i) \in \mathcal{R}$
- $C = \{(x, y, z)\}$, where $x \in \{0,1\}^\lambda$, $y, z \in \{0,1\}^m$, and $(x, y), (x, z) \in \mathcal{R}$

Given $r = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in R$, for each $i \in [t]$, define a random variable W_i that is set to y_i or z_i uniformly at random. The payoff function $p : R \times C \to [0,1]$ is then defined as follows, with r distributed uniformly over the appropriate domain:

$$p\left((\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), (x, y, z)\right) = \mathop{\mathbb{E}}_{j \leftarrow [t], r} \left[\mathsf{SD}\left(f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{W}|_{j \leftarrow y}, r\right), f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{W}|_{j \leftarrow z}, r\right)\right)\right]$$

Proposition 3.8. *The value of the game G_W defined above is at most $\sqrt{2 \ln 2 \cdot \rho}$.*

Proof. It is sufficient to show that for any distribution (X, Y, Z) over C, there is a distribution $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$ over R such that the expected payoff under these strategies is at most the required bound. Given such a distribution (X, Y, Z), consider $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$ defined by $(x_i, y_i, z_i) \leftarrow (X, Y, Z)$ for $i \in [t]$. The expected payoff is then as follows, with each W_i set to y_i or z_i at random:

$$\mathop{\mathbb{E}}_{\substack{(x_1,y_1,z_1) \leftarrow (X,Y,Z) \\ \cdots \\ (x_t,y_t,z_t) \leftarrow (X,Y,Z)}} \mathop{\mathbb{E}}_{(x,y,z) \leftarrow (X,Y,Z)} \mathop{\mathbb{E}}_{j \leftarrow [t], r} \left[\mathsf{SD}\left(f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{W}|_{j \leftarrow y}, r\right), f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{W}|_{j \leftarrow z}, r\right)\right)\right].$$

Noting that r and j are sampled independently of all the other quantities[10], by linearity of expectation, the above is the same as:

$$\mathop{\mathbb{E}}_{j \leftarrow [t], r} \left[\mathop{\mathbb{E}}_{\substack{(x_1,y_1,z_1) \leftarrow (X,Y,Z) \\ \cdots \\ (x_t,y_t,z_t) \leftarrow (X,Y,Z)}} \mathop{\mathbb{E}}_{(x,y,z) \leftarrow (X,Y,Z)} \left[\mathsf{SD}\left(f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{W}|_{j \leftarrow y}, r\right), f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{W}|_{j \leftarrow z}, r\right)\right)\right] \right].$$

As (x, y, z) and (x_j, y_j, z_j) are identically distributed and are independent of all other variables, this is the same as:

$$\mathop{\mathbb{E}}_{j \leftarrow [t], r} \left[\mathop{\mathbb{E}}_{\substack{(x_1,y_1,z_1) \leftarrow (X,Y,Z) \\ \cdots \\ (x_t,y_t,z_t) \leftarrow (X,Y,Z)}} \left[\mathsf{SD}\left(f\left(\boldsymbol{x}, \boldsymbol{W}|_{j \leftarrow y_j}, r\right), f\left(\boldsymbol{x}, \boldsymbol{W}|_{j \leftarrow z_j}, r\right)\right)\right] \right].$$

By Proposition 3.6 and linearity of expectation, the above is at most $\sqrt{2 \ln 2 \cdot \rho}$, which proves the proposition. □

By Lemma 3.7 and Proposition 3.8, we have the following proposition.

Proposition 3.9. *For every $\epsilon > 0$, there is a multiset $S = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\}$ of size $\Theta((\lambda + m)/\epsilon^2)$ such that:*

- *for every $i \in [t]$, both (x_i, y_i) and (x_i, z_i) are in $\mathcal{R}$*
- *for every $x \in \{0,1\}^\lambda$ and $y, z \in \{0,1\}^m$ such that $(x, y), (x, z) \in \mathcal{R}$,*

$$\mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \leftarrow S \\ w_i \leftarrow \{y_i, z_i\} \\ j \leftarrow [t], r}} \left[\mathsf{SD}\left(f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow y}, r\right), f\left(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow z}, r\right)\right)\right] \leqslant \sqrt{2 \ln 2 \cdot \rho} + \epsilon.$$

[10] This requirement of independence, specifically between r and x, is why this proof only provides honest-verifier SWI and does not work for a malicious verifier. The $\mathsf{WI}$of our protocol could potentially be broken by a malicious verifier that chooses r based on x.

Proof of Lemma 3.3. We can now describe the distribution D_λ that we will instantiate the protocol in Fig. 1 with. Recall that $\rho(\lambda)$ is the compression rate of the batch protocol we started with when run on $t(\lambda)$ instances of size λ.

> Let S be the multiset guaranteed by Proposition 3.9 for $\epsilon = \sqrt{\rho(\lambda)}$. The distribution D_λ is sampled as follows:
>
> 1. Sample $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \leftarrow S$.
> 2. For each $i \in [t(\lambda)]$, set w_i to y_i or z_i uniformly at random.
> 3. Output $\{(x_i, w_i)\}_{i \in [t(\lambda)]}$.

As S is of size $\Theta((\lambda + m(\lambda))/\epsilon^2) = \Theta((\lambda + m(\lambda))/\rho(\lambda))$, which is polynomial in λ, the distribution D_λ can be sampled non-uniformly in $\mathsf{poly}(\lambda)$ time. In any element $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ of S, we are guaranteed that each (x_i, y_i) and (x_i, z_i) is in $\mathcal{R}$. So the support of D_λ is contained in $(\mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*))^{t(\lambda)}$, as required.

To argue HVSWI of the protocol $\langle \mathsf{P_{WI}} \leftrightarrows \mathsf{V_{WI}} \rangle$ when instantiated with this distribution, we need to show that for every possible pair $(x, y), (x, z) \in \mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*)$, the views of the verifier $\mathsf{V_{WI}}$ on input x when $\mathsf{P_{WI}}$ uses y or z as the witness are statistically close. Fix any such pair.

Note that for any $(\boldsymbol{x}, \boldsymbol{w})$ sampled from D_λ, the view of $\mathsf{V_{WI}}$ on input x, when P uses witness w, is completely determined by the following quantities: $\boldsymbol{x}$, j, r, and $f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow w}, r)$ – all this is missing is the sequence of verifier messages in the protocol, which can be reconstructed efficiently given the verifier randomness r and the prover messages $f(\cdots)$. Thus, by the data processing inequality, the statistical distance between the views of $\mathsf{V_{WI}}$ in the cases where $\mathsf{P_{WI}}$ uses witness y or z is at most the following, where $(\boldsymbol{x}, \boldsymbol{w}) \leftarrow D_\lambda$, $j \leftarrow [t(\lambda)]$, and r is over the appropriate domain:

$$\mathsf{SD}\left((\boldsymbol{x}, j, r, f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow y}, r)), (\boldsymbol{x}, j, r, f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow z}, r))\right).$$

Taking into account the definition of D_λ, this is equal to:

$$\mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \leftarrow S \\ w_i \leftarrow \{y_i, z_i\} \\ j \leftarrow [t], r}} \left[\mathsf{SD}\left(f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow y}, r), f(\boldsymbol{x}|_{j \leftarrow x}, \boldsymbol{w}|_{j \leftarrow z}, r)\right)\right],$$

which, by Proposition 3.9, is at most $\sqrt{2 \ln 2 \cdot \rho(\lambda)} + \epsilon = O(\sqrt{\rho(\lambda)})$. This proves the Lemma 3.3. □

Remark 4. *The prover $\mathsf{P_{WI}}$ in protocol $\langle \mathsf{P_{WI}} \leftrightarrows \mathsf{V_{WI}} \rangle$ we construct is non-uniform even if the prover P from the original batch protocol is uniform. This is because the minimax theorem we use (Lemma 3.7), while constructive, is not uniform. An interesting question here is whether a uniform version of the minimax theorem can be used instead to preserve uniformity of the prover. As far as we can tell, existing uniform minimax theorems ([VZ13], for instance) do not seem useful for this purpose. They require the payoff of the game to be efficiently computable given the strategies, which does not seem to be the case here as it involves computing the statistical distance between two rather arbitrary distributions.*

Remark 5. *The bound of $O(\sqrt{\rho})$ in the statements above (and particularly in Lemma 3.4) is optimal upto constant factors. In the case of Lemma 3.4, a function g that splits its input into blocks of size $\Theta(1/\rho)$ and outputs the majority of the bits in each block witnesses this optimality. This can then be extended to proof systems, where the bits may represent predicates that distinguish between two witnesses.*

3.2 Non-Interactive Protocols

In this section, we prove a version of Theorem 3.1 for non-interactive protocols. It preserves non-interactivity and considers adaptive notions of soundness.

Theorem 3.10. *Suppose an* NP *relation* $\mathcal{R}$ *has a non-interactive batch protocol* $\Pi = (\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ *that, when run on some polynomial* $t = t(\lambda)$ *instances of size* λ*, has compression rate* $\rho = \rho(\lambda) < 1$*. Then,* $\mathcal{R}$ *has a non-interactive protocol* $\Pi_{\mathsf{WI}} = (\mathsf{Gen}_{\mathsf{WI}}, \mathsf{P}_{\mathsf{WI}}, \mathsf{V}_{\mathsf{WI}})$ *with the following properties (on instances of size* λ*):*

- Π_{WI} *is* SWI *with error* $O\left(\sqrt{\rho}\right)$*.*
- *If* Π *is CRS-indistinguishable, then* Π_{WI} *has completeness error negligibly close to that of* Π *run on* t *instances.*
- *If* Π *is somewhere computationally sound, then* Π_{WI} *is adaptively computationally sound.*
- *If* P *is computed by a family of polynomial-sized circuits, then so is* P_{WI}*; and* V_{WI} *and* $\mathsf{Gen}_{\mathsf{WI}}$ *run in uniform polynomial-time given blackbox access to* V *and* TGen*, respectively.*
- *The length of the proof in* Π_{WI} *is that in* Π *plus an additional* $\lambda \cdot t$ *bits. The length of the CRS is the same.*

Fix some relation $\mathcal{R}$ for which there is a non-interactive batch protocol $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ with compression rate ρ as hypothesized. We will show how to construct from this a non-interactive SWI protocol $(\mathsf{Gen}_{\mathsf{WI}}, \mathsf{P}_{\mathsf{WI}}, \mathsf{V}_{\mathsf{WI}})$ for $\mathcal{R}$ in a manner similar to that earlier in this section for general interactive protocols. This protocol follows the template in Fig. 1, which is parametrised by an ensemble of distributions $\mathcal{D}$ and a function t, which we will instantiate later.

Given a non-interactive batch protocol $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$, a function $t : \mathbb{N} \to \mathbb{N}$, and an ensemble of distributions $\mathcal{D} = \{D_\lambda\}$, where the support of D_λ is contained in $(\{0,1\}^\lambda \times \{0,1\}^*)^{t(\lambda)}$, the non-interactive protocol $(\mathsf{Gen}_{\mathsf{WI}}, \mathsf{P}_{\mathsf{WI}}, \mathsf{V}_{\mathsf{WI}})$ are as follows.

$\underline{\mathsf{Gen}_{\mathsf{WI}}(1^\lambda)}$:

- Sample $j \leftarrow [t(\lambda)]$, and $crs \leftarrow \mathsf{TGen}(1^\lambda, 1^{t(\lambda)}, j)$.
- Output (j, crs).

$\underline{\mathsf{P}_{\mathsf{WI}}((j, crs), x, w)}$:

- Sample $\{(x_i, w_i)\}_{i \in [t(\lambda)]}$ from D_λ.
- Compute $\pi \leftarrow \mathsf{P}(crs, (x_1, \ldots, x_{j-1}, x, x_{j+1}, \ldots, x_{t(\lambda)}), (w_1, \ldots, w_{j-1}, w, w_{j+1}, \ldots, w_{t(\lambda)}))$.
- Output $(\boldsymbol{x}, \pi)$.

$\underline{\mathsf{V}_{\mathsf{WI}}((j, crs), x, (\boldsymbol{x}, \pi))}$:

- Accepts iff $\mathsf{V}(crs, (x_1, \ldots, x_{j-1}, x, x_{j+1}, \ldots, x_{t(\lambda)}), \pi)$ accepts.

Figure 2: Template for constructing non-interactive SWI protocols from non-interactive batch protocols

We next state lemmas capturing the properties of this protocol, and use them to prove Theorem 3.10.

Lemma 3.11 (Completeness and Soundness)**.** *Suppose* $\Pi = (\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ *is a non-interactive batch protocol for a relation* $\mathcal{R}$*. Let* t *be any polynomial and* $\mathcal{D} = \{D_\lambda\}$ *be such that the support of* D_λ *is contained within* $(\mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*))^{t(\lambda)}$*. Then, the protocol* $\Pi_{\mathsf{WI}} = (\mathsf{Gen}_{\mathsf{WI}}, \mathsf{P}_{\mathsf{WI}}, \mathsf{V}_{\mathsf{WI}})$ *in Fig. 2, when instantiated with* Π*,* $\mathcal{D}$ *and* t*, is a non-interactive protocol for* $\mathcal{R}$ *that satisfies the following:*

1. *If* Π *has completeness error* $\delta(\lambda)$ *when run with* $t(\lambda)$ *instances of size* λ *and is CRS-indistinguishable, then* Π_{WI} *has completeness error at most* $\delta(\lambda) + \mathrm{negl}(\lambda)$.

2. *If* Π *is somewhere computationally sound, then* Π_{WI} *is adaptively computationally sound.*

Proof. Let $t = t(\lambda)$. If in $\mathsf{Gen}_{\mathsf{WI}}(1^\lambda)$, the crs had been sampled from $\mathsf{Gen}(1^\lambda, 1^t)$ instead of $\mathsf{TGen}(1^\lambda, 1^t, j)$, then the completeness of Π_{WI} follows that of Π, with the same error $\delta(\lambda)$ (by the same arguments as in Lemma 3.2). By the CRS-indistinguishability of Π, and as both P_{WI} and V_{WI} are polynomial-time algorithms, making this change in $\mathsf{Gen}_{\mathsf{WI}}$ can only change the completeness error by a negligible amount.

For soundness, suppose there is a malicious prover $\mathsf{P}^*_{\mathsf{WI}}$ and a non-negligible function μ such that, with $crs \leftarrow \mathsf{Gen}_{\mathsf{WI}}(1^\lambda)$ and $(x, \pi) \leftarrow \mathsf{P}^*_{\mathsf{WI}}(crs)$ we have:

$$\Pr\left[x \in \{0,1\}^\lambda \backslash \mathcal{L}(\mathcal{R}) \wedge \mathsf{V}_{\mathsf{WI}}(crs, x, \pi) \text{ accepts}\right] \geqslant \mu(\lambda).$$

By the definition of the protocol, the above is the same as the following: with $j \leftarrow [t]$, $crs \leftarrow \mathsf{TGen}(1^\lambda, 1^t, j)$, $(x, \boldsymbol{x}, \pi) \leftarrow \mathsf{P}^*_{\mathsf{WI}}(j, crs)$,

$$\Pr\left[x \in \{0,1\}^\lambda \backslash \mathcal{L}(\mathcal{R}) \wedge \mathsf{V}(crs, \boldsymbol{x}|_{j \leftarrow x}, \pi) \text{ accepts}\right] \geqslant \mu(\lambda),$$

which immediately contradicts the somewhere computational soundness of $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ if μ is non-negligible. This proves the lemma. □

Lemma 3.12 (Witness Indistinguishability)**.** *Consider a non-interactive batch protocol* $\Pi = (\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ *for a relation* $\mathcal{R}$ *that has polynomial-sized witnesses. For a polynomial* t*, when the protocol is run with* $t(\lambda)$ *instances of size* λ*, suppose the length of the proof is at most* $\rho(\lambda)t(\lambda)$ *bits for some function* ρ*. Then, there is an efficiently sampleable ensemble of distributions* $\mathcal{D} = \{D_\lambda\}$*, where* D_λ *is supported in* $(\mathcal{R} \cap (\{0,1\}^\lambda \times \{0,1\}^*))^{t(\lambda)}$*, such that the protocol* $(\mathsf{Gen}_{\mathsf{WI}}, \mathsf{P}_{\mathsf{WI}}, \mathsf{V}_{\mathsf{WI}})$ *in Fig. 2, when instantiated with* Π*,* $\mathcal{D}$*, and* t*, is* SWI *with error* $O\left(\sqrt{\rho(\lambda)}\right)$.

Proof Sketch. The proof of this lemma is identical to that of Lemma 3.3, with the only difference being that instead of the verifier's random string r, here we use the CRS sampled by $\mathsf{Gen}_{\mathsf{WI}}$. □

Proof of Theorem 3.10. Consider a relation $\mathcal{R}$ with polynomial-sized witnesses and a non-interactive batch protocol $\Pi = (\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ that, for some polynomial t, when run on $t(\lambda)$ instances of size λ, has completeness error $\delta(\lambda)$, statistical soundness error $\epsilon(\lambda)$, and proofs of length at most $\rho(\lambda)t(\lambda)$ bits for some function ρ. Let $\mathcal{D}$ be the ensemble guaranteed by Lemma 3.12, and consider the protocol $\Pi_{\mathsf{WI}} = (\mathsf{Gen}_{\mathsf{WI}}, \mathsf{P}_{\mathsf{WI}}, \mathsf{V}_{\mathsf{WI}})$ as described in Fig. 2 instantiated with this $\mathcal{D}$ and t. This protocol has the following properties:

- Lemma 3.12 implies that this protocol is SWI with WI error $O\left(\sqrt{\rho(\lambda)}\right)$.
- Lemma 3.11 implies that its completeness error is $\delta(\lambda) + \mathrm{negl}(\lambda)$.
- By Lemma 3.11, if Π is somewhere computationally sound, then Π_{WI} is adaptively computationally sound.
- All V_{WI} does is run V on an input provided by P_{WI} and accept iff it accepts. $\mathsf{Gen}_{\mathsf{WI}}$ similarly only samples from $[t(\lambda)]$ and runs TGen once. P_{WI} also simply runs P on an input and witnesses, and in addition computes samples from D_λ and $[t(\lambda)]$, which can be done in non-uniform polynomial time since $\mathcal{D}$ is efficiently sampleable.
- In addition to the proof from Π, the proof in Π_{WI} consists only of the $t(\lambda)$ instances of length λ sampled by P_{WI}.

The above arguments prove the respective properties of the protocol promised by the theorem. □

3.3 Corollaries

In this section we state some of the known results on transforming HVSWI protocols into SWI and SZK protocols against malicious verifiers. Starting from a public-coin HVSWI proof, Corollary 3.13 gives an SWI proof against malicious verifiers without any additional assumptions. Even if the original HVSWI proof is not public coin, we can use it to obtain an SWI proof under computational assumptions. This transformation is given by Corollary 3.14. Moving on to the setting of computational soundness, assuming OWFs exist, Corollary 3.14 shows that any HVSWI argument can be transformed into an SWI argument. Under the same assumption, Corollary 3.15 gives a transformation from an SWI argument to an SZK argument. Finally, Corollary 3.16 gives a similar transformation from an SWI argument to an SZK argument for non-interactive protocols.

Corollary 3.13. *If there exists a public-coin* HVSWI *proof* Π *for an* NP *relation* $\mathcal{R}$ *then there exists a public-coin* SWI *proof* Π_M *for* $\mathcal{R}$ *with the following properties:*

- Π_M *has negligible completeness and soundness error.*
- Π_M *has* WI *error* $\mathsf{poly}(\lambda) \cdot \varepsilon + 2^{-\Theta(\lambda)}$ *where* λ *is the instance length and* ε *is* WI *error of* Π.
- *If* Π *has* d *rounds then* Π_M *has* $2d$ *rounds.*
- *If the honest prover in* Π *is non-uniform then so is the honest prover in* Π_M.

Proof Sketch. The proof is based on [Vad99, Theorem 6.3.5] which gives a transformation from any public-coin HVSZK proof to a public-coin SZK proof with the following properties:

- Π_M has negligible completeness error $2^{-\lambda}$ and soundness error $1/\lambda$.
- Π_M has ZK error $\mathsf{poly}(\lambda) \cdot \varepsilon + 2^{-\Theta(\lambda)}$ where λ is the instance length and ε is ZK error of Π.
- If Π has d rounds then Π_M has $2d$ rounds.
- If the honest prover in Π is non-uniform then so is the honest prover in Π_M.

We observe that the same transformation also transforms any HVSWI proof to an SWI proof with the same properties. To see that, recall that a protocol $\langle \mathsf{P} \leftrightarrows \mathsf{V} \rangle$ is SWI with error ε if and only if there exists an (unbounded) simulator S such that for every polynomial-size circuit family $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_{\lambda \in \mathbb{N}}$,

$$\{\langle \mathsf{P}(w) \leftrightarrows \mathsf{V}^*_\lambda \rangle(x)\}_{\substack{(x,w)\in\mathcal{R} \\ |x|=\lambda}} \approx_\varepsilon \{\mathsf{S}(x, \mathsf{V}^*_\lambda)\}_{\substack{(x,w)\in\mathcal{R} \\ |x|=\lambda}} \quad .$$

Similarly, the protocol is HVSWI if and only if the above is guaranteed for the honest verifier V. The proof of [Vad99, Theorem 6.3.5] uses the polynomial time simulator of the original SZK proof to construct a polynomial-time simulator for the new proof system. By inspecting the transformation and its analysis, we conclude that if the original protocol has an unbounded simulator instead of a polynomial-time one, the simulator constructed for the new proof system is also unbounded with the same error as in the efficient case.

The SWI proof resulting from the [Vad99] transformation has a non-negligible soundness error. To get an SWI proof with the claimed properties we can repeat the resulting SWI proof in parallel $\mathsf{poly}(\lambda)$ times. This reduces the soundness error to negligible and only increases the WI error by a factor of $\mathsf{poly}(\lambda)$.

□

Corollary 3.14. *Let* Π *be an* HVSWI *protocol for an* NP *relation* $\mathcal{R}$ *with* d *rounds.*

- *Assume there exists a statistically hiding commitment with* d^* *rounds. If* Π *is statistically sound then there exists an* SWI *proof for* $\mathcal{R}$ *with* $O(d \cdot d^*)$ *rounds and soundness error that is negligibly close to that of* Π.

- *Assuming OWFs exist, if* Π *is computationally sound then there exists an* SWI *argument for* $\mathcal{R}$ *with* $O(d)$ *rounds.*

The WI *error of the new protocol is negligibly close to that of* Π*. If the honest prover in* Π *is non-uniform then so is the honest prover in the new protocol.*

Statistically-hiding commitment be constructed in two rounds from CRHFs [DPP97, FS90, HM96], in a constant number of rounds from multi-collision resistant hash functions [BDRV18, KNY18] or distributional CRHFs [BHKY19], and in $O(\lambda)$ rounds from OWFs [HNO+09].

Proof Sketch. The proof is based on the compiler of [GMW86]. We start with the case of computational soundness and then explain how to modify the protocol to obtain statistical soundness. The verifier starts by committing to a random string r_V using a statistically-binding commitment and the prover responds with a random string r_P. Then the prover and verifier execute the HVSWI protocol where the verifier uses the randomness $r_V \oplus r_P$. After each message, the verifier proves using a computational ZK argument that the message was generated correctly. The SWI argument $\langle \mathsf{P}_M \leftrightarrows \mathsf{V}_M \rangle$ is described in Fig. 3. The construction uses a two-message statistically binding commitment and a constant-round computational ZK argument, both of which can be constructed from OWFs [Nao91, FS90]. Next, we sketch the proof of soundness and SZK.

Computational soundness. Assume towards contradiction that there exists a polynomial-size cheating prover P^* that can prove a false statement with non-negligible probability ϵ. We use P^* to break the computational soundness of the HVSWI argument $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$. First we consider a hybrid experiment where we emulate an execution P^* with the verifier V_M, but each execution of the ZK argument $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ is simulated. By the zero-knowledge property of the ZK argument, P^* will continue to produce accepting proofs with probability that is negligibly close to ϵ. In the next hybrid, we modify the value in the initial commitment sent by V_M from r_V to 0^ℓ. By the computational hiding property of the commitment, P^* will continue to produce accepting proofs with probability that is negligibly close to ϵ. Now we can break the soundness of the HVSWI argument $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ by emulating this final hybrid experiment and forwarding the messages of the external verifier V_{WI} to P^* instead of computing them using the randomness $r = r_V \oplus r_P$. Since the string r is uniformly distributed, we convince the external verifier of a false statement with probability that is negligibly close to ϵ.

SWI. Fix any polynomial-size cheating verifier V^* and statement-witness pairs $(x, w_0), (x, w_1) \in \mathcal{R}$. Let ϵ denote the distance between the views $\mathsf{View}_0 = \langle \mathsf{P}_M(w_0) \leftrightarrows \mathsf{V}^* \rangle(x)$ and $\mathsf{View}_1 = \langle \mathsf{P}_M(w_1) \leftrightarrows \mathsf{V}^* \rangle(x)$. Since the commitment COM is statistically binding, we can fix the first commitment message k sampled by P_M such that COM_k is perfectly binding and the distance between View_0 and View_1 remains negligibly close to ϵ. Let $\tilde{r}_V$ be the string that V^* commits to in its first message.

For $b \in \{0, 1\}$ we consider the view of the honest verifier V_{WI} in the interaction of $\langle \mathsf{P}_{\mathsf{WI}}(w_b) \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle(x)$ which consist of the verifier's randomness r and the prover's messages $(\beta_1, \ldots, \beta_d)$. We argue that given this view we can efficiently sample from a distribution that is negligibly close to View_b (with the first commitment message fixed to k). Therefore, it follows that ϵ must be negligibly close to the SWI error of the HVSWI argument.

Given the view $r, (\beta_1, \ldots, \beta_d)$ we sample from a distribution close to View_b as follows. We emulate the execution of V^*, setting the first prover message to k and the second prover message to $r_P = \tilde{r}_V \oplus r$. Since r is uniform, r_P is distributed exactly as in View_b. In every one of the remaining d rounds, starting from $i = 1$ to d we interact with V^* emulating the verifier of the ZK argument $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$. If the ZK argument is accepted then we set the next prover message to β_i, otherwise the prover aborts.

Let E be the event that the verifier V^* proves a true statement in each of the accepting executions of the ZK argument $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$. Conditioned on E, the view sampled above is distributed exactly the same as View_b. By the computational soundness property of the ZK argument, E occurs with all but negligible probability. Therefore, the sampled view is negligibly close to View_b.

An SWI proof. If the original SWI protocol $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ has statistical soundness we can modify the protocol $\langle \mathsf{P}_M \leftrightarrows \mathsf{V}_M \rangle$ described in Fig. 3 and obtain an SWI proof. We make the following modifications:

- We replace the two-message statistically-binding commitment with a statistically-hiding commitment.
- After the verifier sends the commitment c and before the prover sends r_P, have the verifier prove that it knows an opening of c using a SZK argument of knowledge where SZK holds even against an unbounded malicious verifier.[11] (We describe how this SZK argument of knowledge is constructed below.)
- Replace each invocation of the computational ZK argument with a SZK argument of knowledge against an unbounded malicious verifier.

Since the verifier's commitment and ZK arguments are all statistical, we can show statistical soundness following the same argument as in the computational case. To prove SWI, modify the above proof as follows. Since the commitment c is statistically-hiding, the string $\tilde{r}_V$ that V* commits to is not well defined. Instead, we invoke the knowledge extractor of the SZK argument of knowledge and extract an opening to a string $\tilde{r}_V$. To prove SWI we need to show that, with all but negligible probability, all the messages $\beta_1, \ldots, \beta_d$ are computed according to the strategy of the honest verifier in the HVSWI argument V_{WI} using the randomness $\tilde{r}_V \oplus r_P$. If this is not the case for some β_i, we can use the knowledge extractor of the SZK argument of knowledge and obtain an opening of the commitment c to a value other than r_V with non-negligible probability, contradicting the computational binding property of the commitment.

Using a statistically-hiding commitment with d^* rounds, a SZK argument of knowledge against an unbounded malicious verifier with $O(d^*)$ rounds can be constructed following the outline of [FLS90, GK96]: Start from an SWI argument of knowledge against an unbounded verifier in $O(d^*)$ rounds. Such a protocol can be obtained by taking the parallel repetition of the ZK protocol of [Blu81, GMW86] and instantiating the commitment scheme with the statistically hiding commitment. Next, the SWI argument of knowledge is transformed into an SZK argument of knowledge using the compiler of [FLS90]. In more details, the verifier starts by committing to trapdoor statement using a statistically hiding commitment and proving that the committed statement is true using a computational ZK proof of knowledge. Then the prover uses the SWI argument of knowledge to prove that either the original statement or the committed trapdoor statement is true. The required computational ZK proof of knowledge can be constructed by combining the computational ZK proof of [GK96] (instantiated with the statistically-hiding commitment) with a computational WI proof of knowledge (given by parallel repetition of the ZK protocol of [Blu81, GMW86], instantiated with a statistically binding commitment) via the [FLS90] compiler.

[11] This is in contrast to the weaker notion of SZK in Definition 2.5 that only considers polynomial-size malicious verifiers.

Let COM be a two-message statistically binding commitment. Let $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ be a constant-round computational ZK argument. Let $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ be a d-round HVSWI argument where verifier's randomness is of length ℓ. Assume WLOG that d is even.
The SWI argument $\langle \mathsf{P}_M \leftrightarrows \mathsf{V}_M \rangle$ is as follows. The prover and verifier are given an instance $x \in \{0,1\}^\lambda$. The prover is also given a witness $w \in \{0,1\}^*$.

1. P_M samples a first message k for COM and sends it to V_M
2. V_M samples $r_V \leftarrow \{0,1\}^\ell$ and a commitment $c \leftarrow \mathsf{COM}_k(r_V)$, and sends c to P_M
3. P_M samples $r_P \leftarrow \{0,1\}^\ell$ sends it to V_M
4. For $i = 1, \ldots, d$:
   (a) V_M computes the next message of V_{WI} using randomness $r = r_V \oplus r_P$:
   $$\alpha_i \leftarrow \mathsf{V}_{\mathsf{WI}}(x, \alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1}; r) \ ,$$
   and sends α_i to P_M
   (b) V_M and P_M execute the protocol $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ where V_M proves to P_M that there exist strings $\tilde{r}_V$ and σ such that:
   $$c = \mathsf{COM}_k(\tilde{r}_V; \sigma) \ \wedge \ \alpha_i = \mathsf{V}_{\mathsf{WI}}(x, \alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1}; \tilde{r}_V \oplus r_P)$$
   (c) If $\mathsf{V}_{\mathsf{CZK}}$ rejects then P_M aborts. Otherwise, P_M computes the next message of P_{WI}:
   $$\beta_i \leftarrow \mathsf{P}_{\mathsf{WI}}(x, w, \alpha_1, \beta_1, \ldots, \alpha_i) \ ,$$
   and sends β_i to V_M

Figure 3: Malicious-verifier SWI argument from an HVSWI argument and OWFs

□

Corollary 3.15. *Assuming one-way functions exist, if there exists an* SWI *argument* Π *for an* NP*-complete relation* $\mathcal{R}$ *with* d *rounds, then there exists an* SZK *argument* Π_{ZK} *for* $\mathcal{R}$ *with* $O(d)$ *rounds and ZK error that is negligibly close to the WI error of* Π*. If the honest prover in* Π *is non-uniform then so is the honest prover in* Π_{ZK}*.*

Proof Sketch. The proof is based on the compiler of [FLS90]. The verifier starts by sending a random image y of a length-doubling PRG and proving that it knows a corresponding preimage using a computational ZK argument of knowledge. Then, the prover and verifier execute the SWI protocol proving that either the original statement is correct or that y is in the image of the PRG. The SZK argument $\langle \mathsf{P}_{\mathsf{ZK}} \leftrightarrows \mathsf{V}_{\mathsf{ZK}} \rangle$ is described in Fig. 4. The construction uses a PRG and a constant-round computational ZK argument of knowledge, both of which can be constructed from OWFs [HILL99, FS90]. Next, we sketch the proof of soundness and SZK.

Soundness. Assume towards contradiction that there exists a polynomial-size cheating prover P^* that can prove a false statement with non-negligible probability ϵ. We use P^* to break the computational soundness of the SWI argument $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$. First we consider a hybrid experiment where we emulate an execution P^* with the verifier V_{ZK}, but the execution of the computational ZK argument $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ is simulated. By the zero-knowledge property of the ZK argument, P^* will continue to produce accepting proofs with

probability that is negligibly close to ϵ. In the next hybrid, we sample a uniform $y \leftarrow \{0,1\}^{2\lambda}$ instead of sampling y as a random image of the PRG. By the pseudorandomness of the generator, P^* will continue to produce accepting proofs with probability that is negligibly close to ϵ. Now, the statement for the SWI argument $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ is false with probability $1 - 2^{-\lambda}$. Therefore, we can break the soundness of $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ with probability that is negligibly close to ϵ.

SZK. We describe a simulator S. The simulator is given an instance $x \in \mathcal{L}(\mathcal{R})$ and the description of a cheating verifier V^*. S emulates an interaction with V^*. If the verifier $\mathsf{V}_{\mathsf{CZK}}$ rejects in the execution of the computational ZK argument of knowledge $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ then S outputs the transcript of the interaction with V^* up to that point. Otherwise, S invokes the knowledge extractor of $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ on the description of the residual verifier V^* after sending its first message, right before the execution of $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$. If the extractor fails to output a string r such that $y = \mathsf{PRG}(r)$ then S aborts. Otherwise, S continues to emulate an interaction with V^* by executing the honest prover of the SWI argument $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ using the witness $\tilde{w} = \bot$ and $\tilde{r} = r$. Finally, S outputs the transcript of the entire interaction with V^*. Fix any $(x, w) \in \mathcal{R}$ and a polynomial-size cheating verifier V^*. Since the extractor runs in time that is inverse polynomial to the probability that the proof given by V^* is accepted, it follows that S runs in expected polynomial time. By the knowledge soundness property of the computational ZK argument of knowledge the extractor fails to find a witness and the simulation aborts only with negligible probability. Conditioned on the fact that S does not abort, the only difference between the simulated view generated by $\mathsf{S}(x, \mathsf{V}^*)$ and the real view $\langle \mathsf{P}_{\mathsf{ZK}}(w) \leftrightarrows \mathsf{V}^* \rangle(x)$ is the witness used by the honest prover in the execution of $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$. Therefore, the ZK error of the SWI argument $\langle \mathsf{P}_{\mathsf{ZK}} \leftrightarrows \mathsf{V}_{\mathsf{ZK}} \rangle$ is negligibly close to the WI error of the SWI argument $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$.

Let $\mathsf{PRG} : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda}$ be a length-doubling PRG. Let $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ be a constant-round computational ZK argument of knowledge. Let $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ be an SWI argument.
The SZK argument $\langle \mathsf{P}_{\mathsf{ZK}} \leftrightarrows \mathsf{V}_{\mathsf{ZK}} \rangle$ is as follows. The prover and verifier are given an instance $x \in \{0,1\}^{\lambda}$. The prover is also given a witness $w \in \{0,1\}^*$.

1. V_{ZK} samples a string $r \leftarrow \{0,1\}^{\lambda}$ and sends $y = \mathsf{PRG}(r)$ to P_{ZK}
2. V_{ZK} proves to P using $\langle \mathsf{P}_{\mathsf{CZK}} \leftrightarrows \mathsf{V}_{\mathsf{CZK}} \rangle$ that there exists a string $\tilde{r}$ such that $y = \mathsf{PRG}(\tilde{r})$
3. If $\mathsf{V}_{\mathsf{CZK}}$ rejects then P_{ZK} aborts. Otherwise, P_{ZK} proves to V_{ZK} using $\langle \mathsf{P}_{\mathsf{WI}} \leftrightarrows \mathsf{V}_{\mathsf{WI}} \rangle$ that there exist strings $\tilde{w}$ and $\tilde{r}$ such that:
$$(x, \tilde{w}) \in \mathcal{R} \ \vee \ y = \mathsf{PRG}(\tilde{r}) \ ,$$
using the witness $\tilde{w} = w$ and $\tilde{r} = \bot$

Figure 4: SZK argument from an SWI argument and OWFs

□

Corollary 3.16. *Assuming one-way functions exist, if there exists a non-interactive* SWI *argument* Π *for an* NP*-complete relation* $\mathcal{R}$ *in the CRS model, then there exists a non-interactive* SZK *argument* Π_{ZK} *for* $\mathcal{R}$ *in the CRS model with ZK error that is the same as the WI error of* Π*. If the honest prover in* Π *is non-uniform then so is the honest prover in* Π_{ZK}*.*

Proof Sketch. The proof is similar to that of Corollary 3.15 except that the verifier's first message containing a random image y of a length-doubling PRG is generated as part of the CRS and the verifier does not prove knowledge of the preimage of y. Soundness follows by the same argument as in Corollary 3.15. In the proof of SZK, the simulator samples the CRS including y itself and, therefore, it has the corresponding preimage. □

4 NISZK Privacy Amplification

The WI and ZK errors for the protocols obtained in the previous section were only inverse-polynomially small. In this section, we show how to reduce this error to negligibly small for non-interactive protocols, assuming *lossy public-key encryption* (LPKE). To be specific, given a NISZKA with a sufficiently-small inverse-polynomial error, we use LPKE to obtain a NISWIA with negligible WI error (Theorem 4.4). By Corollary 3.16, this (together with the fact that LPKE implies OWF) implies NISZKA with negligible ZK error. (We note that our amplification approach would in fact also work in the interactive setting, but would result in protocols with at least four rounds, where SWI arguments are already known from two-message statistically hiding commitments, and in particular from LPKE.)

We further show in Appendix B that LPKE can be generically constructed (Theorem B.3) from non-interactive BARGs that are *honestly* somewhere extractable.

4.1 Definitions

Lossy public-key encryption (LPKE). We recall the definition of LPKE.

Definition 4.1 (LPKE)**.** *A* lossy public-key encryption *scheme Λ with message-space $\mathcal{M}$ and ciphertext-space $\mathcal{C}$ is a tuple of polynomial-time algorithms* $(\mathsf{KGen}, \mathsf{LGen}, \mathsf{E}, \mathsf{D})$ *with following syntax:*

- $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$*. The randomised (normal) key-generation algorithm, on input a security parameter $\lambda \in \mathbb{N}$, outputs a public-private key-pair (pk, sk). We refer to a public key generated by* KGen *as a* real *key.*
- $pk^* \leftarrow \mathsf{LGen}(1^\lambda)$*. The randomised* lossy *key-generation algorithm, on input a security parameter $\lambda \in \mathbb{N}$, outputs a public key pk^*, which we refer to as a* lossy *key.*
- $c \leftarrow \mathsf{E}(pk, m)$*. The randomised encryption algorithm takes as input a public key pk and a message $m \in \mathcal{M}$, and outputs a ciphertext $c \in \mathcal{C}$.*
- $m := \mathsf{D}(sk, c)$*. The deterministic decryption algorithm takes a secret key sk and a ciphertext $c \in \mathcal{C}$ as input and outputs a message $m \in \mathcal{M}$.*

We require the following properties from Λ:

1. Real public keys are almost-all-keys perfectly correct . *With overwhelming probability over the choice of real keys, perfect correctness of decryption must hold. That is, with overwhelming probability over $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$, for every $m \in \mathcal{M}$*
$$\Pr_{c \leftarrow \mathsf{E}(pk, m)} [\mathsf{D}(sk, c) \neq m] = 0.$$
2. Lossy keys are statistically hiding. *For a random* lossy *key, the distribution of ciphertexts of any two messages must be* statistically *close. To be specific, we say that the lossy keys are δ-*statistically-hiding *if for $(pk, sk) \leftarrow \mathsf{LGen}(1^\lambda)$ and $m_0, m_1 \in \mathcal{M}$:*
$$\mathsf{SD}(\mathsf{E}(pk, m_0), \mathsf{E}(pk, m_1)) \leqslant \delta(\lambda).$$
3. Real and lossy keys are computationally indistinguishable. *For every polynomial-size circuit family of distinguishers $\mathsf{A} = \{\mathsf{A}_\lambda\}_{\lambda \in \mathbb{N}}$, there is a negligible function μ, such that for all $\lambda \in \mathbb{N}$:*
$$\left| \Pr_{(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)} [1 \leftarrow \mathsf{A}_\lambda(pk)] - \Pr_{pk^* \leftarrow \mathsf{LGen}(1^\lambda)} [1 \leftarrow \mathsf{A}_\lambda(pk^*] \right| \leqslant \mu(\lambda).$$

Remark 6 (Amplifying hiding)**.** *Using standard amplification (XOR Lemma), δ can be made as small as $2^{-\mathsf{poly}(\lambda)}$ for any* poly *at the cost of polynomially increasing the size of commitments (c.f. [LM20]).*

Remark 7 (Perfectly-binding commitment from LPKE)**.** *With overwhelming probability over the choice of a random* real *key pk, the encryption algorithm $\mathsf{E}(pk, \cdot)$ acts as a perfectly-binding (non-interactive) commitment, with the random coins used for encryption serving as opening (see [LS19]).*

Multi-party computation (MPC). We adopt conventions and definitions for secure multiparty computation from [IKOS07].

Definition 4.2 (Multi-Party Computation (MPC) [IKOS07])**.** *For $n \in \mathbb{N}$, an* MPC *protocol is a protocol involving n parties $P_1, \ldots, P_n$ that takes place in $\rho = \rho(n)$ rounds of communication. The public input is denoted by x, while the private input and random coins of P_i are denoted by $w_i \in \{0,1\}^{\mathsf{poly}(|x|)}$ and $r_i \in \{0,1\}^{\mathsf{poly}(|x|)}$, respectively. The protocol is specified by its next-message function*

$$\{m_{i,j,k}\}_{j\in[n]} := M(i, x, w_i, r_i, (\{m_{j,i,1}\}_{j\in[n]}, \ldots, \{m_{j,i,k-1}\}_{j\in[n]})),$$

where, for $i \neq j \in [n]$ and $k \in [\rho]$, $m_{i,j,k}$ denotes the returns the message sent by P_i to P_j in round k. The view *of party P_i, denoted by $v_i = v_i(x, w_1, \ldots, w_n; r_1, \ldots r_n)$, consists of the private input and randomness, and the messages it receives over all rounds: i.e.,*

$$v_i := (w_i, r_i, \{m_{j,i,k}\}_{j\in[n],k\in[\rho]}). \tag{2}$$

Two views v_i and v_j are said to be consistent *if the outgoing messages implicit in v_i are identical to the incoming messages in v_j and vice versa.*

Definition 4.3 (t-Perfectly-Secure MPC in the Semi-Honest Model [IKOS07])**.** *For $t \leqslant n \in \mathbb{N}$, an* MPC *protocol M realises an n-party functionality $f = f(x, w_1, \ldots, w_n)$ with* t-perfect-security in the semi-honest model *if the following properties hold:*

1. M realises f with perfect correctness. *That is, for any input $(x, w_1, \ldots, w_n)$, the probability (over the choice of $r_1, \ldots, r_n$) that the output of some player is different from the value of f is* 0.
2. M realises f with perfect t-privacy. *That is, there is a PPT simulator S_M such that for any inputs $x, w_1, \ldots, w_n$ and any set of corrupted players $\mathcal{C} \subseteq [n]$ with $|\mathcal{C}| < t$, the distribution of joint views of players in $\mathcal{C}$, denoted $V_{\mathcal{C}} = V_{\mathcal{C}}(x, w_1, \ldots, w_n)$, is* identical *to $\mathsf{S}_M(\mathcal{C}, x, \{w_i\}_{i\in\mathcal{C}}, f_{\mathcal{C}}(x, w_1, \ldots, w_n))$. Here $f_{\mathcal{C}}(x, w_1, \ldots, w_n) := \{f_i(x, w_1, \ldots, w_n)\}_{i\in\mathcal{C}}$ and f_i denotes the i-th output of f.*

Remark 8. *An $n/2$-perfectly-secure semi-honest MPC protocol was constructed in [BGW88] (also see [AL17]), which is what we rely on in Theorem 4.4.*

4.2 Amplification Theorem

Our amplification protocol relies on two primitives: LPKE (Definition 4.1) and semi-honest and statistically private multi-party computation (MPC) (Definitions 4.2 and 4.3). It is described formally in Fig. 5 and the amplification theorem is stated in Theorem 4.4. The approach is similar in spirit to that in [GJS19] in the sense that the prover executes an MPC protocol "in its head" [IKOS07], commits to the view of each party in the execution and then proves consistency of each pair of views using the underlying proof system.

There are some key differences though:

1. We use LPKE instead of a commitment scheme. LPKE has two (indistinguishable) modes of operation: its real keys act as perfectly-binding commitments with overwhelming probability (see Remark 7), whereas its lossy keys act as statistically-hiding commitments. Since we use lossy keys in the protocol (see Fig. 5, Line 1), we are able to show amplification of privacy using statistical tools. In particular, we build on the approach from [LM20] based on statistical coupling [Ald83]. On the other hand, when arguing soundness we first switch the protocol to a real key and exploit the fact that it acts as a perfectly-binding commitment. Thus, we are able to avoid the argument based on (computational) hardcore lemmas [Imp95, Hol05], which [GJS19] rely on.
2. We commit to the views as a whole (as in [IKOS07]) instead of the fine-grained way of committing in [GJS19]. In more details, [GJS19] commit to the private inputs of parties, their private coins and each message in the transcript using separate commitments; we only commit to the view of each party as a whole.

3. Since we start from a NISZKA with a negligible soundness error, MPC correctness in our protocol is guaranteed and therefore semi-honest privacy suffices. This is in contrast to [GJS19] who rely on malicious security to deal also with a soundness error of the underlying proof system.

Theorem 4.4 (Amplification Theorem)**.** *Consider the protocol* $\Pi_{\mathsf{WI}} = \langle \mathsf{P}_{\mathsf{WI}} \to \mathsf{V}_{\mathsf{WI}} \rangle$ *described in Fig. 5 obtained by instantiating:*

1. M *using an* $n/2$*-perfectly-secure semi-honest MPC protocol.*
2. Π_{ZK} *using an adaptively-sound NISZKA with ZK error* $\varepsilon = 1/100n$*.*
3. Λ *using an LPKE with statistical hiding error* δ*.*

Then Π_{WI} *is NISWIA with non-adaptive soundness (in the CRS model) with following properties:*

- *If* Π_{ZK} *has negligible (resp.,* 0*) completeness error then so does* Π_{WI}*.*
- *The soundness error is negligible in the (computational) security parameter* λ *of the LPKE.*
- *The WI error is* $2^{-n+1} + 2^{n+2}\delta n$*. In particular, taking* $\delta \leqslant 2^{-2n}n$*, the WI error is at most* $O(2^{-n})$*.*
- *If* P_{ZK} *is non-uniform then so is* P_{WI}*.*

Proof. The fact that Π_{WI} is non-uniform if Π_{WI} follows by construction. We prove the rest of the properties in Propositions 4.5 to 4.7.

Proposition 4.5 (Completeness)**.** *If* Π_{ZK} *has completeness error* ε_c*, then* Π_{WI} *has completeness error at most* $(1-(1-\varepsilon_c)^{\binom{n}{2}}) \leqslant \varepsilon_c \binom{n}{2}$*.*

Proof Sketch. Completeness of Π_{WI} reduces to completeness of Π_{ZK} thanks to perfect correctness of M as we argue next. If $(x, w) \in \mathcal{R}$, then by correctness of M all pairs of views (v_i, v_j) are consistent and locally accepting. This implies that $((c_i, c_i), (v_i, q_i, v_j, q_j)) \in \mathcal{R}'_{i,j}$ for every execution (i, j).[12] Since V_{WI} accepts if V_{ZK} accepts all the underlying proofs, and since the CRSs of Π_{ZK} are sampled independently, the lemma follows. □

Proposition 4.6 (Soundness Preserved)**.** Π_{WI} *is non-adaptively sound (with a negligible soundness error).*

Proof Sketch. The proof proceeds in two stages.

- First, let us consider a modified setup algorithm $\mathsf{Gen}'_{\mathsf{WI}}$ where a *real* key pk sampled using KGen is used instead of a lossy key pk^* as in $\mathsf{Gen}_{\mathsf{WI}}$. Now, consider any polynomial-sized circuit family of malicious provers P^*. Since the real and lossy keys of Λ are computationally indistinguishable, the above switch is indistinguishable to P^*.[13] That is, there exists a negligible function μ such that for every $x \notin \mathcal{L}(\mathcal{R})$:

$$\left| \Pr_{\substack{(pk^*, \boldsymbol{crs}) \leftarrow \mathsf{Gen}_{\mathsf{WI}} \\ \boldsymbol{\pi}^* \leftarrow \mathsf{P}^*_n((pk^*, \boldsymbol{crs}), x)}} \left[\mathsf{V}_{\mathsf{WI}}((pk^*, \boldsymbol{crs}), x, \boldsymbol{\pi}^*) = 1\right] - \Pr_{\substack{(pk, \boldsymbol{crs}) \leftarrow \mathsf{Gen}'_{\mathsf{WI}} \\ \boldsymbol{\pi}^* \leftarrow \mathsf{P}^*_n((pk, \boldsymbol{crs}), x)}} \left[\mathsf{V}_{\mathsf{WI}}((pk, \boldsymbol{crs}), x, \boldsymbol{\pi}^*) = 1\right] \right| \leqslant \mu(n).$$

[12] Note that we don't rely on correctness of decryption of Λ here and only use the fact that the encryption algorithm is a map once the random coins are fixed.

[13] Note that this switch does not work when one tries to argue *adaptive* soundness. The cheating prover has the freedom to choose the instance and it could potentially pick $(x, w) \in \mathcal{R}$, use w to generate a honest proof $\boldsymbol{\pi}$ for x and pass it off as a break of soundness. Even though soundness is not really broken here, this switch is hard to test (unless $\mathcal{L}(\mathcal{R})$, the language corresponding to $\mathcal{R}$, is trivial). However, this is not an issue for arguing non-adaptive soundness, since $x \notin \mathcal{L}(\mathcal{R})$ is fixed in advance.

Given a base non-interactive protocol $\Pi_{\mathsf{ZK}} = \langle \mathsf{P}_{\mathsf{ZK}} \to \mathsf{V}_{\mathsf{ZK}} \rangle$, an LPKE $\Lambda = (\mathsf{KGen}, \mathsf{LGen}, \mathsf{E}, \mathsf{D})$ and an MPC protocol M, the non-interactive protocol $\Pi_{\mathsf{WI}} = \langle \mathsf{P}_{\mathsf{WI}} \to \mathsf{V}_{\mathsf{WI}} \rangle$ for $\mathcal{R} \cap \{0,1\}^n \times \{0,1\}^*$, where $\mathcal{R}$ is any **NP** relation, is described below.

$\underline{crs_{\mathsf{WI}} \leftarrow \mathsf{Gen}_{\mathsf{WI}}(1^n)}$

1. Run Λ's *lossy* key-generation algorithm to generate a lossy key $pk^* \leftarrow \mathsf{LGen}(1^\lambda)$
2. For $(i,j) \in \binom{[n]}{2}$, run the setup algorithm of Π_{ZK} to generate CRS: $crs_{i,j} \leftarrow \mathsf{Gen}_{\mathsf{ZK}}(1^n)$
3. Output $crs_{\mathsf{WI}} := (pk^*, crs_{1,2}, \ldots, crs_{n-1,n})$ as the CRS

$\underline{\boldsymbol{\pi} \leftarrow \mathsf{P}_{\mathsf{WI}}(crs_{\mathsf{WI}}, x, w)}$

1. Execute M "in the head", using pk^* to *commit* to the views:
   (a) Generate shares $w_1, \ldots, w_n$ of the witness w: sample $w_1, \ldots, w_{n-1} \leftarrow \{0,1\}^{|w|}$ and then sets
   $$w_n := w \oplus w_1 \oplus \ldots \oplus w_{n-1}.$$
   (b) Samples random coins $r_1, \ldots, r_n$ for the n parties $P_1, \ldots, P_n$.
   (c) Set x as the public input, w_i and r_i, respectively, as P_i's private input and random coins, and run M for the functionality
   $$f(x, w_1, \ldots, w_n) := \mathcal{R}(x, \oplus_{i=1}^n w_i).$$
   Let v_i denote P_i's view in the above execution (see Eq. (2)).
   (d) Commit to the views: for $i \in [n]$, sample random coins $q_i \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}$ and compute $c_i := \mathsf{E}(pk^*, v_i; q_i)$.
2. Prove pairwise consistency in parallel: for each $(i,j) \in \binom{[n]}{2}$, generate proof
   $$\pi_{i,j} \leftarrow \mathsf{P}_{\mathsf{ZK}}(crs_{i,j}, (c_i, c_j), (v_i, q_i, v_j, q_j))$$
   for the NP relation $\mathcal{R}'_{i,j} := \mathcal{R}'_{\mathrm{M},x,pk^*,i,j}$, where $((c_i, c_i), (v_i, q_i, v_j, q_j)) \in \mathcal{R}'_{i,j}$ if
   (a) c_i (respectively, c_j) is the encryption of v_i (respectively, v_j) under pk^* using random coins q_i (respectively q_j); and
   (b) v_i and v_j are consistent (with respect to public input x and protocol M) and locally accepting (i.e., the local outputs are 1).
3. Output $\boldsymbol{\pi} := ((c_1, \ldots, c_n), (\pi_{1,2}, \ldots, \pi_{n-1,n}))$

$\underline{0/1 := \mathsf{V}_{\mathsf{WI}}(crs_{\mathsf{WI}}, x, \boldsymbol{\pi})}$

1. Accept if and only if the underlying NISZKA verifier accepts all proofs. That is, for all $(i,j) \in \binom{[n]}{2}$:
   $$\mathsf{V}_{\mathsf{ZK}}(crs_{i,j}, (c_i, c_j), \pi_{i,j}) = 1.$$

Figure 5: Non-interactive protocol Π_{WI}.

- At this point, by correctness of decryption of Λ's real keys, the ciphertexts act as perfectly-binding commitments with overwhelming probability (see Remark 7), and we show how this allows exploiting any P^* that breaks soundness with respect to $\mathsf{Gen}'_{\mathsf{WI}}$ to break Π_{ZK}'s soundness. Suppose P^* successfully breaks soundness with respect to $\mathsf{Gen}'_{\mathsf{WI}}$ for some instance $x \notin \mathcal{L}(\mathcal{R})$ with non-negligible probability. We claim that there must exist at least one pair of views (v_i^*, v_j^*) that are *inconsistent* with respect to the public input, which is the instance x: otherwise, if all pairs of views are consistent with respect to x, then by perfect correctness of M it can be argued that $x \in \mathcal{L}(\mathcal{R})$ (see [IKOS07, Lemma 2.3] about local vs. global consistency). However, this means that $(c_i^*, c_j^*) \notin \mathcal{L}(\mathcal{R}'_{i,j})$ with overwhelming probability. Since V_{WI} accepts if and only if V_{ZK} accepts *all* the underlying proofs, $\pi^*_{i,j}$ breaks Π_{ZK}'s soundness.[14]

□

Before proving WI in Proposition 4.7, we establish some useful notation.

Notation 1 (Non-Standard String Notation)**.**

- *For a string or vector s of length n and a set $\mathcal{S} \subseteq [n]$, we use $s_{\mathcal{S}}$ to denote $\{s_i\}_{i \in \mathcal{S}}$.*
- *We let N denote $\binom{n}{2}$, and interpret a string $s \in \{0,1\}^N$ as $\{s_{i,j}\}_{(i,j) \in \binom{[n]}{2}}$.*
- *By $\{0,1\}^N_{\geqslant T}$, we denote subset of strings in $\{0,1\}^N$ with Hamming weight* at least *T: i.e. $\{0,1\}^N_{\geqslant T} := \{s \in \{0,1\}^N : \|s\|_0 \geqslant T\}$.*

Notation 2 (Hybrid Distributions)**.**

- *Recall from Section 2 that $\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle(x)$ denotes the random variable corresponding to V_{WI}'s views when the protocol in Fig. 5 is executed on $(x,w) \in \mathcal{R}$, i.e., $(pk^*, \boldsymbol{crs}, \boldsymbol{c}, \boldsymbol{\pi})$, where $(pk^*, \boldsymbol{crs}) \leftarrow \mathsf{Gen}_{\mathsf{WI}}(1^n)$ and $(\boldsymbol{c}, \boldsymbol{\pi}) \leftarrow \mathsf{P}_{\mathsf{WI}}((pk^*, \boldsymbol{crs}), x, w)$.*
- *Similarly, $\langle \mathsf{P}_{\mathsf{ZK}}((v_i, q_i, v_j, q_j)) \to \mathsf{V}_{\mathsf{ZK}} \rangle(c_i, c_j)$ denotes the random variable corresponding to V_{ZK}'s views when the protocol Π_{ZK} is executed on $((c_i, c_j), (v_i, q_i, v_j, q_j)) \in \mathcal{R}'_{i,j}$, i.e., $(crs_{i,j}, \pi_{i,j})$, where $crs_{i,j} \leftarrow \mathsf{Gen}_{\mathsf{ZK}}(1^n)$ and $\pi_{i,j} \leftarrow \mathsf{P}_{\mathsf{ZK}}(crs_{i,j}, (c_i, c_j), (v_i, q_i, v_j, q_j))$.*
- *For $s \in \{0,1\}^N$, we use $\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)$ to denote the hybrid distribution described in Fig. 6, where the views of V_{ZK} in executions (i,j) such that $s_{i,j} = 1$ are simulated using S_{ZK} (thus $\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_{0^N}(x)$ corresponds to the real view). For a distribution S over $\{0,1\}^N$, $\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_S(x)$ is defined as in Fig. 6, with s first sampled according to S.*

Proposition 4.7 (Privacy Amplified)**.** *Π_{WI} is HVSWI with an error $2^{-n+1} + 2^{n+2}\delta n$.*

Proof. The proof proceeds in two steps. We first prove in Claim 4.8 that Π_{WI} is a combiner; that is, if a large enough fraction of the NISZK proofs are perfect ZK, then the resulting protocol is WI (with a negligible WI error). Then, taking a common approach in the literature, we prove in Claim 4.9 that any such combiner is also a good amplifier; that is, provided that every NSIZK has a small enough ZK error ε, the resulting protocol is WI (with a negligible WI error related to that of the corresponding combiner). Specifically, the proof of the latter claim follows the ideas developed in [LM20].

Claim 4.8 (Π_{WI} is a Threshold Combiner)**.** *For $T := N - n/4 + 1$ and any $s \in \{0,1\}^N_{\geqslant T}$, $(x,w), (x,w') \in \mathcal{R}$,*

$$\mathsf{SD}(\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)), \langle \mathsf{P}_{\mathsf{WI}}(w') \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)) \leqslant 2\delta n.$$

[14] This is why we require Π_{ZK} to be adaptively sound to start off with: the instance (c_i^*, c_j^*) and the associated proof $\pi^*_{i,j}$ that breaks Π_{ZK}'s soundness are determined by the output of the cheating prover P^*.

Let $n := |x|$ and $N := \binom{[n]}{2}$. For $s \in \{0,1\}^N$, the distribution $H_1 = H_{1,s}$ is defined below. For a distribution S over $\{0,1\}^N$, the hybrid distribution $\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_S(x)$ is defined as $H_{1,s}$ with s first sampled according to S.

$\underline{(pk^*, \boldsymbol{crs}, \boldsymbol{c}, \boldsymbol{\pi}) \leftarrow H_{1,s}(x,w)}$

1. Sample lossy key $pk^* \leftarrow \mathsf{LGen}(1^n)$
2. Generate the commitment $\boldsymbol{c}$ as specified in Fig. 5, Line 1. That is:
   (a) Run MPC as in Fig. 5, Lines 1a to 1c to generate views $(v_1, \ldots, v_n)$.
   (b) Compute the n-tuple of commitments $\boldsymbol{c}$, where $c_i := \mathsf{E}(pk^*, v_i; q_i)$, as in Fig. 5, Line 1d.
3. Generate V_{ZK}'s views in Fig. 5, Line 2 depending on s. That is, sample $(\boldsymbol{crs}, \boldsymbol{\pi})$, where for $(i,j) \in \binom{[n]}{2}$
$$(crs_{i,j}, \pi_{i,j}) \leftarrow \begin{cases} \langle \mathsf{P}_{\mathsf{ZK}}((v_i, q_i, v_j, q_j)) \to \mathsf{V}_{\mathsf{ZK}} \rangle(c_i, c_j) & \text{if } s_{i,j} = 0 \\ \mathsf{S}_{\mathsf{ZK}}(c_i, c_j) & \text{otherwise.} \end{cases}$$
4. Output $(pk^*, \boldsymbol{crs}, \boldsymbol{c}, \boldsymbol{\pi})$.

Figure 6: Hybrid distribution $H_1 = H_{1,s}$.

Claim 4.9 (Amplification from Threshold Combiners)**.** *For $T := N - n/4 + 1$ and any $s \in \{0,1\}^N_{\geqslant T}$, $(x,w), (x,w') \in \mathcal{R}$,*

$$\mathsf{SD}(\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle(x), \langle \mathsf{P}_{\mathsf{WI}}(w') \to \mathsf{V}_{\mathsf{WI}} \rangle(x)) \leqslant$$
$$2^{-n+1} + 2^{n+1} \cdot \max_{s \in \{0,1\}^N_{\geqslant T}} \mathsf{SD}(\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x), \langle \mathsf{P}_{\mathsf{WI}}(w') \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)) \ .$$

Proof of Claim 4.8. We proceed via a hybrid argument, and let $H_1 = H_{1,s}$ denote the distribution $\langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)$ from Fig. 6. Since $\|s\|_0 \geqslant T = N - n/4 + 1$ and as each proof depends on at most two parties, there exists a set $\mathcal{H} \subset [n]$ determined by s of size at least $n/2$ such that $s_{i,j} = 1$ holds for every $i \in \mathcal{H}$ and $j \in [n] \backslash \{i\}$. We think of these as the *honest* parties of the MPC protocol.

- In hybrid $H_2 = H_{2,s}$, we switch the messages underlying the ciphertexts $c_{\mathcal{H}}$ from honestly-generated views $v_{\mathcal{H}}$ to a dummy message independent of the witness w: see Fig. 7. To see why $\mathsf{SD}(H_1, H_2) \leqslant \delta n$ (for any s), fix any $i \in \mathcal{H}$. Since the view $\langle \mathsf{P}_{\mathsf{ZK}}((v_i, q_i, v_j, q_j)) \to \mathsf{V}_{\mathsf{ZK}} \rangle(c_i, c_j)$ in every execution (i,j), $j \in [n] \backslash \{i\}$, is simulated, it follows that the random coins q_i of the ciphertext c_i (which serve as part of witness for Π_{ZK}) are no longer required for generating proofs. Therefore, it is possible to use δ-statistical-hiding of Λ switch all ciphertexts in $\mathcal{H}$ (of which there are at most n of).[15]
- In the next hybrid $H_3 = H_{3,s}$, we simulate the joint views $v_{\overline{\mathcal{H}}}$ of the remaining parties using the MPC simulator $\mathsf{S}_{\mathsf{MPC}}$: see Fig. 8. The ciphertexts and proofs that depend on $v_{\overline{\mathcal{H}}}$ are generated accordingly. Note that H_3 is distributed identical to H_2 thanks to $n/2$-privacy of M.

We get that $\mathsf{SD}(H_1, H_3) \leqslant \delta n$. By a symmetric argument to above it is possible to show that $\mathsf{SD}(H_3, \langle \mathsf{P}_{\mathsf{WI}}(w') \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)) \leqslant \delta n$. The claim now follows by another application of the triangle inequality. □

[15] Note that even though the random coins used to sample the keys are revealed, we have hiding as long as the keys are sampled correctly.

Let $n := |x|$ and $N := \binom{[n]}{2}$. For $s \in \{0,1\}^N$, the distribution $H_2 = H_{2,s}$ is defined below.

$\underline{(pk^*, \boldsymbol{crs}, \boldsymbol{c}, \boldsymbol{\pi}) \leftarrow H_{2,s}(x, w)}$

1. Sample lossy key $pk^* \leftarrow \mathsf{LGen}(1^n)$
2. Generate the commitment $\boldsymbol{c}$ with dummy messages for the parties in $\mathcal{H}$. That is:
   (a) Run MPC as in Fig. 5, Lines 1a to 1c to generate views $(v_1, \ldots, v_n)$.
   (b) Compute the n-tuple of commitments $\boldsymbol{c}$ depending on h, where $h \in \{0,1\}^n$ denotes the indicator string for $\mathcal{H}$ determined by s:

$$c_i \leftarrow \begin{cases} \mathsf{E}(pk^*, v_i; q_i) \text{ for } q_i \leftarrow \{0,1\}^{\mathsf{poly}(|x|)} & \text{if } h_i = 0 \\ \mathsf{E}(pk^*, 0^{|v_i|}) & \text{otherwise} \end{cases}$$

3. Generate V_{ZK}'s views in Fig. 5, Line 2 depending on s. That is, sample $(\boldsymbol{crs}, \boldsymbol{\pi})$, where for $(i,j) \in \binom{[n]}{2}$

$$(crs_{i,j}, \pi_{i,j}) \leftarrow \begin{cases} \langle \mathsf{P}_{\mathsf{ZK}}((v_i, q_i, v_j, q_j)) \rightarrow \mathsf{V}_{\mathsf{ZK}} \rangle (c_i, c_j) & \text{if } s_{i,j} = 0 \\ \mathsf{S}_{\mathsf{ZK}}(c_i, c_j) & \text{otherwise} \end{cases}$$

4. Output $(pk^*, \boldsymbol{crs}, \boldsymbol{c}, \boldsymbol{\pi})$

Figure 7: Hybrid distribution $H_2 = H_{2,s}$.

Let $n := |x|$ and $N := \binom{[n]}{2}$. For $s \in \{0,1\}^N$, the distribution $H_2 = H_{2,s}$ is defined below.

$\underline{(pk^*, \boldsymbol{crs}, \boldsymbol{c}, \boldsymbol{\pi}) \leftarrow H_{3,s}(x, w)}$

1. Sample lossy key $pk^* \leftarrow \mathsf{LGen}(1^n)$
2. Generate the commitment $\boldsymbol{c}$ with dummy messages for the parties in $\mathcal{H}$ and simulated views for parties in $\overline{\mathcal{H}}$. That is:
   (a) Simulate the MPC (joint) views $v_{\overline{\mathcal{H}}} := \mathsf{S}_{\mathsf{MPC}}(\overline{\mathcal{H}}, x, \{w_i\}_{i \in \overline{\mathcal{H}}}, (1, \ldots, 1))$
   (b) Compute the n-tuple of commitments $\boldsymbol{c}$ depending on h, where $h \in \{0,1\}^n$ denotes the indicator string for $\mathcal{H}$ determined by s:

$$c_i \leftarrow \begin{cases} \mathsf{E}(pk^*, v_i; q_i) \text{ for } q_i \leftarrow \{0,1\}^{\mathsf{poly}(|x|)} & \text{if } h_i = 0 \\ \mathsf{E}(pk^*, 0^{|v_i|}) & \text{otherwise} \end{cases}$$

3. Generate V_{ZK}'s views in Fig. 5, Line 2 depending on s. That is, sample $(\boldsymbol{crs}, \boldsymbol{\pi})$, where for $(i,j) \in \binom{[n]}{2}$

$$(crs_{i,j}, \pi_{i,j}) \leftarrow \begin{cases} \langle \mathsf{P}_{\mathsf{ZK}}((v_i, q_i, v_j, q_j)) \rightarrow \mathsf{V}_{\mathsf{ZK}} \rangle (c_i, c_j) & \text{if } s_{i,j} = 0 \\ \mathsf{S}_{\mathsf{ZK}}(c_i, c_j) & \text{otherwise} \end{cases}$$

4. Output $(pk^*, \boldsymbol{crs}, \boldsymbol{c}, \boldsymbol{\pi})$

Figure 8: Hybrid distribution $H_3 = H_{3,s}$.

Proof of Claim 4.9. For $s \in \{0,1\}^N$ and distribution S over $\{0,1\}^N$, recall the distributions $H_{1,s} = \langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)$ and $H_{1,S} = \langle \mathsf{P}_{\mathsf{WI}}(w) \to \mathsf{V}_{\mathsf{WI}} \rangle_S(x)$ defined in Fig. 6. Similarly, let $H'_{1,s}$ and $H'_{1,S}$ denote $\langle \mathsf{P}_{\mathsf{WI}}(w') \to \mathsf{V}_{\mathsf{WI}} \rangle_s(x)$ and $\langle \mathsf{P}_{\mathsf{WI}}(w') \to \mathsf{V}_{\mathsf{WI}} \rangle_S(x)$, respectively. Recall that our goal is to show that

$$\mathsf{SD}(H_{1,0^N}, H'_{1,0^N}) \leqslant 2^{-n+1} + 2^{n+1} \cdot \max_{s \in \{0,1\}^N_{\geqslant T}} \mathsf{SD}(H_{1,s}, H'_{1,s}) \ .$$

First, for any distributions Z over $\{0,1\}^N_{\geqslant T} \cup \{0^N\}$, we have

$$\begin{aligned}
\mathsf{SD}(H_{1,Z}, H'_{1,Z}) &= \frac{1}{2} \sum_h \left| H_{1,Z}(h) - H'_{1,Z}(h) \right| \\
&= \frac{1}{2} \sum_h \left| \sum_{z \in \{0,1\}^N_{\geqslant T} \cup \{0^N\}} Z(z)(H_{1,z}(h) - H'_{1,z}(h)) \right| \\
&\geqslant \frac{1}{2} \sum_h \left(Z(0^N) \left| H_{1,0^N}(h) - H'_{1,0^N}(h) \right| - \sum_{z \in \{0,1\}^N_{\geqslant T}} Z(z) \left| H_{1,z}(h) - H'_{1,z}(h) \right| \right) \\
&= Z(0^N) \cdot \mathsf{SD}(H_{1,0^N}, H'_{1,0^N}) - \sum_{z \in \{0,1\}^N_{\geqslant T}} Z(z) \cdot \mathsf{SD}(H_{1,z}, H'_{1,z}) \\
&\geqslant Z(0^N) \cdot \mathsf{SD}(H_{1,0^N}, H'_{1,0^N}) - (1 - Z(0^N)) \cdot \max_{z \in \{0,1\}^N_{\geqslant T}} \mathsf{SD}(H_{1,z}, H'_{1,z}).
\end{aligned}$$

Thus, for any distribution Z over $\{0,1\}^N_{\geqslant T} \cup \{0^N\}$ with $Z(0^N) > 0$, and distribution S over $\{0,1\}^N_{\geqslant T}$,

$$\begin{aligned}
\mathsf{SD}(H_{1,0^N}, H'_{1,0^N}) &\leqslant Z(0^N)^{-1} \cdot \mathsf{SD}(H_{1,Z}, H'_{1,Z}) + \max_{z \in \{0,1\}^N_{\geqslant T}} \mathsf{SD}(H_{1,z}, H'_{1,z}) \\
&\leqslant Z(0^N)^{-1} \cdot (\mathsf{SD}(H_{1,Z}, H_{1,S}) + \mathsf{SD}(H_{1,S}, H'_{1,S}) + \mathsf{SD}(H'_{1,S}, H'_{1,Z})) + \max_{z \in \{0,1\}^N_{\geqslant T}} \mathsf{SD}(H_{1,z}, H'_{1,z}) \\
&\leqslant Z(0^N)^{-1} \cdot (\mathsf{SD}(H_{1,Z}, H_{1,S}) + \mathsf{SD}(H'_{1,Z}, H'_{1,S}) + 2Z(0^N)^{-1} \cdot \max_{z \in \{0,1\}^N_{\geqslant T}} \mathsf{SD}(H_{1,z}, H'_{1,z}) \ .
\end{aligned}$$

To complete the proof, we prove the following Lemma.

Lemma 4.10. *There exist two distributions Z and S, where Z is over $\{0,1\}^N_{\geqslant T} \cup \{0^N\}$, with $Z(0^N) > 2^{-n}$, and S is over $\{0,1\}^N_{\geqslant T}$, such that*

$$Z(0^N)^{-1} \cdot \max\left\{\mathsf{SD}(H_{1,Z}, H_{1,S}), \mathsf{SD}(H'_{1,Z}, H'_{1,S})\right\} \leqslant \left(\frac{4eN}{n}\varepsilon\right)^{n/4}. \tag{3}$$

Indeed, for our setting of parameters, i.e., $N = \binom{n}{2}$ and $\varepsilon = 1/100n$, the value of Eq. (3) is at most $1/2^n$.

The proof of the Lemma is based on a coupling argument and roughly follows [LM20]. The proof can be found in Appendix A.

□

This completes the proof of Proposition 4.7. □

□

Acknowledgements

We thank Oded Goldreich for valuable comments and discussion.

Nir Bitansky is a member of the Checkpoint Institute of Information Security and is supported by the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No. 101042417, acronym SPP).

Chethan Kamath is supported by Azrieli International Postdoctoral Fellowship, by the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No. 101042417, acronym SPP), and by ISF grant 1789/19.

Omer Paneth is a member of the Checkpoint Institute of Information Security and is supported by an Azrieli Faculty Fellowship, and ISF grant 1789/19.

Ron Rothblum is funded by the European Union (ERC, FASTPROOF, 101041208). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Prashant Nalini Vasudevan is supported by the National Research Foundation, Singapore, under its NRF Fellowship programme, award no. NRF-NRFF14-2022-0010.

References

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991.

[AL17] Gilad Asharov and Yehuda Lindell. A full proof of the BGW protocol for perfectly secure multiparty computation. *Journal of Cryptology*, 30(1):58–151, January 2017.

[Ald83] David Aldous. Random walks on finite groups and rapidly mixing Markov chains. In *Séminaire de Probabilités XVII 1981/82: Proceedings*, pages 243–297. Springer, 1983.

[AR23] Noga Amit and Guy N. Rothblum. Constant-round arguments from one-way functions. To appear in STOC 2023, 2023.

[BBD+20] Marshall Ball, Elette Boyle, Akshay Degwekar, Apoorvaa Deshpande, Alon Rosen, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. Cryptography from information loss. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 81:1–81:27. LIPIcs, January 2020.

[BDRV18] Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 133–161. Springer, 2018.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988.

[BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 474–482. ACM, 2017.

[BHKY19] Nir Bitansky, Iftach Haitner, Ilan Komargodski, and Eylon Yogev. Distributional collision resistance beyond one-way functions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 667–695. Springer, 2019.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 671–684. ACM, 2018.

[Blu81] Manuel Blum. Coin flipping by telephone. In Allen Gersho, editor, *CRYPTO'81*, volume ECE Report 82-04, pages 11–15. U.C. Santa Barbara, Dept. of Elec. and Computer Eng., 1981.

[BMO90] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. The (true) complexity of statistical zero knowledge. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 494–502. ACM, 1990.

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 394–423, Virtual Event, August 2021. Springer, Heidelberg.

[CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd FOCS*, pages 68–79. IEEE Computer Society Press, February 2022.

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[Del16] Holger Dell. And-compression of NP-complete problems: Streamlined proof and minor observations. *Algorithmica*, 75(2):403–423, 2016.

[DMO00] Giovanni Di Crescenzo, Tal Malkin, and Rafail Ostrovsky. Single database private information retrieval implies oblivious transfer. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 122–138. Springer, Heidelberg, May 2000.

[DPP97] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *J. Cryptol.*, 10(3):163–194, 1997.

[Dru15] Andrew Drucker. New limits to classical and quantum instance compression. *SIAM J. Comput.*, 44(5):1443–1479, 2015.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 308–317. IEEE Computer Society, 1990.

[For89] Lance Fortnow. The complexity of perfect zero-knowledge. *Adv. Comput. Res.*, 5:327–343, 1989.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 416–426. ACM, 1990.

[FS08] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 133–142. ACM Press, May 2008.

[GH98] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Inf. Process. Lett.*, 67(4):205–214, 1998.

[GJS19] Vipul Goyal, Aayush Jain, and Amit Sahai. Simultaneous amplification: The case of non-interactive zero-knowledge. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 608–637. Springer, Heidelberg, August 2019.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *J. Cryptol.*, 9(3):167–190, 1996.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 174–187. IEEE Computer Society, 1986.

[Gol18] Oded Goldreich. On doubly-efficient interactive proof systems. *Found. Trends Theor. Comput. Sci.*, 13(3):158–246, 2018.

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Adv. Comput. Res.*, 5:73–90, 1989.

[GSV98] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 399–408. ACM, 1998.

[GVW02] Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Comput. Complex.*, 11(1-2):1–53, 2002.

[HHRS15] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - tight lower bounds on the round and communication complexities of statistically hiding commitments. *SIAM J. Comput.*, 44(1):193–242, 2015.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. SNARGs for P from sub-exponential DDH and QR. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 520–549. Springer, Heidelberg, May / June 2022.

[HM96] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 201–215. Springer, Heidelberg, August 1996.

[HN10] Danny Harnik and Moni Naor. On the compressibility of *NP* instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

[HNO+09] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 39(3):1153–1218, 2009.

[Hol05] Thomas Holenstein. Key agreement from weak bit agreement. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 664–673. ACM Press, May 2005.

[HRV18] Pavel Hubácek, Alon Rosen, and Margarita Vald. An efficiency-preserving transformation from honest-verifier statistical zero-knowledge to statistical zero-knowledge. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 66–87. Springer, 2018.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th FOCS*, pages 538–545. IEEE Computer Society Press, October 1995.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 723–732. ACM, 1992.

[KLVW22] Yael Tauman Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. *IACR Cryptol. ePrint Arch.*, page 1320, 2022.

[KMY20] Fuyuki Kitagawa, Takahiro Matsuda, and Takashi Yamakawa. NIZK from SNARG. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 567–595. Springer, Heidelberg, November 2020.

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev. Collision resistant hashing for paranoids: Dealing with multiple collisions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 162–194. Springer, 2018.

[KRR+20] Inbar Kaslasi, Guy N. Rothblum, Ron D. Rothblum, Adam Sealfon, and Prashant Nalini Vasudevan. Batch verification for statistical zero knowledge proofs. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 139–167. Springer, 2020.

[KRV21] Inbar Kaslasi, Ron D. Rothblum, and Prashant Nalini Vasudevan. Public-coin statistical zero-knowledge batch verification against malicious verifiers. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III*, volume 12698 of *Lecture Notes in Computer Science*, pages 219–246. Springer, 2021.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[LM20] David Lanzenberger and Ueli Maurer. Coupling of random systems. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 207–240. Springer, Heidelberg, November 2020.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Report 2019/279, 2019. https://eprint.iacr.org/2019/279.

[LY94] Richard J. Lipton and Neal E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *26th ACM STOC*, pages 734–740. ACM Press, May 1994.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptol.*, 4(2):151–158, 1991.

[Oka96] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 649–658. ACM, 1996.

[OVY93] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Interactive hashing simplifies zero-knowledge protocol design. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 267–273. Springer, 1993.

[PP22] Omer Paneth and Rafael Pass. Incrementally verifiable computation via rate-1 batch arguments. In *63rd FOCS*, pages 1045–1056. IEEE Computer Society Press, October / November 2022.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, Heidelberg, August 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008.

[RR20] Guy N. Rothblum and Ron D. Rothblum. Batch verification and proofs of proximity with polylog overhead. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 108–138. Springer, 2020.

[RRR18] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Efficient batch verification for UP. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 22:1–22:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[RRR21] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM J. Comput.*, 50(3), 2021.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[Vad99] Salil Pravin Vadhan. *A study of statistical zero-knowledge proofs*. PhD thesis, Massachusetts Institute of Technology, 1999.

[vN28] J v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[VZ13] Salil P. Vadhan and Colin Jia Zheng. A uniform min-max theorem with applications in cryptography. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 93–110. Springer, Heidelberg, August 2013.

[Wee05] Hoeteck Wee. On round-efficient argument systems. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, volume 3580 of *Lecture Notes in Computer Science*, pages 140–152. Springer, 2005.

[WW22] Brent Waters and David J. Wu. Batch arguments for NP and more from standard bilinear group assumptions. In *CRYPTO (2)*, volume 13508 of *Lecture Notes in Computer Science*, pages 433–463. Springer, 2022.

A Proof of Lemma 4.10

Before stating and (re)proving Lemma 4.10, which completes the proof of Claim 4.9 and therefore Theorem 4.4, we define statistical coupling (Definition A.1) and recall a lemma about statistical coupling (Lemma A.2) that will be key to the proof. We also introduce some notation (Notation 3) that will help reduce clutter.

Definition A.1 (Statistical Coupling)**.** *Let X and Y be two probability distributions defined on a finite set Ω. A joint probability distribution XY on Ω^2 is a* statistical coupling *of X and Y if its marginal distributions are X and Y, respectively, i.e., for every $x \in \Omega$:*

$$X(x) = \sum_{y \in \Omega} XY(x, y),$$

and for every $y \in \Omega$:

$$Y(y) = \sum_{x \in \Omega} XY(x, y).$$

Lemma A.2 (Coupling Lemma [Ald83])**.** *Let X and Y be probability distributions over the same set Ω. Then*

1. *For* every *coupling XY of X and Y,*

$$\mathsf{SD}(X, Y) \leqslant \Pr_{(x,y) \leftarrow XY}[x \neq y]$$

2. *There* exists *an "optimal" coupling XY^* such that*

$$\mathsf{SD}(X, Y) = \Pr_{(x,y) \leftarrow XY^*}[x \neq y]$$

Notation 3 ([LM20])**.** *For two vectors of objects (e.g., distributions) $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$, and a string $s \in \{0,1\}^n$, we use $\langle \boldsymbol{a}/\boldsymbol{b} \rangle_s = \langle a_1/b_1, a_2/b_2, \ldots, a_n/b_n \rangle_s$ to denote the vector $\boldsymbol{c} = (c_1, \ldots, c_n)$ where*

$$c_i := \begin{cases} a_i & \text{if } s_i = 0 \\ b_i & \text{if } s_1 = 1. \end{cases}$$

The following lemma is a restatement of Lemma 4.10 and is based on Lemma 7, Theorem 3 and Corollary 1 from [LM20]. The notation and presentation has been altered for the sake of compatibility with this paper.

Lemma A.3 ([LM20])**.** *There exists two distributions Z and S, where Z is over $\{0,1\}^N_{\geqslant T} \cup \{0^N\}$, with $Z(0^N) > 2^{-n}$, and S is over $\{0,1\}^N_{\geqslant T}$, such that*

$$Z(0^N)^{-1} \cdot \max\left\{\mathsf{SD}(H_{1,Z}, H_{1,S}), \mathsf{SD}(H'_{1,Z}, H'_{1,S})\right\} \leqslant \left(\frac{4eN}{n}\varepsilon\right)^{n/4}, \tag{4}$$

where N, n, T, ε, $H_{1,Z}$, $H_{1,S}$, $H'_{1,Z}$ and $H'_{1,S}$ are as defined in Section 4.

Proof. We proceed in two steps. First, we define what it means for a pair of distributions Z and S, as in the statement of the lemma, to be "good". Then, we show that if Z and S are good then the lemma follows (this roughly corresponds to [LM20, Lemma 7]).

Step I. For two strings $s, e \in \{0,1\}^N$, let $\mathsf{blind}(s,e)$ be the function that returns the substring of s at indices i such that $e_i = 1$.

Definition A.4 (Good pair of distributions)**.** *Let $E = (E_{i,j})_{(i,j)\in\binom{[n]}{2}}$ denote a tuple of N independent Bernoulli distributions with bias at most ε, i.e., for every $(i,j) \in \binom{[n]}{2}$, it holds that $\Pr_{e_{i,j}\leftarrow E_{i,j}}[e_{i,j} = 1] \leqslant \varepsilon$. A pair of distributions Z and S is* good *if*

- *Z is supported on $\{0,1\}^N_{\geqslant T} \cup \{0^N\}$,*
- *$Z(0^N) > 2^{-n}$,*
- *S is supported on $\{0,1\}^N_{\geqslant T}$,*
- *$\mathbb{E}_{e\leftarrow E}\left[\mathsf{SD}(\mathsf{blind}(Z,e), \mathsf{blind}(S,e))\right] \leqslant Z(0^N)\left(\frac{4eN}{n}\varepsilon\right)^{n/4}$.*

In [LM20], a pair of such good distributions is constructed explicitly. For the sake of completeness, we include a description of the [LM20] distributions in Appendix A.1.

Step II. We now prove the lemma given that the pair of distributions Z and S described in Stage I is good. To be specific, we show

$$\mathsf{SD}(H_{1,Z}, H_{1,S}) \leqslant \mathop{\mathbb{E}}_{e\leftarrow E}\left[\mathsf{SD}(\mathsf{blind}(Z,e), \mathsf{blind}(S,e))\right]. \tag{5}$$

The proof of the corresponding claim for H' is similar and is hence omitted.

Recall the distributions $H_{1,Z}$ and $H_{1,S}$ from Fig. 6. Next, consider $H_{1,Z}$ and $H_{1,S}$ with the execution of M and Λ fixed, i.e., the lossy key pk^*, views $\boldsymbol{v}$ and ciphertext-random coin pair $(\boldsymbol{c}, \boldsymbol{q})$ in distribution $H_{1,Z}$ and $H_{1,S}$ are fixed. To prove the lemma, it suffices to show Eq. (5) holds for every $(pk^*, \boldsymbol{v}, \boldsymbol{c}, \boldsymbol{q})$. Hence, from here on, let's consider $H_{1,Z}$ and $H_{1,S}$ with $(pk^*, \boldsymbol{v}, \boldsymbol{c}, \boldsymbol{q})$ fixed.

For $(i,j) \in \binom{[n]}{2}$, let's denote by $R_{i,j}$ and $I_{i,j}$ the random variables corresponding to the real and simulated execution of Π_{ZK}, respectively (see Fig. 6, Line 3). Following [LM20], we denote $H_{1,Z}$ by $H_1(\langle \boldsymbol{R}, \boldsymbol{I}\rangle_Z)$ (see Notation 3). Since Π_{ZK} is ZK with error at most $\varepsilon = 1/100n$, we have $\mathsf{SD}(R_{i,j}, I_{i,j}) \leqslant \varepsilon$. As a result, by Lemma A.2, there exists an optimal coupling RI^*_{ij} of $R_{i,j}$ and $I_{i,j}$ such that

$$\Pr_{((crs_{R,i,j},\pi_{R,i,j}),(crs_{I,i,j},\pi_{I,i,j}))\leftarrow RI^*_{i,j}}\left[(crs_{R,i,j},\pi_{R,i,j}) \neq (crs_{I,i,j},\pi_{I,i,j})\right] \leqslant \varepsilon. \tag{6}$$

Let's use $R^*_{i,j}$ and $I^*_{i,j}$ to denote the first and second argument of $RI^*_{i,j}$, respectively. Then, we have

$$\begin{aligned}
\mathsf{SD}(H_{1,Z}, H_{1,S}) &= \mathsf{SD}(H_1(\langle \boldsymbol{R}, \boldsymbol{I}\rangle_Z), H_1(\langle \boldsymbol{R}, \boldsymbol{I}\rangle_S)) \\
&= \mathsf{SD}(H_1(\langle \boldsymbol{R}^*, \boldsymbol{I}^*\rangle_Z), H_1(\langle \boldsymbol{R}^*, \boldsymbol{I}^*\rangle_S)) \\
&\leqslant \mathsf{SD}(\langle \boldsymbol{R}^*, \boldsymbol{I}^*\rangle_Z, \langle \boldsymbol{R}^*, \boldsymbol{I}^*\rangle_S).
\end{aligned} \tag{7}$$

Here, the second equality follows by the definition of coupling (which requires the marginals to match) and the inequality is a consequence of data processing inequality. To upper bound Eq. (7), we set up a coupling experiment G, described in Fig. 9, involving $\boldsymbol{RI}^*$. To see why G is a valid coupling, we claim that the marginal distributions of z and s sampled as part of G are Z and S respectively, z is independent of $\boldsymbol{R}$, and s is independent of $\boldsymbol{I}$. As a result, the marginal distributions of $\boldsymbol{\zeta}$ and $\boldsymbol{\sigma}$ are the same as $H_{1,Z}$ and $H_{1,S}$ respectively. To see why the (marginal) distribution of z sampled as part of G is Z (the argument for s and S is analogous) and independent of $\boldsymbol{R}$, note that for any e sampled in the first step, z' is distributed as $\mathsf{blind}(Z,e)$, and z is sampled from Z conditioned on z'.

$\underline{(\boldsymbol{\zeta}, \boldsymbol{\sigma}) \leftarrow G_{\boldsymbol{RI}^*}}$

1. Sample $((\boldsymbol{crs}_R, \boldsymbol{\pi}_R), (\boldsymbol{crs}_I, \boldsymbol{\pi}_I)) \leftarrow \boldsymbol{RI}^*$
2. Compute $e = \{e_{i,j}\}_{(i,j)\in\binom{[n]}{2}}$, where $e_{i,j}$ is indicator for the event $R^*_{i,j} = I^*_{i,j}$: i.e., $e_{i,j} = 1 \Leftrightarrow R^*_{i,j} = I^*_{i,j}$, where recall that $R^*_{i,j} = (crs_{R,i,j}, \pi_{R,i,j})$ and $I^*_{i,j} = (crs_{I,i,j}, \pi_{I,i,j})$.
3. Consider the random variables $\mathrm{blind}(Z, e)$ and $\mathrm{blind}(S, e)$ induced by Z and S, and let $\mathrm{blind}(Z, e)\mathrm{blind}(Z, e)^*$ denote the optimal coupling between the two.
4. Sample $(z', s') \leftarrow \mathrm{blind}(Z, e)\mathrm{blind}(S, e)^*$.
5. Sample $z \leftarrow Z$ conditioned on $\mathrm{blind}(z, e) = z'$ and $s \leftarrow S$ conditioned on $\mathrm{blind}(s, e) = s'$
6. Set $\boldsymbol{\zeta} := (\langle \boldsymbol{crs}_R / \boldsymbol{crs}_I \rangle_z, \langle \boldsymbol{\pi}_R / \boldsymbol{\pi}_I \rangle_z)$ and $\boldsymbol{\sigma} := (\langle \boldsymbol{crs}_R / \boldsymbol{crs}_I \rangle_s, \langle \boldsymbol{\pi}_R / \boldsymbol{\pi}_I \rangle_s)$
7. Output $(\boldsymbol{\zeta}, \boldsymbol{\sigma})$

Figure 9: Coupling experiment G.

Therefore, by Lemma A.2 ("for every" claim), we have from Eq. (7) that

$$\begin{aligned}
\mathsf{SD}(\langle \boldsymbol{R}^*, \boldsymbol{I}^* \rangle_Z, \langle \boldsymbol{R}^*, \boldsymbol{I}^* \rangle_S) &\leqslant \Pr_G(\boldsymbol{\zeta} \neq \boldsymbol{\sigma}) \\
&= \sum_{e' \in \binom{[n]}{2}} \Pr_G(\boldsymbol{\zeta} \neq \boldsymbol{\sigma}, e = e') \\
&= \sum_{e' \in \binom{[n]}{2}} \Pr_G(\mathrm{blind}(z, e) \neq \mathrm{blind}(s, e), e = e') \\
&= \sum_{e' \in \binom{[n]}{2}} \Pr_{G|_{e=e'}}[\mathrm{blind}(z, e') \neq \mathrm{blind}(s, e')] \cdot \Pr_G(e = e') \\
&= \sum_{e' \in \binom{[n]}{2}} \mathsf{SD}(\mathrm{blind}(Z, e'), \mathrm{blind}(S, e')) \cdot \Pr_G(e = e') \qquad (8)
\end{aligned}$$

where $\boldsymbol{\zeta}$, $\boldsymbol{\sigma}$, e and (z, s) above are sampled as part of G, and Eq. (8) follows by optimality of $\mathrm{blind}(Z, e)\mathrm{blind}(S, e)^*$. Since each $e_{i,j}$ in G is distributed as required (because Π_{ZK} is zero-knowledge with error ε and the executions are independent), we get from Eqs. (7) and (8) that

$$Z(0^N)^{-1} \cdot \mathsf{SD}(H_{1,Z}, H_{1,S}) \leqslant Z(0^N)^{-1} \cdot \mathop{\mathbb{E}}_{e' \leftarrow E}\left[\mathsf{SD}(\mathrm{blind}(Z, e'), \mathrm{blind}(S, e))\right] \leqslant \left(\frac{4eN}{n}\varepsilon\right)^{n/4},$$

where the final inequality follows from the fact that Z and S is a pair of good distributions.

□

A.1 Good Distributions

We recall the good pair of distributions Z and S defined in [LM20], described using multisets, $\mathcal{Z}$ and $\mathcal{S}$, respectively:[16]

$$\mathcal{S} := \bigcup_{j\in\{T,T+2,\cdots,N\}} \left\{ \left(b, \binom{j-1}{T-1}\right) : b \in \{0,1\}^N, \|b\|_0 = j \right\}, \text{ and}$$

$$\mathcal{Z} := \{(0^N, 1)\} \cup \bigcup_{j\in\{T+1,T+3,\cdots,N\}} \left\{ \left(b, \binom{j-1}{T-1}\right) : b \in \{0,1\}^N, \|b\|_0 = j \right\}.$$

The proof that Z and S constitute a pair of good distributions can be found in [LM20, Theorem 3 and Corollary 1].

B LPKE via Non-Interactive BARGs

In this section, we recall the definition of *somewhere extractable* BARGs from the literature, and also define a variant thereof, which we call *honestly somewhere extractable.* We prove that somewhere-extractable BARGs imply (single databased) *private information retrieval* (PIR), which in turn is known to imply *statistically sender-private oblivious transfer* and lossy public-key (LPKE) encryption [DMO00, PVW08].

The resulting LPKE suffers from a negligible decryption error, which makes it insufficient for the NISZKA amplification theorem in Section 4. We observe that if the BARGs satisfy honest-somewhere extraction then the resulting LPKE has a stronger correctness guarantee, which is also sufficient for our amplification theorem.

B.1 PIR from Somewhere Extractability

Definition B.1 (Somewhere Extractability)**.** *A batch protocol* $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ *for a relation* $\mathcal{R}$ *is* somewhere extractable *if it satisfies CRS indistinguishability, and if there is a PPT extractor* E *such that, for every polynomial* t *and polynomial-size circuit family of provers* $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_{\lambda\in\mathbb{N}}$*, there is a negligible function* μ *such that for every* $\lambda \in \mathbb{N}$*,* $t = t(\lambda)$*, and* $i^* \in [t]$*:*

$$\Pr_{crs^*,td,\mathsf{E}}[\mathsf{V}(crs^*, (x_1,\ldots,x_t), \pi) \text{ accepts } \wedge \ (x_{i^*}, w) \notin \mathcal{R}] \leqslant \mu(\lambda),$$

where $(crs^*, td) \leftarrow \mathsf{TGen}(1^\lambda, 1^t, i^*)$*,* $((x_1,\ldots,x_t),\pi) \leftarrow \mathsf{P}^*_\lambda(crs^*, i^*)$*, and* $w \leftarrow \mathsf{E}(td, i^*, crs^*, (x_1,\ldots,x_t), \pi)$*.*

Definition B.2 (PIR [CKGS98])**.** *A one-round, single-database* PIR *is a tuple of polynomial-time algorithms* $(\mathsf{Q}, \mathsf{D}, \mathsf{R})$ *with the following syntax:*

- $(k, Q) \leftarrow \mathsf{Q}(1^\lambda, \ell, i)$*. The randomized* user query *algorithm takes as input a security parameter* $\lambda \in \mathbb{N}$*, a parameter* $\ell \in \mathbb{N}$ *that represents the length of the database, and a target index* $i \in [\ell]$*. It outputs a key* k *and a query* Q*.*
- $a := \mathsf{D}(D, Q)$*. The deterministic* database answer *algorithm takes as input a database* $D := (D_1,\ldots,D_\ell) \in \{0,1\}^\ell$ *and a query* Q *and outputs an answer* a*.*
- $d := \mathsf{R}(k, a)$*. The deterministic* user reconstruct *algorithm takes as input the key* k *and answer* a *and outputs a data bit* d*.*

We require the following properties:

[16] A multiset $\mathcal{M}$ over a domain Ω is represented as $\{(x, m_x) : x \in \Omega\}$ where $m_x \in \mathbb{N}$ is the multiplicity of the element x. The cardinality of $\mathcal{M}$ is then defined as $|\mathcal{M}| := \sum_{x\in\Omega} m_x$. The probability distribution M induced by $\mathcal{M}$ is defined naturally: the probability of an element $x \in \Omega$ is $m_x/|\mathcal{M}|$.

PIR scheme $(\mathsf{Q}, \mathsf{D}, \mathsf{R})$, built using a somewhere-extractable BARG $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ and hard sampler for the relation $\mathcal{R}_f$ from Eq. (9).

$\underline{(k, Q) \leftarrow \mathsf{Q}(1^\lambda, \ell, i)}$

1. Use the hard sampler for $\mathcal{R}_f$ to generate ℓ instance-witness pairs

$$\boldsymbol{q} := (((y_{1,0}, y_{1,1}), x_{1,0}, x_{1,1}), \ldots, ((y_{\ell,0}, y_{\ell,1}), x_{\ell,0}, x_{\ell,1})).$$

2. Use TGen to sample a CRS with trapdoor set up at index i: $(crs^*, td) \leftarrow \mathsf{TGen}(1^\lambda, 1^\ell, i)$
3. Output $(td, (crs^*, \boldsymbol{q}))$

$\underline{a := \mathsf{D}(D, Q)}$

1. Run the batch prover on witnesses determined by D:

$$\pi \leftarrow \mathsf{P}(crs^*, ((y_{1,0}, y_{1,1}), \ldots, (y_{\ell,0}, y_{\ell,1})), (x_{1,D_1}, \ldots, x_{\ell,D_\ell})).$$

2. Output π

$\underline{d := \mathsf{R}(k, a)}$

1. Use BARG extractor to extract witness at i: $w \leftarrow \mathsf{E}(td, i, crs^*, ((y_{1,0}, y_{1,1}), \ldots, (y_{\ell,0}, y_{\ell,1})), \pi)$
2. Halt without output if the BARG verifier rejects, i.e., $\mathsf{V}(crs^*, ((y_{1,0}, y_{1,1}), \ldots, (y_{\ell,0}, y_{\ell,1}), \pi)) = 0$
3. Otherwise,

$$d := \begin{cases} 0 & \text{if } f(w) = y_{i,0} \\ 1 & \text{otherwise.} \end{cases}$$

4. Output d

Figure 10: PIR scheme $(\mathsf{Q}, \mathsf{D}, \mathsf{R})$.

1. Correctness of reconstruction. *There exists a negligible function μ such that for every $\lambda \in \mathbb{N}$, $\ell \in \mathsf{poly}(\lambda)$, database $D \in \{0,1\}^\ell$ and query $i \in [\ell]$:*

$$\Pr_{(k,Q) \leftarrow \mathsf{Q}(1^\lambda, \ell, i)} [\mathsf{R}(k, \mathsf{D}(D, Q)) = D_i] \geqslant 1 - \mu(\lambda).$$

2. Succinctness. *We say that the PIR is succinct if $|a| \leqslant \ell^\epsilon$ for some $\epsilon < 1$. We say that the PIR is* fully *succinct if there exists* poly *such that $|a| \leqslant \mathsf{poly}(\lambda)$.*
3. Computational user privacy. *No efficient adversary can distinguish between user queries on two target indices. That is, for every polynomial-size circuit family of distinguishers $\mathsf{A} = \{\mathsf{A}_\lambda\}_{\lambda \in \mathbb{N}}$, there is a negligible function μ, such that for all $\lambda \in \mathbb{N}$, $\ell \in \mathsf{poly}(\lambda)$ and $i, j \in [\ell]$*

$$\left| \Pr_{(k,Q) \leftarrow \mathsf{Q}(1^\lambda, \ell, i)} [1 \leftarrow \mathsf{A}_\lambda(Q)] - \Pr_{(k,Q) \leftarrow \mathsf{Q}(1^\lambda, \ell, j)} [1 \leftarrow \mathsf{A}_\lambda(Q)] \right| \leqslant \mu(\lambda).$$

The PIR scheme constructed from somewhere-extractable BARG is described in Fig. 10. It relies on the

fact that somewhere-extractable BARG implies one-way functions (OWFs) , and given a OWF f, we can define an NP relation

$$\mathcal{R}_f := \{((y_0, y_1), x) : f(x) = y_0 \vee f(x) = y_1\} \tag{9}$$

that allows sampling an instance along with *two* witnesses. To be precise, the hard sampler for $\mathcal{R}_f$ invokes the OWF on two random preimages x_0 and x_1, and then outputs the instance $(y_0 := f(x_0), y_1 := f(x_1))$.

Theorem B.3 (Somewhere-Extractable Non-Interactive BARG Implies PIR)**.** *If there exists a somewhere-extractable non-interactive BARG* $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$*, then the scheme in Fig. 10 is a one-round, single database PIR. If the size of* BARG*s is independent of the number of instances, then the PIR is fully succinct.*

Proof Sketch. User privacy follows from CRS indistinguishability (Definitions 2.15 and B.1). Correctness, follows from somewhere extractability and one-wayness. By somewhere extractability, it is guaranteed that with overwhelming probability the extractor returns *some* witness of $(y_{i,0}, y_{i,1})$, i.e., *some* pre-image of $y_{i,0}$ or $y_{i,1}$ under f. One-wayness of f ensures that it returns a witness corresponding to y_{i,D_i} and not $y_{i,\overline{D}_i}$. Indeed, since Q generates the BARG proof based only on the witness/pre-image x_{i,D_i}, it is oblivious of the other witnesses $x_{i,\overline{D}_i}$. As a result, the extractor outputting $x_{i,\overline{D}_i}$ is tantamount to breaking f's one-wayness. □

B.2 Honest Somewhere Extractability

Definition B.4 (Honest Somewhere Extractability)**.** *A batch protocol* $(\mathsf{Gen}, \mathsf{TGen}, \mathsf{P}, \mathsf{V})$ *for a relation* $\mathcal{R}$ *is* honestly somewhere extractable *if it satisfies CRS indistinguishability, and if there is a PPT extractor* E *such that, for every* $\lambda \in \mathbb{N}$*,* $t = t(\lambda)$*,* $(x_1, w_1), \ldots, (x_t, w_t) \in \mathcal{R}$ *and* $i^* \in [t]$*:*

$$\Pr_{crs^*, td, \mathsf{E}}[w_{i^*} \neq w] = 0,$$

where $(crs^*, td) \leftarrow \mathsf{TGen}(1^\lambda, 1^t, i^*)$*,* $\pi \leftarrow \mathsf{P}(crs^*, (x_1, w_1), \ldots, (x_t, w_t))$*, and* $w \leftarrow \mathsf{E}(td, i^*, crs^*, (x_1, \ldots, x_t), \pi)$*.*

Remark 9. *We can in fact further weaken the above requirement, asking for perfect correctness for* almost any CRS*. Namely, that with overwhelming probability over the choice of CRS, extraction is perfect.*

Going back to the construction if Fig. 10, in case the BARG satisfies *honest* somewhere extractability (Definition B.4), then the construction satisfies perfect correctness of reconstruction. Indeed, the BARG proof generated by Q is honest, the extractor is guaranteed to return the actual witness used at position i, which is x_{i,D_i}.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il