Micali's SNARG, Function Vector Commitments, and Fiat–Shamir

Ziyi Guan
ziyiguan@mit.edu
MIT

Eylon Yogev
eylon.yogev@biu.ac.il
Bar-Ilan University

Abstract

Micali's construction of succinct non-interactive arguments (SNARGs) combines a probabilistically checkable proof (PCP), a vector commitment, and the Fiat–Shamir transformation. Its security is established in the random oracle model, but replacing the random oracle with an explicit hash function presents a fundamental challenge: Fiat–Shamir can fail for interactive arguments, and certain choices of the commitment scheme make Micali's construction unsound for every instantiation of the Fiat–Shamir hash.

We show how to instantiate Micali's construction in the standard model assuming LWE and function vector commitments with *function statistical binding* for P/poly, obtaining a non-adaptively sound SNARG for NP. Our construction combines such a commitment with an LWE-based correlation-intractaxble hash and a PCP satisfying a new property, *shadow soundness*. A PCP shadow is a short digest that preserves the information needed to determine the verifier's decision. By statistically binding the commitment to this shadow, we obtain the sparse relation needed to prove Fiat–Shamir soundness.

We construct shadow PCPs for NP whose shadow algorithms lie in NC, and we give a feasibility instantiation of the commitment scheme from strong assumptions. Our results thus identify a concrete cryptographic target for instantiating Micali's SNARG under weaker assumptions, and show that the framework does not admit an attack that succeeds for every choice of its components.

Keywords: SNARGs, Fiat–Shamir, function vector commitments, shadow PCPs, correlation-intractable hashing

Contents

1 Introduction **3**
1.1 Our results . . . 5

2 Technical overview **7**
2.1 PCP shadows . . . 7
2.2 Function vector commitments . . . 9
2.3 Our SNARG compiler . . . 10

3 Preliminaries **13**
3.1 Correlation intractability . . . 13
3.2 Non-interactive arguments . . . 14
3.3 Probabilistically checkable proofs . . . 14

4 PCP shadows **16**
4.1 Shadow PCP for NP . . . 17
4.2 Amplification of shadow soundness . . . 18

5 Function vector commitment scheme **21**
5.1 Definition . . . 21

6 SNARG from function VC and shadow PCP **23**
6.1 SNARG construction . . . 24
6.2 SNARG soundness . . . 25

A Function VC for circuits **31**
A.1 Fully homomorphic encryption . . . 31
A.2 Construction . . . 32

Acknowledgments **35**

References **35**

1 Introduction

Succinct non-interactive arguments (SNARGs) for NP [Mic00] constitute a fundamental primitive in modern cryptography. They allow a prover to convince a verifier of the validity of a statement with a proof whose size is significantly smaller than the time required to verify the statement directly, and ideally, much smaller than the witness itself. This "non-interactive" feature is particularly crucial for real-world applications, enabling scenarios such as delegating computation to untrusted servers, scaling blockchains via rollups, and constructing privacy-preserving cryptocurrencies.

Building on the seminal work of Kilian [Kil92], Micali [Mic00] provided the first construction of a SNARG for NP in the random oracle model (ROM). Since then, there has been tremendous progress in the design of efficient SNARGs. However, the vast majority of efficient constructions are proven secure only within idealized models. Some of these works rely solely on the random oracle model or the generic group model (GGM) to establish security (e.g., [Gro16; BCS16; CY21a; CY21b; ACY23; AHIV17; BBHR18; ZCF24; ACFY24; ACFY25]). Others rely on the random oracle model in addition to other specific cryptographic assumptions (e.g., [BBBPWM18; AGLMS23; AFLN24; BC25]). While these models provide a heuristic rationale for security, they do not guarantee security in the standard model, where the hash function or group is instantiated with concrete algorithms.

A parallel line of work has constructed SNARGs for NP in the common reference string (CRS) model based on various *non-falsifiable* assumptions (e.g., knowledge of exponent assumptions) [Gro10; BCCT12; DFH12; BCCT13; Lip13; GGPR13; BCIOP13; BCP13; BISW17; Bit+14; BISW18; ACLMT23; CLM23]. In the CRS model, the prover and verifier rely on a trusted reference string (or random string) generated during a setup phase.

SNARGs from falsifiable assumptions. Despite their practical significance and widespread deployment, the theoretical foundations of SNARGs remain somewhat unsatisfactory. For a long time, the existence of SNARGs based solely on falsifiable assumptions remained an open question. The first construction of a SNARG for NP from (sub-exponentially) falsifiable assumptions was provided by Sahai and Waters [SW14], who gave a (non-adaptive) construction based on indistinguishability obfuscation (iO) and one-way functions.[1]

To get around this challenge, an alternative approach is to relax the target setting—either by restricting the class of languages or by weakening the notion being achieved. Along these lines, several works construct SNARGs from falsifiable assumptions for P [CJJ22; KVZ21; HJKS22; CGJJZ23], P/poly [GZ21; WW23b; WW23a; BCFL23; WW24b; Wee25a; Wee25b], batch NP [CJJ22; CJJ21; GSWW22; WW22; DGKV22; PP22; KLVW23; CGJJZ23; DWW24], monotone policy batch NP [BBKLP23; NWW24; NWW25], and NP languages with propositional proofs of non-membership [JKLM25].

Building adaptively sound SNARGs for NP from falsifiable assumptions is a challenging problem, as any such result must circumvent known black-box separations [GW11; CGKS23]. While recent works have improved the Sahai–Waters construction to achieve adaptive soundness [WW24a; WZ24; WW25], these results still necessitate the full power of iO. Although iO is a powerful primitive, it is considered a heavy assumption, both in terms of the complexity of its construction and the concrete efficiency of its instantiations. Consequently, constructing SNARGs from standard, well-studied assumptions (such as the hardness of lattice problems) without relying on iO or idealized models

[1] Recall that iO alone does not imply one-way functions and must be supplemented with a cryptographic assumption (e.g., one-way functions). However, this additional assumption can sometimes be replaced by a weaker worst-case assumption [KMNPRY22].

remains a major open problem in the field. In concurrent work, Devadas et al. [DHKKLM26] constructed a SNARG for NP based on the learning with errors (LWE) assumption (or other standard assumptions) and a mathematical conjecture about multivariate polynomials over the reals (low-norm Nullstellensatz hypothesis).

The challenge of Fiat–Shamir for arguments. If we allow interactions, the situation is much better. In his seminal work, Kilian [Kil92] showed how to construct succinct *interactive* arguments for NP assuming only the existence of collision-resistant hash functions (CRH); moreover, it was later shown that even multi-collision resistant hash functions suffice [KNY18]. Kilian's protocol works by having the prover construct a PCP for the statement and commit to it using a vector commitment scheme (typically instantiated via a Merkle tree). The verifier then queries specific locations of the PCP, for which the prover provides openings to the commitment.

To remove interaction, Micali [Mic00] proposed applying the Fiat–Shamir transformation to Kilian's protocol. The idea is to replace the verifier's random queries with the output of a cryptographic hash function applied to the prover's commitment. Micali proved that this construction yields a secure SNARG in the ROM. Consequently, a natural approach to obtain a SNARG from falsifiable assumptions is to instantiate this blueprint using a concrete hash function in the standard model.

In recent years, there has been significant progress in instantiating the Fiat–Shamir heuristic in the standard model. A line of work has successfully established hash functions that can securely instantiate Fiat–Shamir for specific classes of interactive protocols based on standard assumptions (e.g., [CCR16; KRR17; CCRR18; HL18; Can+19; PS19; BKM20; JJ21; HLR21; CJJ22; HJKS22; KLV23]). However, a crucial limitation of these works is that they primarily apply to interactive *proofs* (i.e., protocols with statistical soundness). Kilian's protocol, however, is an interactive *argument* (i.e., it has only computational soundness) due to its reliance on a succinct commitment scheme.

Standard techniques for instantiating Fiat–Shamir generally fail when compiling argument systems into non-interactive counterparts, rendering the instantiation of Micali's construction in the standard model a formidable challenge. Indeed, there is a line of work that highlights the risks of replacing the random oracle with a concrete hash function in an argument scheme, regardless of how the hash function is implemented [CGH04; Bar01; GK03; KRS25; AY25].

Furthermore, Bartusek et al. [BBHMR19] identified strong barriers specifically targeting Micali's blueprint. They constructed a specific CRH for which the Micali transformation results in an unsound protocol for *any* concrete instantiation of the Fiat–Shamir hash function. This implies that any successful instantiation of Micali's construction cannot treat the CRH as a generic component. Rather, it must leverage some specific structure of the underlying CRH.

This raises a natural question:

Can Micali's SNARG be instantiated with an explicit hash function while preserving soundness?

Function vector commitments. A promising primitive for bridging this gap is a *function vector commitment* (function VC), which we focus on in this work. A function VC is parameterized by a family of functions $\mathcal{F}$, and its public parameters are generated relative to a prescribed function $f \in \mathcal{F}$ that remains hidden from the committer. It supports the usual openings to individual coordinates of the committed vector m, while its *function statistical binding* property statistically binds the commitment to the value $f(m)$. One can observe that function VCs immediately yield a succinct

two-message designated-verifier argument.[2] However, it is not clear how to use a function VC within Micali's construction, where the verifier's queries are obtained by hashing the commitment itself and are therefore correlated with it.

1.1 Our results

In this work, we show how to instantiate Micali's construction in the standard model: a function VC for committing to the PCP proof and an explicit Fiat–Shamir hash function. In particular, we use an LWE-based correlation-intractable hash function in place of the random oracle and show that function statistical binding is sufficient to establish soundness.

Theorem 1. *Assume LWE and the existence of function vector commitments with function statistical binding for* P/poly*. Then Micali's construction can be instantiated as a non-adaptively sound SNARG for* 3SAT *in the standard model.*

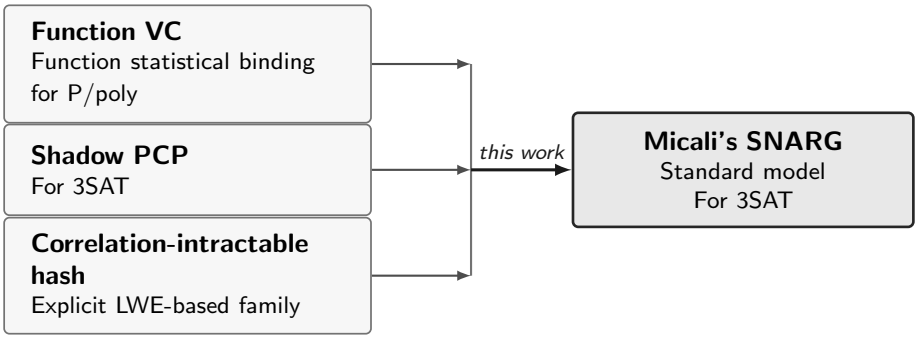


Figure 1: Our construction combines a function VC, a shadow PCP, and an LWE-based correlation-intractable hash to obtain a standard-model instantiation of Micali's SNARG.

To prove Theorem 1, we introduce a new property of probabilistically checkable proofs which we term ***shadow soundness***. Central to this notion is a *shadow algorithm* that compresses the PCP proof while preserving enough information to determine whether the PCP verifier accepts. More generally, if the shadow algorithm belongs to a function class $\mathcal{F}$, our construction only requires a vector commitment with function statistical binding for $\mathcal{F}$. We show that every language in NP has a shadow PCP whose shadow algorithm is in NC. Thus, function vector commitments for NC already suffice.

Thus, our result isolates one cryptographic primitive whose construction would give SNARGs for NP: a succinct vector commitment that is statistically binding to the output of a hidden NC circuit applied to the committed string. We do not know how to build such a commitment from falsifiable assumptions such as LWE, but as a feasibility result one can construct it from indistinguishability obfuscation (see Section A).

[2]For a relation R and instance x, the verifier defines $f_x(w) \coloneqq R(x, w)$, generates function VC parameters together with an extraction trapdoor relative to f_x, and sends the public parameters as the first message. The prover commits to a witness w and responds with the commitment and a valid opening at a fixed location. The verifier checks the local opening, uses the trapdoor to extract the value bound to the commitment, and accepts if and only if this value is 1. Function statistical binding ensures that an accepting opening is consistent with some w' satisfying $R(x, w') = 1$.

Concurrent work via non-signaling PCPs. In concurrent work, Devadas et al. [DHKKLM26] propose a route to SNARGs for NP from standard cryptographic assumptions via exponential-length, constant-locality non-signaling PCPs. Their proposed instantiation relies on a low-norm Nullstellensatz hypothesis. Subsequent works have shown that this hypothesis is false: Jin [Jin26] refutes it for the AND code, and Lombardi [Lom26] gives counterexamples to several generalizations that would also have sufficed for the construction. These results invalidate the proposed assumptions underlying this route to soundness, although they do not rule out the broader non-signaling-PCP approach. Our reduction offers a different target that is unaffected by these refutations: function vector commitments with function statistical binding for NC. This primitive already admits an instantiation from sub-exponentially secure iO and LWE (Section A), providing a feasibility result for our approach. Moreover, our reduction only requires binding to the output of a hidden NC circuit, rather than arbitrary polynomial-size circuits. Constructing such commitments from weaker assumptions therefore remains a concrete direction toward instantiating Micali's SNARG without iO.

Refuting universal attacks on the Micali framework. As discussed, a line of work has established "universal" impossibility results for the Fiat–Shamir transformation, showing that for certain protocols, the compiled argument is insecure regardless of the concrete hash function used [BBHMR19]. However, such impossibility results generally do not apply to interactive *proofs*. This is because the existence of correlation-intractable hash (CIH) functions guarantees that, for a broad class of proofs, a secure instantiation of Fiat–Shamir *does* exist. The mere existence of such a hash function, even if inefficient, serves as a counterexample that rules out any universal attack.

Our work provides a parallel guarantee for the Kilian–Micali *argument* system. By constructing a concrete, secure instantiation of Micali's construction (via a specific PCP, a vector commitment, and an LWE-based hash), we provide an existential proof that refutes the possibility of universal attacks against this paradigm. Consequently, the Micali framework is not inherently unsound; any successful attack on the transformation must necessarily rely on specific, contrived choices of the underlying components rather than the structure itself.

2 Technical overview

We give a high-level overview of our techniques. As mentioned, the general blueprint of the SNARG construction is the work of [Mic00] (see also [CY24]): the prover computes a PCP proof and commits to it using a vector commitment; the PCP verifier's randomness is derived from the commitment via a Fiat–Shamir hash; and the prover then supplies the queried PCP symbols together with opening proofs. Our contribution lies in the choice of the three components and in how they interact.

To motivate our specific design, we first recall how the Fiat–Shamir transformation is typically instantiated in the standard model for statistically sound protocols. For 3-message interactive *proofs* (protocols with statistical soundness), secure instantiation is achieved using a correlation-intractable hash (CIH) function [CCR16; KRR17; CCRR18; HL18; Can+19; PS19; BKM20; JJ21; HLR21; CJJ22; HJKS22; KLV23]. This works because such protocols induce an underlying ***sparse relation***: for any false statement, the set of "bad" verifier challenges (those that lead to acceptance) is negligibly small. A CIH function is simply defined to avoid outputting values within this sparse set.

However, this standard paradigm fails for Kilian's protocol. Because Kilian's protocol is an interactive *argument*, relying on the computational binding of the vector commitment, there is no underlying sparse relation. Since the commitment is only computationally binding and the committed string is hidden from the verifier, one cannot define a fixed sparse set of bad challenges in the standard information-theoretic sense. Thus, one cannot directly apply the same CIH approach, and some new ideas are necessary. In particular, we need to have some statistical property hiding under the computational commitment.

Our construction keeps the structure of Micali's SNARG and changes what is required of each component so that such a statistical property exists. The PCP is a *shadow PCP* (Section 2.1), the Merkle tree over a collision-resistant hash is replaced by a *function vector commitment* (Section 2.2), and the random oracle is replaced by a correlation-intractable hash (Section 2.3). The statistical property is the PCP's *shadow*: a short digest of the committed proof string that determines the verifier's decision on every choice of randomness. The commitment is statistically binding to the shadow, even though it is only computationally binding to the proof string itself. Once the shadow is fixed, the set of verifier randomness that it accepts is sparse, and we are back in the setting where a correlation-intractable hash applies. This is how we couple the PCP and the commitment to induce a sparse relation. Our approach is inspired by works that construct SNARGs for subclasses of NP from somewhere statistically binding hashing [BBHMR19; KVZ21; CJJ21; CJJ22; PP22; DGKV22; KLVW23].

2.1 PCP shadows

Consider the $\mathsf{3SAT}$ language over n variables. The first component is an information-theoretic PCP satisfying a soundness notion we call ***shadow soundness***. A shadow is a digest of a PCP string Π (honest or malicious) into a short state z; in our construction the state has size $\tilde{O}(\sqrt{n})$. The digest is computed by a (possibly randomized) algorithm F.

The shadow is designed to predict the verifier's output $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho)$ on every choice of randomness. This is captured by a decider D that takes the shadow and the PCP randomness ρ and outputs a decision bit. *Correctness* asks that for every fixed ρ, with high probability over the randomness of F, $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho) = \mathsf{D}(\mathbb{x}, z, \rho)$. We additionally require *sparsity*, which says that for $\mathbb{x} \notin L$ the decider accepts only a sparse set of randomness, and *consistency*, which says that proof strings agreeing on the verifier's query set produce shadows on which the decider agrees. Formally:

Definition 1 (Shadow soundness)**.** *A* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *has shadow soundness if there exists a function* F *with randomness complexity* r_F *and a deterministic algorithm* D *(a decider) such that for every instance* $\mathbb{x} \notin L(R)$*, the following holds:*

1. ***Correctness.*** *For any* $\widetilde{\Pi}$ *and verifier randomness* $\rho \in \{0,1\}^{\mathsf{r}}$*:*

$$\Pr\left[\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho) \neq \mathsf{D}(\mathbb{x}, z, \rho) \;\middle|\; \begin{array}{l} \gamma \leftarrow \{0,1\}^{\mathsf{r}_\mathsf{F}} \\ z := \mathsf{F}(\mathbb{x}, \widetilde{\Pi}; \gamma) \end{array}\right] \leq \epsilon_\mathsf{c}(\mathbb{x})\,.$$

2. ***Sparsity.*** *There exists an efficiently computable set* Z *where* $\mathsf{Im}(\mathsf{F}(\mathbb{x}, \cdot)) \subseteq Z$ *such that for any shadow state* $z \in Z$*:*

$$\Pr\left[\mathsf{D}(\mathbb{x}, z, \rho) = 1 \mid \rho \leftarrow \{0,1\}^{\mathsf{r}}\right] \leq \epsilon_\mathsf{s}(\mathbb{x})\,.$$

3. ***Consistency.*** *For every shadow randomness* $\gamma \in \{0,1\}^{\mathsf{r}_\mathsf{F}}$*, verifier randomness* $\rho \in \{0,1\}^{\mathsf{r}}$*,* $\widetilde{\Pi}$ *and* $\widetilde{\Pi}'$ *such that* $\widetilde{\Pi}[\mathsf{Q}] = \widetilde{\Pi}'[\mathsf{Q}]$*, where* Q *is the query set of the PCP verifier* $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho)$*:*

$$\mathsf{D}(\mathbb{x}, z, \rho) = \mathsf{D}(\mathbb{x}, z', \rho)\,,$$

where $z := \mathsf{F}(\mathbb{x}, \widetilde{\Pi}; \gamma)$ *and* $z' := \mathsf{F}(\mathbb{x}, \widetilde{\Pi}'; \gamma)$*.*

Moreover, we say that PCP *has shadow state size* s_F *if* F *outputs at most* s_F *bits.*

Remark: shadow soundness and ETH. The exponential time hypothesis (ETH) posits that solving 3SAT on n variables requires time exponential in n. This implies that no PCP for 3SAT can have a proof of size $o(n)$, as this would allow solving the problem in $2^{o(n)}$ time by iterating over all proofs. Since our construction yields a shadow state of size $o(n)$, it might appear to contradict ETH (or its randomized variant) by using the shadow as a succinct witness.

However, this does not yield a subexponential-time algorithm, because the guarantees in Definition 1 are stated only for $\mathbb{x} \notin L$ and only for states in the set Z. When $\mathbb{x} \in L$, the shadow of the honest proof need not lie in Z, so enumerating Z need not find an accepting state. Enumerating all states of the right length does not help either: sparsity holds only inside Z, and a state outside Z may be accepted on every choice of randomness whether or not $\mathbb{x} \in L$. Since membership in the image of $\mathsf{F}(\mathbb{x}, \cdot)$ cannot be tested without a full proof string, the enumerator cannot tell such a state apart from the shadow of a genuine proof.

How to use shadow PCPs. The idea is to have the prover commit to the PCP string in a way that is statistically binding to its *shadow*. To this end we introduce a vector commitment satisfying *function statistical binding* with respect to a family of functions $\mathcal{F}$: the commitment is succinct, so it cannot be statistically binding to the message, but it is statistically binding to $f(m)$ for a hidden $f \in \mathcal{F}$ with short output. The function f will be the shadow algorithm. When the shadow is randomized, its randomness must stay hidden from the prover, since a prover who knows it can tailor the proof string to the digest; this is why we also require the commitment to hide f. We elaborate more on our VC below in Section 2.2.

Constructing shadow PCPs: the block transformation. We give a generic transformation that turns a nonadaptive PCP with small randomness complexity, $O(1)$ queries, and constant alphabet into a shadow PCP with a deterministic shadow. Let $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ be such a PCP for 3SAT with randomness complexity r. The new prover $\mathbf{P}'$ is unchanged. The new verifier $\mathbf{V}'$ partitions the 2^{r} choices of randomness of $\mathbf{V}$ into consecutive blocks of size $2^{\mathsf{r}/2}$, samples a block

index b, and accepts if and only if $\mathbf{V}$ accepts on every randomness in block b. The shadow of a proof string is the list of $2^{\mathsf{r}/2}$ bits recording, for each block, whether $\mathbf{V}$ accepts throughout it; the decider reads off the bit for block b. Correctness and consistency hold with no error, and sparsity follows from the soundness of $\mathbf{V}$: for $\mathtt{x} \notin L$, only a small fraction of blocks can be accepted throughout.

Lemma 1 (Informal)**.** *Any nonadaptive PCP for* 3SAT *with randomness complexity* r*,* $O(1)$ *queries, and constant alphabet can be transformed into a shadow PCP with a deterministic shadow of size* $2^{\mathsf{r}/2}$*, query complexity* $O(2^{\mathsf{r}/2})$*, and, for every fixed* $\mathtt{x}$*, a shadow algorithm computable by polynomial-size,* $O(\log n)$*-depth circuits.*

Applying the transformation to a PCP for 3SAT with randomness complexity $\log n + O(\log\log n)$ gives shadow state and query complexity $\tilde{O}(\sqrt{n})$. For every fixed $\mathtt{x}$, nonadaptivity allows the query locations for each verifier random string to be hardwired, and the $O(1)$-query predicate over a constant alphabet can also be hardwired. Evaluating these predicates in parallel and computing each block conjunction with a balanced tree gives polynomial-size, $O(\log n)$-depth circuits for the shadow algorithm.

Since the shadow algorithm is deterministic, the correctness error in Definition 1 is zero and the shadow randomness plays no role. We keep randomized shadows in the definition because they are natural (a randomized digest could plausibly be much smaller), but note that using one costs a stronger hiding requirement on the vector commitment; see Section 2.3.

2.2 Function vector commitments

The second component is a new notion of vector commitment, which we call a ***function vector commitment*** (function VC). It is parameterized by a family of functions $\mathcal{F}$ and is statistically binding to the value of a hidden $f \in \mathcal{F}$ on the committed vector. We believe the notion is of independent interest.

The notion generalizes somewhere statistically binding (SSB) hashing [HW15; OPWW15], where the hidden object is an index rather than a function. Recall that an SSB hash is initialized with a secret index i hidden from the adversary; while the hash is compressing (and thus only computationally binding globally), it guarantees that one can *extract* the i-th bit of the message, i.e., it is statistically binding specifically for the i-th bit. Formally, a hash family $\mathcal{H} = \{h_{\mathsf{k}}\}_{\mathsf{k} \leftarrow \mathcal{H}.\mathsf{Gen}}$ is an SSB hash if there exists an efficient extractor $\mathcal{H}.\mathsf{Extract}$ such that for any (unbounded) adversary $\mathcal{A}$ and index i,

$$\Pr\left[\begin{array}{l|l} \mathcal{H}.\mathsf{Check}(\mathsf{k}, v, i, \sigma, \mathsf{pf}) = 1 & (\mathsf{k}, \mathsf{td}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda, i) \\ \wedge\, \sigma \neq \widetilde{\sigma} & (v, \sigma, \mathsf{pf}) \coloneqq \mathcal{A}(\mathsf{k}) \\ & \widetilde{\sigma} \coloneqq \mathcal{H}.\mathsf{Extract}(\mathsf{td}, v) \end{array}\right] \leq \mathrm{negl}(\lambda)\,,$$

where v is the hash output chosen by $\mathcal{A}$, σ is the alleged value at the i-th position of the preimage of v, and pf is an opening proof.

In our case, we initialize the VC with a hidden function f. Although the commitment is succinct, it is statistically binding with respect to the evaluation of f. We formalize this via a property called *function statistical binding*. Specifically, a function VC is a tuple of efficient algorithms

$$\mathsf{VC} = (\mathsf{VC.Gen}, \mathsf{VC.Commit}, \mathsf{VC.Open}, \mathsf{VC.Check}, \mathsf{VC.Extract})$$

where $(\mathsf{VC.Gen}, \mathsf{VC.Commit}, \mathsf{VC.Open}, \mathsf{VC.Check})$ works similarly to a standard VC, and function statistical binding is defined via the extractor $\mathsf{VC.Extract}$:

Definition 2 (Informal)**.** VC *satisfies* **function statistical binding** *if for every function* f *and (unbounded) adversary* A_{VC}*,*

$$\Pr\left[\begin{array}{l|l} \mathsf{VC.Check}(\mathsf{pp},\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf})=1 & (\mathsf{pp},\mathsf{td})\leftarrow\mathsf{VC.Gen}(1^\lambda,f)\\ \wedge\,\forall\,\widetilde{m}\in\Sigma^\ell:(f(\widetilde{m})\neq y\ \vee\ \widetilde{m}[\mathsf{Q}]\neq\mathsf{a}) & (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf})\coloneqq A_{\mathsf{VC}}(\mathsf{pp})\\ & y\coloneqq\mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\end{array}\right]\leq\mathrm{negl}(\lambda)\,.$$

Note that function statistical binding is stated via the extractor $\mathsf{VC.Extract}$: given the trapdoor, one can extract a value y from the commitment, and any accepting local opening must be consistent with some message m satisfying $f(m)=y$. This is stronger than merely requiring that commitments to messages with different f-values differ, and it is what the soundness analysis uses: from an accepting opening we can always pass to a full proof string whose shadow equals the extracted value. We also note that when f is a projection $m\mapsto m[i]$, the definition coincides with SSB hashing with local openings.

We also require *function hiding*: for $f\neq f'$ in $\mathcal{F}$, public parameters generated for f and for f' are computationally indistinguishable. Function hiding is used twice. It keeps the shadow randomness hidden from the prover, as discussed in Section 2.1. More importantly, it makes the CRS independent of the instance: the shadow algorithm $\mathsf{F}(\mathbb{x},\cdot)$ depends on $\mathbb{x}$, whereas a SNARG must have a single CRS for all instances. Our setup therefore generates parameters for a fixed dummy function, and the security proof for a given $\mathbb{x}\notin L$ switches, via function hiding, to parameters for $\mathsf{F}(\mathbb{x},\cdot)$, where binding to the shadow is available.

2.3 Our SNARG compiler

The construction in this section is generic: it uses a shadow PCP whose shadow algorithm lies in some function family $\mathcal{F}$, a function VC for the same family $\mathcal{F}$, and a correlation-intractable hash family from LWE. Theorem 1 then follows by setting $\mathcal{F}\coloneqq\mathsf{NC}$ and plugging in the shadow PCP of Lemma 1.

Warm-up: designated verifier SNARG. To build intuition, we first describe how to construct a *designated verifier* SNARG (dv-SNARG) using a shadow PCP and a function vector commitment. In this setting, the verifier holds a secret key—specifically, the trapdoor td of the vector commitment. We then demonstrate how to modify the construction to be *publicly verifiable*.

The designated verifier construction is given as follows.

- $\mathcal{P}$:
  1. Compute the PCP proof Π.
  2. Commit to Π using the function VC and obtain cm.
  3. Sample verifier randomness ρ and provide the local openings $(\mathsf{Q},\mathsf{a},\mathsf{pf})$ corresponding to ρ.
  4. Send $(\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf})$.
- $\mathcal{V}$:
  1. Use $\mathsf{VC.Extract}$ to extract the PCP shadow z underneath the commitment cm.
  2. Sample verifier randomness ρ (not necessarily the same as the one sampled by the prover).
  3. Check the following:
     (a) $\mathsf{VC.Check}(\mathsf{pp},\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf})=1$;
     (b) $\mathsf{D}(\mathbb{x},z,\rho)=1$.

By the function statistical binding property of the VC, we know that if $\mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) = 1$, there exists a PCP proof string $\widetilde{\Pi}$ such that $\widetilde{\Pi}[\mathsf{Q}] = \mathsf{a}$ and $\mathsf{F}(\mathbb{x}, \widetilde{\Pi}) = z$. Then the correctness of the PCP shadow ensures that the second check using $\mathsf{D}(\mathbb{x}, z, \rho)$ is consistent with the PCP verifier check $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho)$.

Consequently, our main challenge is how to convert this dv-SNARG to a publicly verifiable one. The key idea is to use a correlation-intractable hash family to "hide" the secret key of the verifier into a sparse relation.

Correlation-intractable hash instead of secret key. We now describe how to modify the construction to be publicly verifiable. The scheme follows the blueprint of Micali's construction [Kil92; Mic00]: On input an instance $\mathbb{x}$ and a witness $\mathbb{w}$, the SNARG prover $\mathcal{P}$ computes the PCP proof Π and commits to it using the VC scheme, obtaining a short commitment cm. It then derives the PCP verifier's randomness ρ via Fiat–Shamir, namely by computing $\rho := h(\mathsf{cm})$ (where h is the Fiat–Shamir hash function), and uses ρ to determine the PCP query locations. The prover outputs cm together with the VC openings to the queried positions of Π. The verifier $\mathcal{V}$ recomputes ρ, checks that all openings are valid with respect to cm, and then runs the PCP verifier on the opened symbols.

To prove Theorem 1, we instantiate this general construction and outline its soundness analysis.

Consider $\mathbb{x} \notin L(R)$. As mentioned at the beginning of this section, one challenge is that interactive arguments only have computational soundness and do not induce sparse relations to which we can apply CIH. However, since our function VC satisfies function statistical binding, we are able to leverage the statistical soundness of the underlying PCP. More formally, let F be the shadow algorithm that computes the PCP shadow and D be the decider. Assume that the SNARG adversary $\widetilde{\mathcal{P}}$ successfully fools the SNARG verifier $\mathcal{V}$, which means that $\widetilde{\mathcal{P}}$ outputs $(\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf})$ such that

$$\rho = h_{\mathsf{CI}}(\mathsf{cm}) \text{ and } \mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) = 1 \text{ and } \mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x}; \rho) = 1\,,$$

where h_{CI} is the CIH function and $\mathbf{V}$ is the PCP verifier. By the function statistical binding property of the function VC, since $\mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) = 1$, we can find $\widetilde{\Pi}$ such that

$$\mathsf{F}(\mathbb{x}, \widetilde{\Pi}; \gamma) = \mathsf{VC.Extract}(\mathsf{td}, \mathsf{cm}) \text{ and } \widetilde{\Pi}[\mathsf{Q}] = \mathsf{a}\,,$$

which implies that

$$\rho = h_{\mathsf{CI}}(\mathsf{cm}) \text{ and } \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 1\,.$$

This enables us to use PCP shadows to bound the SNARG soundness error.

Recall from Section 2.1 that when the shadow algorithm is randomized, its randomness γ must remain hidden from the prover; otherwise it is easy to choose a proof string that fools the decider. In the construction, γ is sampled at setup and determines the function $\mathsf{F}(\mathbb{x}, \cdot; \gamma)$ that the vector commitment binds to, and by function hiding the public parameters reveal nothing about it. Intuitively, a prover that causes the PCP verifier and the shadow decider to disagree must predict γ, and hence break function hiding. For our instantiation the shadow algorithm is deterministic, but we carry out the analysis for randomized shadows for the sake of generality.

To formalize the intuition above, we set $z := \mathsf{F}(\mathbb{x}, \widetilde{\Pi}; \gamma)$ and consider two cases: $\mathsf{D}(\mathbb{x}, z, \rho) = 0$ and $\mathsf{D}(\mathbb{x}, z, \rho) = 1$. Specifically, we argue that neither of the following events can happen with high probability:

$$\begin{bmatrix} \rho = h_{\mathsf{CI}}(\mathsf{cm}) \\ \wedge\, \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 1 \\ \wedge\, \mathsf{D}(\mathbb{x}, z, \rho) = 0 \end{bmatrix} \text{ and } \begin{bmatrix} \rho = h_{\mathsf{CI}}(\mathsf{cm}) \\ \wedge\, \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 1 \\ \wedge\, \mathsf{D}(\mathbb{x}, z, \rho) = 1 \end{bmatrix}.$$

The probability of the event on the left-hand side can be bounded by the correctness error of PCP shadows (the first property in Definition 1), which ensures that for any PCP verifier randomness ρ, the PCP verifier $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho)$ and the shadow decider $\mathsf{D}(\mathbb{x},z,\rho)$ agree except with small probability over shadow randomness γ. In other words, in order to find $\widetilde{\Pi}$ and ρ such that $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho)$ and $\mathsf{D}(\mathbb{x},z,\rho)$ disagree with high probability, one has to know the secret randomness γ of the shadow algorithm F. Therefore, if $\widetilde{\mathcal{P}}$ succeeds with high probability, we would be able to correctly guess γ using $\widetilde{\mathcal{P}}$ with high probability, which violates the function-hiding property of the function VC. We note that in the formal analysis in Section 6.2, we additionally rely on the consistency of PCP shadows (the third property in Definition 1).

The other event can be analyzed using the CIH. By the sparsity of PCP shadows (the second property in Definition 1), the following relation is sparse:

$$R_{\mathbb{x},\mathsf{td}} \coloneqq \{(\mathsf{cm},\rho) : \mathsf{D}(\mathbb{x},z,\rho)=1 \text{ where } z \coloneqq \mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\} \ .$$

Therefore, $\rho = h_{\mathsf{cl}}(\mathsf{cm})$ and $\mathsf{D}(\mathbb{x},z,\rho)=1$ can happen with probability at most the CIH error. CIH for all sparse relations is not known to follow from LWE alone: existing LWE-based constructions either require the relation to be computable by a bounded-size circuit [PS19], or they rely on relations with an inherent product structure [HLR21]. In our construction, we adopt the latter approach of [HLR21]: we amplify shadow soundness by parallel repetition of the shadow PCP, which naturally induces a sparse product relation (see Sections 4 and 6).

Plugging in the shadow PCP constructed in Section 2.1 completes the proof of Theorem 1: its shadow algorithm lies in NC, and a function VC for $\mathsf{P/poly}$ is in particular a function VC for NC.

On non-adaptive security. Our soundness guarantee is non-adaptive. We use the fact that the instance is fixed before the CRS is generated in two places. First, the function-hiding switch replaces the dummy function in the public parameters with the instance-dependent shadow algorithm $\mathsf{F}(\mathbb{x},\cdot)$. Second, the sparse relation $R_{\mathbb{x},\mathsf{td}}$ on which we invoke correlation intractability depends on $\mathbb{x}$, and correlation intractability holds only for relations fixed independently of the hash key. Obtaining adaptive soundness within this framework would require addressing both.

3 Preliminaries

Definition 3.1. *For every function $f\colon X \to Y$, the image of f, denoted as $\mathsf{Im}(f)$, is*

$$\mathsf{Im}(f) \coloneqq \{f(x) \in Y : x \in X\}\,.$$

Definition 3.2. *A* **relation** *R is a set of pairs $(\mathbb{x}, \mathbb{w})$ where $\mathbb{x}$ is an instance and $\mathbb{w}$ a witness. The corresponding* **language** *$L(R)$ is the set of instances $\mathbb{x}$ for which there exists a witness $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in R$.*

3.1 Correlation intractability

Definition 3.3. *For every relation ensemble $\mathbf{R} = \{R_\phi \subseteq \mathcal{X}_\phi \times \mathcal{W}_\phi\}$, we say that $\mathbf{R}$ is $\tau(\cdot)$-***sparse** *if for every ϕ and $x \in \mathcal{X}_\phi$,*

$$\Pr\left[(x, y) \in R_\phi \mid y \leftarrow \mathcal{W}_\phi\right] \leq \tau(\phi)\,.$$

Definition 3.4 (Correlation intractability)**.** *For every security parameter λ, let $\mathbf{R}_\lambda$ be a class of τ-sparse relations $R_\phi \subseteq \mathcal{X}_\lambda \times \mathcal{W}_\lambda$. A hash family $\mathcal{H} = \{\mathcal{H}_\lambda\colon \mathcal{X}_\lambda \to \mathcal{W}_\lambda\}_\lambda$ is* **correlation intractable for $\mathbf{R} = \{\mathbf{R}_\lambda\}_\lambda$** *if for every security parameter λ, $t_{\mathsf{CI}} \in \mathbb{N}$, and two-stage adversary $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ of total size at most t_{CI} such that $R_\phi \in \mathbf{R}_\lambda$ for every (ϕ, aux) in the support of $\mathsf{A}_0(1^\lambda)$,*

$$\Pr\left[(x, h(x)) \in R_\phi \;\middle|\; \begin{array}{l}(\phi, \mathsf{aux}) \leftarrow \mathsf{A}_0(1^\lambda)\\ h \leftarrow \mathcal{H}_\lambda\\ x \leftarrow \mathsf{A}_1(\mathsf{aux}, h)\end{array}\right] \leq \epsilon_{\mathsf{CI}}(\lambda, t_{\mathsf{CI}})\,.$$

Definition 3.5 (Product relation)**.** *A relation $R \subseteq \mathcal{X} \times \mathcal{W}^t$ is a product relation if for every $\mathbb{x}$, the set $R_\mathbb{x} \coloneqq \{\mathbb{w} : (\mathbb{x}, \mathbb{w}) \in R\}$ is the Cartesian product of sets $S_{\mathbb{x},1}, \ldots, S_{\mathbb{x},t}$:*

$$R_\mathbb{x} = S_{\mathbb{x},1} \times \cdots \times S_{\mathbb{x},t}\,.$$

Definition 3.6 (Efficient product verifiability)**.** *A product relation R is T-size product verifiable if there exists a size-T circuit C such that for every $\mathbb{x}$ and its corresponding $S_{\mathbb{x},1}, \ldots, S_{\mathbb{x},t}$, $\mathbb{w}_i \in S_{\mathbb{x},i}$ if and only if $C(\mathbb{x}, \mathbb{w}_i, i) = 1$.*

Definition 3.7 (Product sparsity)**.** *A product relation $R \subseteq \mathcal{X} \times \mathcal{W}^t$ has sparsity τ if for every $\mathbb{x}$ and its corresponding $S_{\mathbb{x},1}, \ldots, S_{\mathbb{x},t}$, $|S_{\mathbb{x},i}| \leq \tau\,|\mathcal{W}|$.*

Theorem 3.8 ([HLR21])**.** *For every λ, let $\mathbf{R}_\lambda$ be the class of all T-size product-verifiable relations $R \subseteq \mathcal{X} \times \mathcal{W}^{t_\lambda}$ with product sparsity τ, where $t_\lambda > \lambda/(1-\tau)$. There exists a hash family $\mathcal{H}_{\mathsf{CI}} = \{\mathcal{H}_{\mathsf{CI}}^{(\lambda)}\colon \mathcal{X} \to \mathcal{W}^{t_\lambda}\}_{\lambda\in\mathbb{N}}$ that is correlation intractable for $\mathbf{R} = \{\mathbf{R}_\lambda\}_{\lambda\in\mathbb{N}}$ under the hardness of LWE. Moreover, $\mathcal{H}_{\mathsf{CI}}$ depends only on $\mathcal{X}, \mathcal{W}, T, t_\lambda, \tau$ and can be evaluated in time $\mathsf{poly}(\log|\mathcal{X}|, t_\lambda, T)$.*

Since the hash family and error bound are uniform over $\mathbf{R}_\lambda$, conditioning on any (ϕ, aux) output by A_0, hardwiring aux into A_1, and averaging over A_0 reduces the definition to the fixed-relation case.

3.2 Non-interactive arguments

A *non-interactive argument* for a relation R (in the common reference string model) is a tuple of algorithms $\mathsf{SNARG} = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ satisfying the following properties.

Definition 3.9. $\mathsf{SNARG} = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *has* **perfect completeness** *if for every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, public parameter $\mathsf{pp} \in \mathsf{Setup}(1^\lambda, n)$, and instance-witness pair $(\mathbb{x}, \mathbb{w}) \in R$ with $|\mathbb{x}| \leq n$,*

$$\Pr\left[\mathcal{V}(\mathsf{pp}, \mathbb{x}, \pi) = 1 \mid \pi \leftarrow \mathcal{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w})\right] = 1\,.$$

Definition 3.10. $\mathsf{SNARG} = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *has* **non-adaptive soundness error** $\epsilon_{\mathsf{SNARG}}$ *if for every security parameter $\lambda \in \mathbb{N}$, instance $\mathbb{x} \notin L(R)$, circuit size bound $t_{\mathsf{SNARG}} \in \mathbb{N}$, and t_{SNARG}-size circuit $\widetilde{\mathcal{P}}$,*

$$\Pr\left[\mathcal{V}(\mathsf{pp}, \mathbb{x}, \pi) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, |\mathbb{x}|) \\ \pi \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}) \end{array}\right] \leq \epsilon_{\mathsf{SNARG}}(\lambda, \mathbb{x}, t_{\mathsf{SNARG}})\,.$$

Definition 3.11. $\mathsf{SNARG} = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *has* **adaptive soundness error** $\epsilon_{\mathsf{SNARG}}$ *if for every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, circuit size bound $t_{\mathsf{SNARG}} \in \mathbb{N}$, and t_{SNARG}-size circuit $\widetilde{\mathcal{P}}$,*

$$\Pr\left[\begin{array}{l} |\mathbb{x}| \leq n \\ \wedge\, \mathbb{x} \notin L(R) \\ \wedge\, \mathcal{V}(\mathsf{pp}, \mathbb{x}, \pi) = 1 \end{array} \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, n) \\ (\mathbb{x}, \pi) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}) \end{array}\right] \leq \epsilon_{\mathsf{SNARG}}(\lambda, n, t_{\mathsf{SNARG}})\,.$$

Theorem 3.12 ([WW25])**.** *Assume the existence of sub-exponential indistinguishability obfuscation (iO) for Boolean circuits and sub-exponential one-way functions (OWF). There exists a SNARG for any* NP *relation R that satisfies the following:*

- *proof size* $\mathsf{poly}(\lambda)$*;*
- *public parameter size* $\mathsf{poly}(\lambda, |R|)$*, where $|R|$ is the size of the Boolean circuit computing R;*
- *perfect completeness;*
- *adaptive soundness error* $\mathrm{negl}(\lambda)$*.*

3.3 Probabilistically checkable proofs

A *probabilistically checkable proof* (PCP) for a relation R is a pair of algorithms $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$, where $\mathbf{P}$ produces a proof string and the probabilistic verifier $\mathbf{V}$ has oracle access to it. For verifier randomness $\rho \in \{0,1\}^{\mathsf{r}}$, we write $\mathbf{V}^{\Pi}(\mathbb{x}; \rho)$ for the deterministic execution of the verifier with oracle Π. The verifier is *nonadaptive* if its query set is determined by $(\mathbb{x}, \rho)$ before it receives any oracle answers. For a query set Q and answers a, $\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x}; \rho)$ denotes the execution that answers queries in Q according to a and rejects if the verifier queries any position outside Q.

Definition 3.13 (Completeness)**.** $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *for a relation R has* **perfect completeness** *if, for every instance-witness pair $(\mathbb{x}, \mathbb{w}) \in R$,*

$$\Pr\left[\mathbf{V}^{\Pi}(\mathbb{x}; \rho) = 1 \;\middle|\; \Pi \leftarrow \mathbf{P}(\mathbb{x}, \mathbb{w}),\ \rho \leftarrow \{0,1\}^{\mathsf{r}}\right] = 1\,.$$

Definition 3.14 (Soundness)**.** $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *for a relation R has* **soundness error** ϵ_{PCP} *if, for every instance $\mathbb{x} \notin L(R)$ and (unbounded) circuit $\widetilde{\mathbf{P}}_{\mathsf{PCP}}$,*

$$\Pr\left[\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 1 \;\middle|\; \widetilde{\Pi} \leftarrow \widetilde{\mathbf{P}}_{\mathsf{PCP}},\ \rho \leftarrow \{0,1\}^{\mathsf{r}}\right] \leq \epsilon_{\mathsf{PCP}}(\mathbb{x})\,.$$

We consider several efficiency measures for a PCP:

- the *proof alphabet* Σ is the alphabet over which a PCP string is written;
- the *proof length* ℓ is the number of alphabet symbols in the PCP string;
- the *query complexity* $\mathsf{q} \in [\ell]$ is the number of queries that the PCP verifier makes to the PCP string (each query is an index in $[\ell]$ and is answered by the corresponding symbol in Σ in the PCP string);
- the *randomness complexity* r is the number of random bits used by the PCP verifier.

An efficiency measure may be a function of the instance $\mathbb{x}$ (e.g., of its size $|\mathbb{x}|$). In the statements below, n denotes the size of the instance.

Theorem 3.15 ([BS05; Din07; BMVY25])**.** *There is a* PCP *for* 3SAT *that satisfies the following:*

- *perfect completeness;*
- *soundness error* $\epsilon_{\mathsf{PCP}} = 1/2$*;*
- *non-adaptive verifier;*
- *query complexity* $\mathsf{q} = O(1)$*;*
- *randomness complexity* $\mathsf{r} = \log n + O(\log \log n)$*;*
- *proof length* $\ell = n \cdot \mathsf{polylog} n$*;*
- *alphabet* $\{0, 1\}$.

4 PCP shadows

We define and construct PCPs with *shadow soundness*. Informally, this notion requires that, given an instance $\mathbb{x}$ and a (malicious) PCP proof $\widetilde{\Pi}$, there exists an efficiently computable *shadow* z that succinctly represents the PCP verifier's behavior with respect to its randomness: namely, there exists a decider such that, given the instance $\mathbb{x}$ and the shadow z, it outputs the verifier's decision for any verifier randomness ρ with high probability. We then show that there exists a PCP for the canonical NP-complete language $\mathsf{3SAT}$ that satisfies shadow soundness, and whose shadow can be computed by *polynomial-size circuits*.

Definition 4.1 (Shadow soundness)**.** *A* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *has shadow soundness if there exists a function* F *with randomness complexity* r_{F} *and a deterministic algorithm* D *(a decider) such that for every instance* $\mathbb{x} \notin L(R)$*, the following holds:*

1. ***Correctness.*** *For any* $\widetilde{\Pi}$ *and verifier randomness* $\rho \in \{0,1\}^{\mathsf{r}}$*:*

$$\Pr\left[\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho) \neq \mathsf{D}(\mathbb{x},z,\rho) \;\middle|\; \begin{array}{l} \gamma \leftarrow \{0,1\}^{\mathsf{r}_{\mathsf{F}}} \\ z := \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma) \end{array}\right] \leq \epsilon_{\mathsf{c}}(\mathbb{x})\,.$$

2. ***Sparsity.*** *There exists an efficiently computable set* Z *where* $\mathsf{Im}(\mathsf{F}(\mathbb{x},\cdot)) \subseteq Z$ *such that for any shadow state* $z \in Z$*:*

$$\Pr\left[\mathsf{D}(\mathbb{x},z,\rho) = 1 \mid \rho \leftarrow \{0,1\}^{\mathsf{r}}\right] \leq \epsilon_{\mathsf{s}}(\mathbb{x})\,.$$

3. ***Consistency.*** *For every shadow randomness* $\gamma \in \{0,1\}^{\mathsf{r}_{\mathsf{F}}}$*, verifier randomness* $\rho \in \{0,1\}^{\mathsf{r}}$*,* $\widetilde{\Pi}$ *and* $\widetilde{\Pi}'$ *such that* $\widetilde{\Pi}[\mathsf{Q}] = \widetilde{\Pi}'[\mathsf{Q}]$*, where* Q *is the query set of the PCP verifier* $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho)$*:*

$$\mathsf{D}(\mathbb{x},z,\rho) = \mathsf{D}(\mathbb{x},z',\rho)\,,$$

where $z := \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)$ *and* $z' := \mathsf{F}(\mathbb{x},\widetilde{\Pi}';\gamma)$*.*

Moreover, we say that PCP *has shadow state size* s_{F} *if* F *outputs at most* s_{F} *bits. We use* t_{F} *and* t_{D} *to denote upper bounds on the sizes of* F *and* D*, respectively.*

In our construction of the SNARG (Section 6), we actually rely on a stronger sparsity property, called *product sparsity*.

Definition 4.2 (Product sparsity)**.** *We say* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *has shadow soundness with product sparsity with product parameter* t *if there exists a deterministic algorithm* D' *such that for every* $\mathbb{x}$*,* z*, and* $\rho = (\rho_1, \ldots, \rho_t)$ *where* $\rho_i \in \{0,1\}^{\frac{\mathsf{r}}{t}}$*,*

$$\mathsf{D}(\mathbb{x},z,\rho) = \bigwedge_{i\in[t]} \mathsf{D}'(\mathbb{x},z,\rho_i)\,.$$

Moreover, there exists an efficiently computable set Z *where* $\mathsf{Im}(\mathsf{F}(\mathbb{x},\cdot)) \subseteq Z$ *such that for any state* $z \in Z$ *and* $i \in [t]$*:*

$$\Pr\left[\mathsf{D}'(\mathbb{x},z,\rho_i) = 1 \;\middle|\; (\rho_1,\ldots,\rho_t) \leftarrow \{0,1\}^{t\cdot\frac{\mathsf{r}}{t}}\right] \leq \epsilon_{\mathsf{s}}(\mathbb{x})\,.$$

Remark 4.3. A PCP that satisfies product sparsity with error ϵ_{s} and product parameter t also satisfies sparsity with error ϵ_{s}^t. We use product sparsity because, in the SNARG construction in Section 6, we use correlation-intractable hash functions for sparse product relations (see Theorem 3.8). If there exist correlation-intractable hash functions for all sparse relations, it suffices to use plain sparsity.

4.1 Shadow PCP for NP

The theorem below describes the PCP we build to construct the SNARG in Section 6.

Theorem 4.4. *There is a* PCP *for* $\mathsf{3SAT}$ *that satisfies the following:*

- *perfect completeness;*
- *soundness error* $\epsilon_{\mathsf{PCP}} = 1/2$*;*
- *non-adaptive verifier;*
- *query complexity* $\mathsf{q} \leq \tilde{O}(\sqrt{n})$*;*
- *randomness complexity* $\mathsf{r} = \frac{1}{2}\log n + O(\log\log n)$*;*
- *proof length* $\ell = \tilde{O}(n)$*;*
- *alphabet* $\{0,1\}^{O(1)}$*;*
- *shadow soundness with:*
  1. *shadow state size* $\mathsf{s}_{\mathsf{F}} = \tilde{O}(\sqrt{n})$*;*
  2. *shadow algorithm circuit size* $\mathsf{t}_{\mathsf{F}} = \mathsf{poly}(n)$ *(with depth* $\mathsf{polylog}(n)$*);*
  3. *shadow randomness complexity* $\mathsf{r}_{\mathsf{F}} = 0$*;*
  4. *product sparsity error* $\epsilon_{\mathsf{s}} = 1/2$ *and product parameter* $t = 1$*;*
  5. *correctness error* $\epsilon_{\mathsf{c}} = 0$*;*

Proof. We start by describing the PCP construction, then we present and analyze the circuit that computes its shadow.

PCP construction. Let $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ be the PCP for $\mathsf{3SAT}$ in Theorem 3.15. Fix $B \in \mathbb{N}$. Set $N_{\mathsf{blk}} := 2^{\mathsf{r}}/B$. We break $\{0,1\}^{\mathsf{r}}$ into N_{blk} consecutive blocks of size B: $R_1, \ldots, R_{N_{\mathsf{blk}}}$. To prove our theorem, we choose B to be $2^{\mathsf{r}/2}$. $\mathsf{PCP}' = (\mathbf{P}', \mathbf{V}')$ works as follows:

- *Generating the proof string:* $\mathbf{P}'$ runs $\mathbf{P}(\mathbb{x}, \mathbb{w})$ to get the PCP proof Π and sends Π to $\mathbf{V}'$.
- *Verifier's checks:*
  1. Sample $b \leftarrow [N_{\mathsf{blk}}]$.
  2. For every $\rho \in R_b$: Check if $\mathbf{V}^{\Pi}(\mathbb{x}; \rho) = 1$.

We claim that the soundness error of PCP' is ϵ_{PCP}. Note that by soundness of PCP, there are at most $\epsilon_{\mathsf{PCP}} \cdot 2^{\mathsf{r}}$ accepting verifier random strings. Therefore, there are at most $\frac{\epsilon_{\mathsf{PCP}} \cdot 2^{\mathsf{r}}}{B} = \epsilon_{\mathsf{PCP}} \cdot N_{\mathsf{blk}}$ randomness blocks that contain only accepting verifier random strings. It follows that $\mathbf{V}'$ accepts with probability at most ϵ_{PCP}. Hence, PCP' satisfies the following:

- non-adaptive verifier;
- perfect completeness;
- soundness error ϵ_{PCP};
- proof length ℓ;
- randomness complexity $\log N_{\mathsf{blk}}$;
- query complexity $\mathsf{q} \cdot B$;
- alphabet Σ.

Shadow algorithm construction. The following circuit F gives shadow soundness for $\mathsf{3SAT}$:

$\mathsf{F}(\mathbb{x}, \widetilde{\Pi})$:

1. For every PCP verifier randomness $\rho \in \{0,1\}^{\mathsf{r}}$: let $v_\rho := \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho)$.
2. Initialize $z := 0^{N_{\mathsf{blk}}}$.
3. For every randomness block $j \in [N_{\mathsf{blk}}]$: let $z[j] := \bigwedge_{\rho \in R_j} v_\rho$.
4. Output z.

Fix $\mathbb{x}$. Since the base verifier is nonadaptive, for every ρ its $O(1)$ query locations can be hardwired, as can its Boolean predicate on the answers from the constant alphabet. All values v_ρ can therefore be evaluated in parallel. Computing each block conjunction with a balanced tree adds depth $O(\mathsf{r}) = O(\log n)$, and the total number of gates is polynomial in n. Thus, $\mathsf{F}(\mathbb{x}, \cdot)$ has polynomial-size, $O(\log n)$-depth circuits. Moreover, the shadow state size is $\sqrt{2^{\mathsf{r}}}$ if we choose $B := \sqrt{2^{\mathsf{r}}}$.

Correctness. Fix $\widetilde{\Pi}$ and PCP verifier randomness b. F computes exactly the new PCP verifier $\mathbf{V}'$.

Sparsity. Let

$$Z := \left\{ z \in \{0,1\}^{N_{\mathsf{blk}}} : |\{j \in [N_{\mathsf{blk}}] : z[j] = 1\}| \leq N_{\mathsf{blk}}/2 \right\} .$$

Membership in Z is efficiently computable. Since the soundness error of PCP' is $1/2$, every state in $\mathsf{Im}(\mathsf{F}(\mathbb{x}, \cdot))$ belongs to Z. Moreover, for every $z \in Z$, the decider accepts a uniformly sampled block index with probability at most $1/2$. Thus, the sparsity error is $1/2$.

Consistency. This follows directly from the structure of PCP'.

□

4.2 Amplification of shadow soundness

We prove a parallel repetition theorem for amplifying shadow soundness of PCPs.

Theorem 4.5. *Consider a* PCP *that satisfies the following:*

- *perfect completeness;*
- *soundness error* ϵ_{PCP}*;*
- *proof length* ℓ*;*
- *randomness complexity* r*;*
- *query complexity* q*;*
- *alphabet* Σ*;*
- *shadow soundness with:*
  1. *shadow state size* s_{F}*;*
  2. *shadow algorithm circuit size* t_{F}*;*
  3. *shadow algorithm circuit depth* d_{circ}*;*
  4. *shadow randomness complexity* r_{F}*;*
  5. *sparsity error* ϵ_{s}*;*
  6. *correctness error* ϵ_{c}*.*

Let $\mathsf{PCP}_t := \mathsf{PR}[\mathsf{PCP}, t]$ *be the* t*-wise parallel repetition of* PCP *specified in Theorem 4.7.* PCP_t *satisfies the following:*

- *perfect completeness;*
- *soundness error* $\epsilon_{\mathsf{PCP}}^t$*;*
- *proof length* ℓ*;*
- *randomness complexity* $t \cdot \mathsf{r}$*;*

- *query complexity* $t \cdot \mathsf{q}$*;*
- *alphabet* Σ*;*
- *shadow soundness with:*
  1. *shadow state size* s_{F}*;*
  2. *shadow algorithm circuit size* t_{F}*;*
  3. *shadow algorithm circuit depth* d_{circ}*;*
  4. *shadow randomness complexity* r_{F}*;*
  5. *product sparsity error* ϵ_{s} *and product parameter* t*;*
  6. *correctness error* $2t \cdot \epsilon_{\mathsf{c}}$*.*

Corollary 4.6. *There is a* PCP *for* 3SAT *that satisfies the following:*

- *perfect completeness;*
- *soundness error* $\epsilon_{\mathsf{PCP}} = 2^{-(\log n)^2}$*;*
- *non-adaptive verifier;*
- *query complexity* $\mathsf{q} \leq \tilde{O}(\sqrt{n})$*;*
- *randomness complexity* $\mathsf{r} \leq \tilde{O}((\log n)^3)$*;*
- *proof length* $\ell = \tilde{O}(n)$*;*
- *alphabet* $\{0,1\}^{O(1)}$*;*
- *shadow soundness with:*
  1. *shadow state size* $\mathsf{s}_{\mathsf{F}} = \tilde{O}(\sqrt{n})$*;*
  2. *shadow algorithm circuit size* $\mathsf{t}_{\mathsf{F}} = \mathsf{poly}(n)$ *with depth* $\mathsf{polylog}(n)$*;*
  3. *shadow randomness complexity* $\mathsf{r}_{\mathsf{F}} = 0$*;*
  4. *product sparsity error* $\epsilon_{\mathsf{s}} = 1/2$ *and product parameter* $t = (\log n)^2$*;*
  5. *correctness error* $\epsilon_{\mathsf{c}} = 0$*;*

Construction 4.7. Consider $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$. We first recall how the t-wise parallel repetition $\mathsf{PCP}_t = (\vec{\mathbf{P}}, \vec{\mathbf{V}})$ works:

- **Generating the PCP proof**: $\vec{\mathbf{P}}$ computes $\Pi \leftarrow \mathbf{P}(\mathbb{x}, \mathbb{w})$ and sends Π to $\vec{\mathbf{V}}$.
- **Verifier's checks**:
  1. $\vec{\mathbf{V}}$ samples $(\rho_1, \ldots, \rho_t) \leftarrow \{0,1\}^{t \cdot \mathsf{r}}$.
  2. $\vec{\mathbf{V}}$ accepts iff for every $i \in [t]$, $\mathbf{V}^{\Pi}(\mathbb{x}; \rho_i) = 1$.

Proof of Theorem 4.5. **Perfect completeness.** Perfect completeness follows because the repeated verifier accepts if every invocation of the perfectly complete base verifier accepts.

Soundness. We show that the soundness error of PCP_t is $\epsilon_{\mathsf{PCP}}^t$. Let $\widetilde{\Pi}$ be a malicious PCP proof string. We bound the following probability:

$$\begin{aligned}
&\Pr\left[\vec{\mathbf{V}}^{\widetilde{\Pi}}(\mathbb{x}; \rho_1, \ldots, \rho_t) = 1 \;\middle|\; (\rho_1, \ldots, \rho_t) \leftarrow \{0,1\}^{t \cdot \mathsf{r}}\right] \\
&= \left(\Pr\left[\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 1 \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{r}}\right]\right)^t \\
&\leq \epsilon_{\mathsf{PCP}}^t \,.
\end{aligned}$$

Shadow algorithm. Let T be the shadow algorithm for PCP and D be the decider. We construct $\mathsf{T}_t, \mathsf{D}'_t, \mathsf{D}_t$ for PCP_t as follows:

- $\mathsf{T}_t(\mathbb{x}, \Pi; \gamma)$: Output $z \coloneqq \mathsf{T}(\mathbb{x}, \Pi; \gamma)$.
- $\mathsf{D}'_t(\mathbb{x}, z, \rho_i)$: Output $\mathsf{D}(\mathbb{x}, z, \rho_i)$.
- $\mathsf{D}_t(\mathbb{x}, z, (\rho_1, \ldots, \rho_t))$: Output $\bigwedge_{i \in [t]} \mathsf{D}'_t(\mathbb{x}, z, \rho_i)$.

Thus, T_t has the same circuit size and depth as T. Let Z be the efficiently computable set from the shadow-soundness guarantee for PCP. Since $\mathsf{T}_t = \mathsf{T}$, we have $\mathsf{Im}(\mathsf{T}_t(\mathbb{x}, \cdot)) \subseteq Z$.

We first analyze the product sparsity. Fix $z \in Z$. For every $i \in [t]$, we bound the following probability:

$$\Pr\left[\mathsf{D}'_t(\mathbb{x}, z, \rho_i) = 1 \;\middle|\; (\rho_1, \ldots, \rho_t) \leftarrow \{0,1\}^{t \cdot \mathsf{r}}\right] \leq \epsilon_{\mathsf{s}}\,.$$

Now we analyze correctness. Fix $\widetilde{\Pi}$ and verifier randomness $(\rho_1, \ldots, \rho_t) \leftarrow \{0,1\}^{t \cdot \mathsf{r}}$. We bound the following probability:

$$\begin{aligned}
&\Pr\left[\vec{\mathbf{V}}^{\widetilde{\Pi}}(\mathbb{x}; \rho_1, \ldots, \rho_t) \neq \mathsf{D}_t(\mathbb{x}, z, (\rho_1, \ldots, \rho_t)) \;\middle|\; \begin{array}{l} \gamma \leftarrow \{0,1\}^{\mathsf{r}_\mathsf{T}} \\ z \coloneqq \mathsf{T}_t(\mathbb{x}, \widetilde{\Pi}; \gamma) \end{array}\right] \\
&= \Pr\left[\begin{array}{l} \forall i \in [t], \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho_i) = 1 \\ \wedge\, \exists i \in [t], \mathsf{D}(\mathbb{x}, z, \rho_i) = 0 \end{array} \;\middle|\; \begin{array}{l} \gamma \leftarrow \{0,1\}^{\mathsf{r}_\mathsf{T}} \\ z \coloneqq \mathsf{T}(\mathbb{x}, \widetilde{\Pi}; \gamma) \end{array}\right] \\
&\quad + \Pr\left[\begin{array}{l} \exists i \in [t], \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho_i) = 0 \\ \wedge\, \forall i \in [t], \mathsf{D}(\mathbb{x}, z, \rho_i) = 1 \end{array} \;\middle|\; \begin{array}{l} \gamma \leftarrow \{0,1\}^{\mathsf{r}_\mathsf{T}} \\ z \coloneqq \mathsf{T}(\mathbb{x}, \widetilde{\Pi}; \gamma) \end{array}\right] \\
&\leq 2t \cdot \epsilon_{\mathsf{c}}\,.
\end{aligned}$$

Consistency. Fix shadow randomness γ, verifier randomness $(\rho_1, \ldots, \rho_t)$, and proof strings $\widetilde{\Pi}, \widetilde{\Pi}'$ that agree on the repeated verifier's query set. They agree on the query set for every ρ_i, so consistency of the base PCP implies that the corresponding base decider outputs agree for every $i \in [t]$. Taking the conjunction over all i proves consistency of D_t.

□

5 Function vector commitment scheme

We introduce a new notion of vector commitment, called a *function vector commitment (VC) scheme*, that allows one to commit to a vector m relative to a function f. Specifically, *(statistical) binding* requires that for every $m \neq m'$ such that $f(m) \neq f(m')$, the corresponding commitments cm and cm' are distinct. In fact, we enforce a stronger property, called *function statistical binding*, which requires that, given a trapdoor and a commitment, a deterministic extractor returns a value y such that every accepting local opening is consistent with some message m satisfying $f(m) = y$.

We begin by formally defining function VC in Section 5.1. Then, in Section A, we construct a function VC for polynomial-size circuits.

5.1 Definition

A *function vector commitment (VC) scheme* over alphabet Σ for a family of functions $\mathcal{F}$ is a tuple of algorithms

$$\mathsf{VC} = (\mathsf{VC.Gen}, \mathsf{VC.Commit}, \mathsf{VC.Open}, \mathsf{VC.Check}, \mathsf{VC.Extract})$$

with the following syntax:

- $\mathsf{VC.Gen}(1^\lambda, f) \to (\mathsf{pp}, \mathsf{td})$: On input a security parameter $\lambda \in \mathbb{N}$ and a description of a function f, $\mathsf{VC.Gen}$ samples a public parameter pp and a corresponding trapdoor td.
- $\mathsf{VC.Commit}(\mathsf{pp}, m) \to (\mathsf{cm}, \mathsf{aux})$: On input a public parameter pp and a message m, $\mathsf{VC.Commit}$ produces a commitment cm and the corresponding auxiliary state aux.
- $\mathsf{VC.Open}(\mathsf{pp}, \mathsf{aux}, \mathsf{Q}) \to \mathsf{pf}$: On input a public parameter pp, an auxiliary state aux, and a query set $\mathsf{Q} \subseteq [\ell]$, $\mathsf{VC.Open}$ outputs an opening proof string pf attesting that $m[\mathsf{Q}]$ is a restriction of m to Q.
- $\mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) \to \{0, 1\}$: On input a public parameter pp, a commitment cm, a query set $\mathsf{Q} \subseteq [\ell]$, an answer string $\mathsf{a} \in \Sigma^{\mathsf{Q}}$, and an opening proof string pf, $\mathsf{VC.Check}$ determines if pf is a valid proof for $\mathsf{a} \in \Sigma^{\mathsf{Q}}$ being a restriction of the message committed in cm to Q.
- $\mathsf{VC.Extract}(\mathsf{td}, \mathsf{cm}) \to y$: On input the trapdoor td and a commitment cm, the extractor $\mathsf{VC.Extract}$ deterministically outputs a string y.

Moreover, VC has the following properties.

Definition 5.1 (Completeness)**.** VC *has* **perfect completeness** *if for every security parameter $\lambda \in \mathbb{N}$, message $m \in \Sigma^\ell$, auxiliary function $f \in \mathcal{F}$, and query set $\mathsf{Q} \subseteq [\ell]$,*

$$\Pr\left[\mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, m[\mathsf{Q}], \mathsf{pf}) = 1 \;\middle|\; \begin{array}{l} (\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f) \\ (\mathsf{cm}, \mathsf{aux}) \leftarrow \mathsf{VC.Commit}(\mathsf{pp}, m) \\ \mathsf{pf} \leftarrow \mathsf{VC.Open}(\mathsf{pp}, \mathsf{aux}, \mathsf{Q}) \end{array}\right] = 1\,.$$

Definition 5.2 (Function hiding)**.** VC *is* **function hiding** *if for every security parameter $\lambda \in \mathbb{N}$, auxiliary functions $f, f' \in \mathcal{F}$, and t_{vc}-size adversary A_{vc},*

$$\left|\Pr\left[A_{\mathsf{vc}}(\mathsf{pp}) = 1 \;\middle|\; (\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f)\right] - \Pr\left[A_{\mathsf{vc}}(\mathsf{pp}) = 1 \;\middle|\; (\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f')\right]\right| \leq z_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})\,.$$

Definition 5.3 (Function statistical binding)**.** VC *satisfies* **function statistical binding** *if for every security parameter* $\lambda \in \mathbb{N}$*, auxiliary function* $f \in \mathcal{F}$*, adversary size* $t_{\mathsf{vc}} \in \mathbb{N}$*, and* t_{vc}*-size adversary* A_{vc}*, the following holds:*

$$\Pr\left[\begin{array}{l|l} \mathsf{VC.Check}(\mathsf{pp},\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}) = 1 & (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f) \\ \wedge \forall\, \widetilde{m} \in \Sigma^\ell : & (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}) \coloneqq A_{\mathsf{vc}}(\mathsf{pp}) \\ \quad (\widetilde{m}[\mathsf{Q}] \neq \mathsf{a} \ \vee\ f(\widetilde{m}) \neq y) & y \coloneqq \mathsf{VC.Extract}(\mathsf{td},\mathsf{cm}) \end{array}\right] \leq \kappa_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})\,.$$

The following lemma states that function hiding prevents any efficient adversary from guessing the randomness used to select the hidden function with probability significantly better than random guessing. We use this property in the analysis of SNARG soundness in Section 6.2.

Lemma 5.4 (Randomness hiding)**.** *Fix* VC*. For every* $\lambda \in \mathbb{N}$*, randomized function* f*, adversary size* $t_{\mathsf{vc}} \in \mathbb{N}$*, and* t_{vc}*-time adversary* A*, the following holds:*

$$\Pr\left[\rho' = \rho \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^r \\ (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f_\rho) \\ \rho' \leftarrow A(\mathsf{pp}) \end{array}\right] \leq \frac{1}{2^r} + z_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})\,.$$

Proof. Fix $\rho_0 \coloneqq 0^r$ and let $g \coloneqq f_{\rho_0}$ be a fixed function. For every fixed $\bar{\rho} \in \{0,1\}^r$, define

$$p_{\bar{\rho}} \coloneqq \Pr\left[\rho' = \bar{\rho} \;\middle|\; \begin{array}{l} (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f_{\bar{\rho}}) \\ \rho' \leftarrow A(\mathsf{pp}) \end{array}\right],$$
$$q_{\bar{\rho}} \coloneqq \Pr\left[\rho' = \bar{\rho} \;\middle|\; \begin{array}{l} (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, g) \\ \rho' \leftarrow A(\mathsf{pp}) \end{array}\right].$$

Consider the function-hiding distinguisher that has $\bar{\rho}$ hardwired, runs $A(\mathsf{pp})$ to obtain ρ', and outputs 1 if and only if $\rho' = \bar{\rho}$. Function hiding between $f_{\bar{\rho}}$ and g gives

$$p_{\bar{\rho}} \leq q_{\bar{\rho}} + z_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})$$

for every $\bar{\rho}$, where the equality-test overhead is absorbed into t_{vc}. Averaging these inequalities over $\bar{\rho} \leftarrow \{0,1\}^r$ gives

$$\begin{aligned} &\Pr\left[\rho' = \rho \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^r \\ (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f_\rho) \\ \rho' \leftarrow A(\mathsf{pp}) \end{array}\right] \\ &= 2^{-r} \sum_{\bar{\rho} \in \{0,1\}^r} p_{\bar{\rho}} \\ &\leq 2^{-r} \sum_{\bar{\rho} \in \{0,1\}^r} q_{\bar{\rho}} + z_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}}) \\ &\leq 2^{-r} + z_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})\,. \end{aligned}$$

The last inequality holds because the distribution of pp in the definition of $q_{\bar{\rho}}$ is independent of $\bar{\rho}$, and the events $\rho' = \bar{\rho}$ are pairwise disjoint. □

6 SNARG from function VC and shadow PCP

Theorem 6.1. *Fix a family of functions $\mathcal{F}$. Let* PCP *be a shadow PCP for* 3SAT *with a shadow algorithm in $\mathcal{F}$ that satisfies the following:*

- *perfect completeness;*
- *query complexity* q *and verifier circuit size* $\mathsf{vt}_{\mathsf{PCP}}$*;*
- *shadow soundness with:*
  1. *shadow state size* s_{F}*;*
  2. *shadow algorithm circuit size* t_{F} *and depth* d_{circ}*;*
  3. *shadow decider circuit size* t_{D}*;*
  4. *shadow randomness complexity* r_{F}*;*
  5. *product sparsity error* ϵ_{s} *and product parameter* t*;*
  6. *correctness error* $\epsilon_{\mathsf{c}}(\mathbb{x})$*.*

Assume that $\mathcal{F}$ contains the circuit f_0 that outputs $0^{\mathsf{s}_{\mathsf{F}}}$ on every input and is padded to the common description length of $\mathcal{F}$. Let VC *be a function VC for $\mathcal{F}$ that satisfies the following:*

- *perfect completeness;*
- *the function hiding error* $z_{\mathsf{VC}} = z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}})$*;*
- *function statistical binding error* $\kappa_{\mathsf{VC}} = \kappa_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}})$*;*
- *generation time* $t_{\mathsf{VC.Gen}}$*;*
- *verification circuit size* $t_{\mathsf{VC.Check}}$*;*
- *commitment size* s_{cm}*;*
- *public parameter size* s_{pp}*;*
- *proof size* s_{pf}*.*

Let $\mathcal{H}_{\mathsf{CI}}$ be a correlation-intractable hash family for sparse product relations that satisfies the following:

- *error* $\epsilon_{\mathsf{CI}} = \epsilon_{\mathsf{CI}}(\lambda, t_{\mathsf{CI}})$*;*
- *hash key size* s_{CI}*, key-sampling circuit size* $t_{\mathcal{I}}$*, and evaluation circuit size* t_h*.*

Let SNARG *be the argument constructed in Section 6.1. For every $\lambda \in \mathbb{N}$, instance $\mathbb{x} \notin L(R)$, and circuit size bound $t_{\mathsf{SNARG}} \in \mathbb{N}$, the non-adaptive soundness error* $\epsilon_{\mathsf{SNARG}}$ *of* SNARG *satisfies:*

$$\epsilon_{\mathsf{SNARG}}(\lambda, \mathbb{x}, t_{\mathsf{SNARG}}) \leq z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) + \kappa_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) + \epsilon_{\mathsf{CI}}(\lambda, t_{\mathsf{CI}}) + \epsilon_{\mathsf{c}}(\mathbb{x}) + \epsilon_{\mathsf{c}}(\mathbb{x}) \cdot 2^{\mathsf{r}_{\mathsf{F}}} \cdot z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \,,$$

where

$$\begin{aligned} t_{\mathsf{CI}} &= O(t_{\mathsf{SNARG}} + t_{\mathsf{VC.Gen}} + \mathsf{t}_{\mathsf{F}} + \mathsf{r}_{\mathsf{F}} + n), \\ t_{\mathsf{VC}} &= O\left(t_{\mathsf{SNARG}} + t_{\mathcal{I}} + t_h + t_{\mathsf{VC.Check}} + \mathsf{vt}_{\mathsf{PCP}} + 2^{\mathsf{r}_{\mathsf{F}}}(\mathsf{t}_{\mathsf{F}} + \mathsf{t}_{\mathsf{D}})\right) . \end{aligned}$$

The proof size as *of* SNARG *satisfies:*

$$\mathsf{as} \leq s_{\mathsf{cm}} + \mathsf{q} \cdot \log \ell + \mathsf{q} \cdot \log |\Sigma| + \mathsf{s}_{\mathsf{pf}} \,.$$

The public parameter size s_{pp} *of* SNARG *satisfies:*

$$s_{\mathsf{pp}} \leq s_{\mathsf{pp}} + s_{\mathsf{CI}} \,.$$

Remark 6.2. The construction only requires leveled FHE. For an instance-size bound n, the circuit computing the shadow algorithm in Theorem 4.6 has size $\mathsf{poly}(n)$ and depth $\mathsf{polylog}(n)$. The depth-universal circuit of [CH85] therefore has depth $D = \mathsf{polylog}(n)$. This bound is fixed before key generation and is the same for every circuit in $\mathcal{F}$.

Corollary 6.3. *Assume iO and LWE are sub-exponentially hard. For every $t_{\mathsf{SNARG}} = \mathsf{poly}(n)$ for which the reduction sizes in Theorem 6.1 lie within the security ranges of Theorems A.1 and 3.8, the Micali SNARG has non-adaptive soundness error $\epsilon_{\mathsf{SNARG}} = \mathrm{negl}(\lambda)$, proof size $\tilde{O}(\sqrt{n}) \cdot \mathsf{poly}(\lambda)$, and public parameter size $\mathsf{poly}(\lambda, n)$.*

Proof. We apply Theorem 6.1 with the PCP from Theorem 4.6, the VC from Theorem A.1, and the CIH from Theorem 3.8. Set $\lambda_{\mathsf{FHE}} \coloneqq \lambda$ and $\lambda_{\mathsf{SNARG}} \coloneqq \lambda$.

Soundness error. By the size premise in the corollary, $t_{\mathsf{VC}} < 2^{\lambda_{\mathsf{FHE}}^{1/c}}$ for some $c > 1$. Since t_{VC} includes the polynomial-size circuit computing the shadow algorithm, $\log n = \mathsf{poly}(\lambda_{\mathsf{FHE}})$. Thus, the bound $D = \mathsf{polylog}(n)$ from Theorem 6.2 satisfies $D = \mathsf{poly}(\lambda_{\mathsf{FHE}})$, as required by Theorem A.1. Moreover, $z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) = \mathrm{negl}(\lambda)$ and $\kappa_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) = \mathrm{negl}(\lambda)$. By the efficiency of $\mathsf{VC.Gen}$ and the shadow-PCP bounds, $t_{\mathsf{CI}} = \mathsf{poly}(\lambda, n)$, and the corollary's premise gives $\epsilon_{\mathsf{CI}}(\lambda, t_{\mathsf{CI}}) = \mathrm{negl}(\lambda)$. Also, $\epsilon_{\mathsf{c}}(\mathbb{x}) = 0$.

Proof size. We bound the size of cm, (Q, a), and pf separately.

- *Commitment size:* Note that the PCP shadow has state size $\tilde{O}(\sqrt{n})$. According to Theorem A.1, the size of the commitment cm is $\tilde{O}(\sqrt{n}) \cdot \mathsf{poly}(\lambda_{\mathsf{FHE}}, D) = \tilde{O}(\sqrt{n}) \cdot \mathsf{poly}(\lambda)$.
- *Query and answer size:* PCP has proof length $\ell = \tilde{O}(n)$, query complexity $\mathsf{q} \leq \tilde{O}(\sqrt{n})$, and alphabet $\{0,1\}^{O(1)}$. Hence, we need $O(\mathsf{q} \cdot \log \ell) = \tilde{O}(\sqrt{n})$ bits to represent Q and $O(\mathsf{q}) = \tilde{O}(\sqrt{n})$ bits to represent a. In total, $|\mathsf{Q}| + |\mathsf{a}| = \tilde{O}(\sqrt{n})$.
- *Opening proof size:* According to Theorem A.1, the size of pf is $\mathsf{poly}(\lambda_{\mathsf{SNARG}}) = \mathsf{poly}(\log n, \lambda)$.

Therefore, the proof size is $\tilde{O}(\sqrt{n}) \cdot \mathsf{poly}(\lambda)$.

Public parameter size. The size of the public parameter is dominated by the public parameter of VC, which is $\mathsf{poly}(\lambda, n, D) = \mathsf{poly}(\lambda, n)$ by Theorems A.1 and 6.2.

□

6.1 SNARG construction

We construct $\mathsf{SNARG} = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ as follows:

- $\mathsf{Setup}(1^\lambda, n)$:
  1. Sample $(\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f_0)$.
  2. Sample $h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)}$.
  3. Output $\mathsf{pp} \coloneqq (\mathsf{pp}, h_{\mathsf{CI}})$.
- $\mathcal{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w})$:
  1. Parse pp as $(\mathsf{pp}, h_{\mathsf{CI}})$.
  2. Compute $\Pi \leftarrow \mathbf{P}(\mathbb{x}, \mathbb{w})$.
  3. Compute $(\mathsf{cm}, \mathsf{aux}) \coloneqq \mathsf{VC.Commit}(\mathsf{pp}, \Pi)$.

4. Compute $\rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm})$.
5. Compute the PCP verifier query set Q using $\mathbb{x}$ and ρ.
6. Compute $\mathsf{pf} \coloneqq \mathsf{VC.Open}(\mathsf{pp}, \mathsf{aux}, \mathsf{Q})$ and $\mathsf{a} \coloneqq \Pi[\mathsf{Q}]$.
7. Output $\pi \coloneqq (\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf})$.

- $\mathcal{V}(\mathsf{pp}, \mathbb{x}, \pi)$:
  1. Parse pp as $(\mathsf{pp}, h_{\mathsf{CI}})$.
  2. Parse π as $(\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf})$.
  3. Compute $\rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm})$.
  4. Check the following:
     - $\mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) = 1$;
     - $\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x}; \rho) = 1$.
  5. If any of the above checks fails, output 0; otherwise, output 1.

6.2 SNARG soundness

Fix $\mathbb{x} \notin L(R)$. Let F and D be the shadow algorithm and decider, respectively. Let $\widetilde{\mathcal{P}}$ be an adversary for the SNARG of size t_{SNARG}. Our goal is to bound the following probability:

$$\Pr\left[\mathcal{V}(\mathsf{pp}, \mathbb{x}, \pi) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, n) \\ \pi \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}) \end{array}\right].$$

By definition of the construction, the above can be written as

$$\Pr\left[\begin{array}{l} \mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) = 1 \\ \wedge\, \mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x}; \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} (\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f_0) \\ h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)} \\ (\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}}) \\ \rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm}) \end{array}\right].$$

Consider the following adversary A_{VC} for function hiding of VC:

$A_{\mathsf{VC}}(\mathsf{pp})$:
1. Sample $h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)}$.
2. Compute $(\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}})$.
3. Compute $\rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm})$.
4. Output 1 iff the following checks pass:
   (a) $\mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) = 1$;
   (b) $\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x}; \rho) = 1$.

The size of A_{VC} is at most t_{VC} as defined in Theorem 6.1. By function hiding of VC, the following holds:

$$\begin{aligned} &\Pr\left[\begin{array}{l} \mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) = 1 \\ \wedge\, \mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x}; \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} (\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f_0) \\ h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)} \\ (\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}}) \\ \rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm}) \end{array}\right] \\ &= \Pr\left[A_{\mathsf{VC}}(\mathsf{pp}) = 1 \;\middle|\; (\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, f_0)\right] \end{aligned}$$

$$\leq \Pr\left[A_{\mathsf{vc}}(\mathsf{pp})=1 \;\middle|\; \begin{array}{l}\gamma \leftarrow \{0,1\}^{\mathsf{r_F}}\\ (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, \mathsf{F}(\mathbb{x},\cdot;\gamma))\end{array}\right] + z_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})$$

$$= \Pr\left[\begin{array}{l}\mathsf{VC.Check}(\mathsf{pp},\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf})=1\\ \wedge\, \mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\end{array} \;\middle|\; \begin{array}{l}\gamma \leftarrow \{0,1\}^{\mathsf{r_F}}\\ (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, \mathsf{F}(\mathbb{x},\cdot;\gamma))\\ h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)}\\ (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}})\\ \rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm})\end{array}\right] + z_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})\,.$$

Using Theorem 5.3, the following upper bounds the probability above:

$$\Pr\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\, \exists\, \widetilde{\Pi} : \widetilde{\Pi}[\mathsf{Q}]=\mathsf{a} \;\wedge\; \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\end{array} \;\middle|\; \begin{array}{l}\gamma \leftarrow \{0,1\}^{\mathsf{r_F}}\\ (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, \mathsf{F}(\mathbb{x},\cdot;\gamma))\\ h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)}\\ (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}})\\ \rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm})\\ z \coloneqq \mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\end{array}\right] + \kappa_{\mathsf{vc}}(\lambda, t_{\mathsf{vc}})\,.$$

Let us define the experiment above as Exp. Formally, we have:

$$\mathsf{Exp} \coloneqq \left[\begin{array}{l}\gamma \leftarrow \{0,1\}^{\mathsf{r_F}}\\ (\mathsf{pp},\mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, \mathsf{F}(\mathbb{x},\cdot;\gamma))\\ h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)}\\ (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}})\\ \rho = (\rho_1,\ldots,\rho_t) \coloneqq h_{\mathsf{CI}}(\mathsf{cm})\\ z \coloneqq \mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\end{array}\right].$$

By the law of total probability, the above probability is equal to:

$$\Pr_{\mathsf{Exp}}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\, \exists\, \widetilde{\Pi} : \widetilde{\Pi}[\mathsf{Q}]=\mathsf{a} \;\wedge\; \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\, \mathsf{D}(\mathbb{x},z,\rho)=1\end{array}\right] + \Pr_{\mathsf{Exp}}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\, \exists\, \widetilde{\Pi} : \widetilde{\Pi}[\mathsf{Q}]=\mathsf{a} \;\wedge\; \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\, \mathsf{D}(\mathbb{x},z,\rho)=0\end{array}\right].$$

We use the security of the CIH and the function VC to bound the above probabilities, respectively.

6.2.1 Security from CIH

Claim 6.4. *Let ϵ_{CI} be the error for the CIH family $\mathcal{H}_{\mathsf{CI}}$, and let t_{CI} be as defined in Theorem 6.1. The following holds:*

$$\Pr_{\mathsf{Exp}}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\, \exists\, \widetilde{\Pi} : \widetilde{\Pi}[\mathsf{Q}]=\mathsf{a} \;\wedge\; \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\, \mathsf{D}(\mathbb{x},z,\rho)=1\end{array}\right] \leq \epsilon_{\mathsf{CI}}(\lambda, t_{\mathsf{CI}})\,.$$

Proof. By the product-sparsity property of the PCP's shadow soundness (Theorem 4.2), there exists a deterministic algorithm D' such that for every $\mathbb{x}$, z, and $\rho = (\rho_1,\ldots,\rho_t) \in \{0,1\}^{t\cdot\frac{\mathsf{r}}{t}}$,

$$\mathsf{D}(\mathbb{x},z,\rho) = \bigwedge_{i\in[t]} \mathsf{D}'(\mathbb{x},z,\rho_i)\,.$$

For every trapdoor td sampled by $\mathsf{VC.Gen}$, we define a sparse relation $R_{\mathbb{x},\mathsf{td}}$:

$$R_{\mathbb{x},\mathsf{td}} \coloneqq \left\{ (\mathsf{cm}, \rho) : \begin{array}{l} z \coloneqq \mathsf{VC.Extract}(\mathsf{td}, \mathsf{cm}) \\ z \in Z \\ \forall i \in [t], \mathsf{D}'(\mathbb{x}, z, \rho_i) = 1 \end{array} \right\} .$$

We index this relation by $\phi \coloneqq (\mathbb{x}, \mathsf{td})$, so that $R_\phi = R_{\mathbb{x},\mathsf{td}}$. We show that for any $\mathbb{x} \notin L(R)$, the relation $R_{\mathbb{x},\mathsf{td}}$ is a ϵ_{s}-sparse product relation over the entire commitment domain. Fix any commitment cm, and let

$$z_{\mathsf{cm}} \coloneqq \mathsf{VC.Extract}(\mathsf{td}, \mathsf{cm}) .$$

For every $i \in [t]$, define the coordinate set

$$S_{\mathsf{cm},i} \coloneqq \begin{cases} \left\{ \rho_i \in \{0,1\}^{\mathsf{r}/t} : \mathsf{D}'(\mathbb{x}, z_{\mathsf{cm}}, \rho_i) = 1 \right\}, & \text{if } z_{\mathsf{cm}} \in Z, \\ \emptyset, & \text{otherwise.} \end{cases}$$

The witness set associated with cm is

$$\{\rho \in \{0,1\}^{\mathsf{r}} : (\mathsf{cm}, \rho) \in R_{\mathbb{x},\mathsf{td}}\} = \prod_{i \in [t]} S_{\mathsf{cm},i} .$$

Indeed, if $z_{\mathsf{cm}} \notin Z$, both sides are empty. Otherwise, the equality follows directly from the coordinate-wise acceptance condition in the definition of $R_{\mathbb{x},\mathsf{td}}$. Moreover, for every commitment cm and every $i \in [t]$,

$$\frac{|S_{\mathsf{cm},i}|}{2^{\mathsf{r}/t}} \leq \epsilon_{\mathsf{s}} .$$

When $z_{\mathsf{cm}} \notin Z$, the left-hand side is zero; otherwise, the inequality is exactly the product-sparsity guarantee in Theorem 4.2. Thus $R_{\mathbb{x},\mathsf{td}}$ is a ϵ_{s}-sparse product relation.

It is also T-size product verifiable: for fixed $\phi = (\mathbb{x}, \mathsf{td})$, the coordinate verifier computes $z \coloneqq \mathsf{VC.Extract}(\mathsf{td}, \mathsf{cm})$ and accepts (cm, ρ_i, i) if and only if $z \in Z$ and $\mathsf{D}'(\mathbb{x}, z, \rho_i) = 1$. This is equivalent to $\rho_i \in S_{\mathsf{cm},i}$, and T is the total circuit size of extraction, membership testing for Z, and D', up to $O(1)$, with $\mathbb{x}$ and td hardwired.

We construct a two-stage adversary $\mathsf{A}_{\mathbb{x}} = (\mathsf{A}_{\mathbb{x},0}, \mathsf{A}_{\mathbb{x},1})$ against the CIH family using the relation $R_\phi = R_{\mathbb{x},\mathsf{td}}$ as follows:

- $\mathsf{A}_{\mathbb{x},0}(1^\lambda)$:
  1. Sample $\gamma \leftarrow \{0,1\}^{\mathsf{r}_{\mathsf{F}}}$.
  2. Sample $(\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{VC.Gen}(1^\lambda, \mathsf{F}(\mathbb{x}, \cdot; \gamma))$.
  3. Set $\phi \coloneqq (\mathbb{x}, \mathsf{td})$ and $\mathsf{aux} \coloneqq (\mathsf{pp}, \mathsf{td})$.
  4. Output (ϕ, aux).
- $\mathsf{A}_{\mathbb{x},1}(\mathsf{aux}, h_{\mathsf{CI}})$:
  1. Parse aux as $(\mathsf{pp}, \mathsf{td})$.
  2. Compute $(\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}})$.
  3. Output cm.

Taking T to be the uniform bound above over all generated trapdoors, every (ϕ, aux) in the support of $\mathsf{A}_{\mathbb{x},0}(1^\lambda)$ satisfies $R_\phi \in \mathbf{R}_\lambda$. The total size of $(\mathsf{A}_{\mathbb{x},0}, \mathsf{A}_{\mathbb{x},1})$ is at most t_{CI}. We can conclude that:

$$\begin{aligned}
&\Pr_{\mathsf{Exp}}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\exists\widetilde{\Pi}:\widetilde{\Pi}[\mathsf{Q}]=\mathsf{a}\ \wedge\ \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\,\mathsf{D}(\mathbb{x},z,\rho)=1\end{array}\right]\\
&\leq \Pr\left[(\mathsf{cm},\rho)\in R_\phi \;\middle|\; \begin{array}{l}(\phi,\mathsf{aux})\leftarrow\mathsf{A}_{\mathbb{x},0}(1^\lambda)\\ h_{\mathsf{CI}}\leftarrow\mathcal{H}^{(\lambda)}_{\mathsf{CI}}\\ \mathsf{cm}:=\mathsf{A}_{\mathbb{x},1}(\mathsf{aux},h_{\mathsf{CI}})\\ \rho:=h_{\mathsf{CI}}(\mathsf{cm})\end{array}\right]\\
&= \Pr\left[(\mathsf{cm},h_{\mathsf{CI}}(\mathsf{cm}))\in R_\phi \;\middle|\; \begin{array}{l}(\phi,\mathsf{aux})\leftarrow\mathsf{A}_{\mathbb{x},0}(1^\lambda)\\ h_{\mathsf{CI}}\leftarrow\mathcal{H}^{(\lambda)}_{\mathsf{CI}}\\ \mathsf{cm}:=\mathsf{A}_{\mathbb{x},1}(\mathsf{aux},h_{\mathsf{CI}})\end{array}\right]\\
&\leq \epsilon_{\mathsf{CI}}(\lambda,t_{\mathsf{CI}})\,.
\end{aligned}$$

□

6.2.2 Security from function VC

Claim 6.5. *Let z_{VC} be the function hiding error of* VC*. The following holds:*

$$\Pr_{\mathsf{Exp}}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\exists\widetilde{\Pi}:\widetilde{\Pi}[\mathsf{Q}]=\mathsf{a}\ \wedge\ \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\,\mathsf{D}(\mathbb{x},z,\rho)=0\end{array}\right]\leq\epsilon_{\mathsf{c}}(\mathbb{x})+\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r}_{\mathsf{F}}}\cdot z_{\mathsf{VC}}(\lambda,t_{\mathsf{VC}})\,,$$

where t_{VC} is as defined in Theorem 6.1.

Proof. According to PCP shadow consistency (Theorem 4.1), the following holds:

$$\begin{aligned}
&\Pr_{\mathsf{Exp}}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\exists\widetilde{\Pi}:\widetilde{\Pi}[\mathsf{Q}]=\mathsf{a}\ \wedge\ \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\,\mathsf{D}(\mathbb{x},z,\rho)=0\end{array}\right]\\
&= \Pr\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\exists\widetilde{\Pi}:\widetilde{\Pi}[\mathsf{Q}]=\mathsf{a}\ \wedge\ \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\,\mathsf{D}(\mathbb{x},z,\rho)=0\end{array}\;\middle|\; \begin{array}{l}\gamma\leftarrow\{0,1\}^{\mathsf{r}_{\mathsf{F}}}\\ (\mathsf{pp},\mathsf{td})\leftarrow\mathsf{VC.Gen}(1^\lambda,\mathsf{F}(\mathbb{x},\cdot;\gamma))\\ h_{\mathsf{CI}}\leftarrow\mathcal{H}^{(\lambda)}_{\mathsf{CI}}\\ (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}):=\widetilde{\mathcal{P}}(\mathsf{pp},h_{\mathsf{CI}})\\ \rho:=h_{\mathsf{CI}}(\mathsf{cm})\\ z:=\mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\end{array}\right]\\
&\leq \Pr\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\exists\widetilde{\Pi}:\widetilde{\Pi}[\mathsf{Q}]=\mathsf{a}\ \wedge\ \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\,\mathsf{D}(\mathbb{x},z,\rho)=0\\ \wedge\,\mathsf{D}(\mathbb{x},z',\rho)=0\end{array}\;\middle|\; \begin{array}{l}\gamma\leftarrow\{0,1\}^{\mathsf{r}_{\mathsf{F}}}\\ (\mathsf{pp},\mathsf{td})\leftarrow\mathsf{VC.Gen}(1^\lambda,\mathsf{F}(\mathbb{x},\cdot;\gamma))\\ h_{\mathsf{CI}}\leftarrow\mathcal{H}^{(\lambda)}_{\mathsf{CI}}\\ (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}):=\widetilde{\mathcal{P}}(\mathsf{pp},h_{\mathsf{CI}})\\ \rho:=h_{\mathsf{CI}}(\mathsf{cm})\\ z:=\mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\\ \widetilde{\Pi}':=\sigma^\ell\\ \widetilde{\Pi}'[\mathsf{Q}]:=\mathsf{a}\\ z':=\mathsf{F}(\mathbb{x},\widetilde{\Pi}';\gamma)\end{array}\right].
\end{aligned}$$

We define the last experiment above as Exp'.

First note that

$$\Pr_{\mathsf{Exp}'}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\,\exists\,\widetilde{\Pi}:\widetilde{\Pi}[\mathsf{Q}]=\mathsf{a}\ \wedge\ \mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)=z\\ \wedge\,\mathsf{D}(\mathbb{x},z,\rho)=0\\ \wedge\,\mathsf{D}(\mathbb{x},z',\rho)=0\end{array}\right]\leq\Pr_{\mathsf{Exp}'}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\,\mathsf{D}(\mathbb{x},z',\rho)=0\end{array}\right].$$

For every $\mathbb{x}$, ρ, and $\widetilde{\Pi}$, we define the set

$$S_{\mathbb{x},\rho,\widetilde{\Pi}}\coloneqq\{\gamma\in\{0,1\}^{\mathsf{r_F}}:\mathsf{D}(\mathbb{x},\mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma),\rho)=0\}\,.$$

By correctness of PCP shadow soundness (Theorem 4.1), we know that for every $\mathbb{x}\notin L(R)$, every $\widetilde{\Pi}$, and every ρ such that $\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho)=1$,

$$\Pr\left[\mathsf{D}(\mathbb{x},z,\rho)=0\;\middle|\;\begin{array}{l}\gamma\leftarrow\{0,1\}^{\mathsf{r_F}}\\ z\coloneqq\mathsf{F}(\mathbb{x},\widetilde{\Pi};\gamma)\end{array}\right]\leq\epsilon_{\mathsf{c}}(\mathbb{x})\,.$$

Therefore,

$$\left|S_{\mathbb{x},\rho,\widetilde{\Pi}}\right|\leq\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}\,.$$

Let $N\coloneqq\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}$. Since the cardinality above is integral, if $N<1$ then every such set is empty and the probability in the claim is zero. We may therefore assume $N\geq 1$. Moreover, on the event that $\gamma\in S_{\mathbb{x},\rho,\widetilde{\Pi}'}$, this set is nonempty, and a uniform element of it equals γ with probability at least $1/N$. We can deduce that

$$\Pr_{\mathsf{Exp}'}\left[\begin{array}{l}\mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x};\rho)=1\\ \wedge\,\mathsf{D}(\mathbb{x},z',\rho)=0\end{array}\right]\leq\Pr_{\mathsf{Exp}'}\left[\begin{array}{l}\mathbf{V}^{\widetilde{\Pi}'}(\mathbb{x};\rho)=1\\ \wedge\,\gamma\in S_{\mathbb{x},\rho,\widetilde{\Pi}'}\end{array}\right]\leq\Pr_{\mathsf{Exp}'}\left[\begin{array}{l}\left|S_{\mathbb{x},\rho,\widetilde{\Pi}'}\right|\leq\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}\\ \wedge\,\gamma\in S_{\mathbb{x},\rho,\widetilde{\Pi}'}\end{array}\right].$$

By definition of conditional probability, we can deduce that

$$\begin{aligned}&\Pr_{\mathsf{Exp}'}\left[\begin{array}{l}\left|S_{\mathbb{x},\rho,\widetilde{\Pi}'}\right|\leq\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}\\ \wedge\,\gamma\in S_{\mathbb{x},\rho,\widetilde{\Pi}'}\end{array}\right]\\ &=\Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star\leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}}}\left[\begin{array}{l}\gamma^\star=\gamma\\ \wedge\,\left|S_{\mathbb{x},\rho,\widetilde{\Pi}'}\right|\leq\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}\\ \wedge\,\gamma\in S_{\mathbb{x},\rho,\widetilde{\Pi}'}\end{array}\right]\left(\Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star\leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}}}\left[\begin{array}{l}\gamma^\star=\gamma\\ \text{conditioned on}\\ \left|S_{\mathbb{x},\rho,\widetilde{\Pi}'}\right|\leq\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}\\ \wedge\,\gamma\in S_{\mathbb{x},\rho,\widetilde{\Pi}'}\end{array}\right]\right)^{-1}\\ &\leq\Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star\leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}}}\left[\gamma^\star=\gamma\right]\left(\Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star\leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}}}\left[\begin{array}{l}\gamma^\star=\gamma\\ \text{conditioned on}\\ \left|S_{\mathbb{x},\rho,\widetilde{\Pi}'}\right|\leq\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}\\ \wedge\,\gamma\in S_{\mathbb{x},\rho,\widetilde{\Pi}'}\end{array}\right]\right)^{-1}\\ &\leq\Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star\leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}}}\left[\gamma^\star=\gamma\right]\cdot\epsilon_{\mathsf{c}}(\mathbb{x})\cdot 2^{\mathsf{r_F}}\,,\end{aligned}$$

where the last inequality follows because

$$\Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star \leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}}} \left[\begin{array}{l} \gamma^\star = \gamma \\ \text{conditioned on} \\ \left| S_{\mathbb{x},\rho,\widetilde{\Pi}'} \right| \leq \epsilon_{\mathsf{c}}(\mathbb{x}) \cdot 2^{\mathsf{r_F}} \\ \wedge\, \gamma \in S_{\mathbb{x},\rho,\widetilde{\Pi}'} \end{array} \right] \geq \frac{1}{\epsilon_{\mathsf{c}}(\mathbb{x}) \cdot 2^{\mathsf{r_F}}} \,.$$

We proceed to bound the remaining probability above.

Consider the following function VC adversary A_{VC}:

$A_{\mathsf{VC}}(\mathsf{pp})$:
1. Sample $h_{\mathsf{CI}} \leftarrow \mathcal{H}_{\mathsf{CI}}^{(\lambda)}$.
2. Run $(\mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf}) \coloneqq \widetilde{\mathcal{P}}(\mathsf{pp}, h_{\mathsf{CI}})$.
3. Compute $\rho \coloneqq h_{\mathsf{CI}}(\mathsf{cm})$.
4. Initialize $\widetilde{\Pi}' \coloneqq (\sigma)^{\ell}$, where σ is an arbitrary element in Σ.
5. Set $\widetilde{\Pi}'[\mathsf{Q}] \coloneqq \mathsf{a}$.
6. Sample uniformly at random $\gamma^\star \leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}$.
7. Output $\gamma^\star$.

Notice that Step 6 in A_{VC} can be implemented as follows:

1. Initialize an empty set $S \coloneqq \emptyset$.
2. For every $\gamma \in \{0,1\}^{\mathsf{r_F}}$:
   - Compute $z \coloneqq \mathsf{F}(\mathbb{x}, \widetilde{\Pi}'; \gamma)$.
   - If $\mathsf{D}(\mathbb{x}, z, \rho) = 0$ then add γ to S.
3. Sample uniformly at random $\gamma^\star \leftarrow S$.
4. Output $\gamma^\star$.

Hence, the size of A_{VC} is at most t_{VC} as defined in Theorem 6.1.

From Theorem 5.4 and the construction of A_{VC},

$$\Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star \leftarrow S_{\mathbb{x},\rho,\widetilde{\Pi}'}}} [\gamma^\star = \gamma] = \Pr_{\substack{\mathsf{Exp}'\\ \gamma^\star \leftarrow A_{\mathsf{VC}}(\mathsf{pp})}} [\gamma^\star = \gamma] \leq \frac{1}{2^{\mathsf{r_F}}} + z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \,.$$

Hence,

$$\begin{aligned} \Pr_{\mathsf{Exp}'} \left[\begin{array}{l} \mathbf{V}^{[\mathsf{Q},\mathsf{a}]}(\mathbb{x}; \rho) = 1 \\ \wedge\, \exists\, \widetilde{\Pi} : \widetilde{\Pi}[\mathsf{Q}] = \mathsf{a} \,\wedge\, \mathsf{F}(\mathbb{x}, \widetilde{\Pi}; \gamma) = z \\ \wedge\, \mathsf{D}(\mathbb{x}, z, \rho) = 0 \\ \wedge\, \mathsf{D}(\mathbb{x}, z', \rho) = 0 \end{array} \right] &\leq \left(\frac{1}{2^{\mathsf{r_F}}} + z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \right) \cdot \epsilon_{\mathsf{c}}(\mathbb{x}) \cdot 2^{\mathsf{r_F}} \\ &= \epsilon_{\mathsf{c}}(\mathbb{x}) + \epsilon_{\mathsf{c}}(\mathbb{x}) \cdot 2^{\mathsf{r_F}} \cdot z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \,. \end{aligned}$$

□

A Function VC for circuits

Theorem A.1. *For every instance-size bound $n \in \mathbb{N}$, let $\ell = \ell(n)$, $s = s(n)$, $t = t(n)$, and $d_{\mathsf{circ}} = d_{\mathsf{circ}}(n)$. Let $\mathcal{F}$ be the family of functions $f\colon \Sigma^\ell \to \Sigma^s$ computed on binary encodings by Boolean circuits of size at most t and depth at most d_{circ}, with circuit descriptions padded to a common length. Let $D = D(n)$ be the depth of the depth-universal circuit for this family; by [CH85], $D = O(d_{\mathsf{circ}})$ and the universal circuit has size $\mathsf{poly}(\ell, s, t, d_{\mathsf{circ}})$. Assume iO and LWE are sub-exponentially secure. There exists VC for $\mathcal{F}$ such that for every security parameter $\lambda \in \mathbb{N}$, $\lambda_{\mathsf{FHE}} = \lambda_{\mathsf{FHE}}(\lambda)$, and $\lambda_{\mathsf{SNARG}} = \lambda_{\mathsf{SNARG}}(\lambda)$, VC satisfies the following:*

- *perfect completeness;*
- *if $D \le \mathsf{poly}(\lambda_{\mathsf{FHE}})$, then for every $c > 1$ and $t_{\mathsf{VC}} \le 2^{\lambda_{\mathsf{FHE}}^{1/c}}$, there exists c' such that the function hiding error $z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \le 2^{-\lambda_{\mathsf{FHE}}^{1/c'}}$;*
- *for every $c > 1$ and $t_{\mathsf{VC}} \le 2^{\lambda_{\mathsf{SNARG}}^{1/c}}$, the function statistical binding error satisfies $\kappa_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \le \mathrm{negl}(\lambda_{\mathsf{SNARG}})$;*
- *generation time $\mathsf{poly}(\lambda, \lambda_{\mathsf{SNARG}}, \lambda_{\mathsf{FHE}}, \ell, s, t, d_{\mathsf{circ}}, D)$;*
- *commitment size $s \cdot \mathsf{poly}(\lambda_{\mathsf{FHE}}, D)$;*
- *public parameter size $\mathsf{poly}(\lambda, \lambda_{\mathsf{SNARG}}, \lambda_{\mathsf{FHE}}, \ell, s, t, d_{\mathsf{circ}}, D)$;*
- *trapdoor size $\mathsf{poly}(\lambda_{\mathsf{FHE}}, D)$;*
- *proof size $\mathsf{poly}(\lambda_{\mathsf{SNARG}})$.*

A.1 Fully homomorphic encryption

A ***leveled fully homomorphic encryption scheme*** with depth bound D, alphabet Σ, message size ℓ, and ciphertext size n is a tuple of efficient algorithms $\mathsf{FHE} = (\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$ such that:

- $\mathsf{FHE.Gen}(1^\lambda, 1^D)$ is a randomized algorithm that takes as input the security parameter and the depth bound and outputs a public key pk and a secret key sk.
- $\mathsf{FHE.Enc}(\mathsf{pk}, m)$ is a randomized algorithm that takes as input the public key pk and a message m and outputs a ciphertext ct.
- $\mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{ct})$ is a deterministic algorithm that takes as input the secret key sk and a ciphertext ct and outputs a message m'.
- $\mathsf{FHE.Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ is a deterministic algorithm that takes as input the public key pk, a circuit f of depth at most D that takes in k inputs and outputs a string in Σ^ℓ, and a vector of ciphertexts $(\mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ and outputs another ciphertext ct'.

We require the following properties for an FHE scheme:

- *Correctness:* For every $\lambda, D \in \mathbb{N}$ and message m,

$$\Pr\left[\mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{FHE.Enc}(\mathsf{pk}, m)) = m \;\middle|\; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Gen}(1^\lambda, 1^D)\right] = 1\,.$$

- *Semantic security:* For every $\lambda, D \in \mathbb{N}$, adversary time bound $t_{\mathsf{FHE}} \in \mathbb{N}$, and t_{FHE}-time adversary $\mathcal{A}$,

$$\left| \Pr\left[b = 1 \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Gen}(1^\lambda, 1^D) \\ (m_0, m_1, \mathsf{aux}) \leftarrow \mathcal{A}(\mathsf{pk}) \\ \mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, m_0) \\ b \leftarrow \mathcal{A}(\mathsf{aux}, \mathsf{pk}, \mathsf{ct}) \end{array} \right] - \Pr\left[b = 1 \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Gen}(1^\lambda, 1^D) \\ (m_0, m_1, \mathsf{aux}) \leftarrow \mathcal{A}(\mathsf{pk}) \\ \mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, m_1) \\ b \leftarrow \mathcal{A}(\mathsf{aux}, \mathsf{pk}, \mathsf{ct}) \end{array} \right] \right| \leq \epsilon_{\mathsf{FHE}}(\lambda, D, t_{\mathsf{FHE}}) \,.$$

- e_{FHE}*-expansion:* For every $\lambda, D \in \mathbb{N}$, message m, and $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Supp}(\mathsf{FHE.Gen}(1^\lambda, 1^D))$,

$$|\mathsf{FHE.Enc}(\mathsf{pk}, m)| \leq e_{\mathsf{FHE}}(\lambda, D, |m|) \,.$$

- c_{FHE}*-compactness:* For every $\lambda, D \in \mathbb{N}$, circuit f of depth at most D that takes in k inputs and outputs a string in Σ^ℓ, inputs $(m_1, \ldots, m_k)$, $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Supp}(\mathsf{FHE.Gen}(1^\lambda, 1^D))$, $\mathsf{ct}_i \in \mathsf{Supp}(\mathsf{FHE.Enc}(\mathsf{pk}, m_i))$ for every $i \in [k]$,

$$|\mathsf{FHE.Eval}(\mathsf{pk}, f, (\mathsf{ct}_1, \ldots, \mathsf{ct}_k))| \leq c_{\mathsf{FHE}}(\lambda, D, |f(m_1, \ldots, m_k)|) \,.$$

- *Perfect homomorphism:* For every $\lambda, D \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Supp}(\mathsf{FHE.Gen}(1^\lambda, 1^D))$, circuit f of depth at most D, and inputs $(m_1, \ldots, m_k)$, let $\mathsf{ct}_i \in \mathsf{Supp}(\mathsf{FHE.Enc}(\mathsf{pk}, m_i))$ for every $i \in [k]$. Then

$$\mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{FHE.Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)) = f(m_1, \ldots, m_k) \,.$$

There is a long line of work that constructs FHE using LWE. The theorem below states the existence of FHE with parameters and security that we need in this paper.

Theorem A.2 ([BGV12; Bra12; GSW13; BDGM19])**.** *Assume the sub-exponential hardness of LWE for the parameter families used by the scheme. There exists a leveled FHE scheme such that for every security parameter $\lambda \in \mathbb{N}$ and depth bound $D \in \mathbb{N}$, it supports circuits of depth at most D and satisfies the following:*

- *size of the secret key* $\mathsf{poly}(\lambda, D)$*;*
- *size of the public key* $\mathsf{poly}(\lambda, D)$*;*
- *expansion* $e_{\mathsf{FHE}}(\lambda, D, \ell) = \ell \cdot \mathsf{poly}(\lambda, D)$*;*
- *compactness* $c_{\mathsf{FHE}}(\lambda, D, \ell) = \ell \cdot \mathsf{poly}(\lambda, D)$*;*
- *perfect correctness;*
- *if $D \leq \mathsf{poly}(\lambda)$, then sub-exponential semantic security: for every $c > 1$, there exists c' such that if $t_{\mathsf{FHE}} \leq 2^{\lambda^{1/c}}$, then $\epsilon_{\mathsf{FHE}}(\lambda, D, t_{\mathsf{FHE}}) \leq 2^{-\lambda^{1/c'}}$;*
- *perfect homomorphism.*

A.2 Construction

Proof of Theorem A.1. Fix the family $\mathcal{F}$ from Theorem A.1. For every message m, let $\mathcal{U}_m$ be the depth-universal circuit of [CH85] with m fixed as its input, so that $\mathcal{U}_m(f) = f(m)$ for every $f \in \mathcal{F}$. Its depth is at most D. Let $\mathsf{FHE} = (\mathsf{FHE.Gen}, \mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$ be the leveled FHE in

Theorem A.2 with depth bound D. Let $\mathsf{SNARG} = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ be the SNARG from Theorem 3.12 for the following NP relation R:

$$R \coloneqq \{(((\mathsf{pk}, \mathsf{ct}_f), \mathsf{cm}, \mathsf{Q}, \mathsf{a}), m) : m \in \Sigma^\ell \wedge m[\mathsf{Q}] = \mathsf{a} \wedge \mathsf{FHE.Eval}(\mathsf{pk}, \mathcal{U}_m, \mathsf{ct}_f) = \mathsf{cm}\}\,.$$

Here an instance is well formed if all components have their prescribed canonical encodings, $|\mathsf{cm}| \le c_{\mathsf{FHE}}(\lambda_{\mathsf{FHE}}, D, s)$, $\mathsf{Q} \subseteq [\ell]$, and $\mathsf{a} \in \Sigma^{\mathsf{Q}}$. Let n_R be an upper bound on the length of every such instance $((\mathsf{pk}, \mathsf{ct}_f), \mathsf{cm}, \mathsf{Q}, \mathsf{a})$ when pk and ct_f have the lengths prescribed by FHE key generation and encryption. By the FHE public-key size, expansion, and compactness bounds, and the bounds on Q and a,

$$n_R = \mathsf{poly}(\lambda_{\mathsf{FHE}}, D, \ell, s, t, d_{\mathsf{circ}})\,.$$

We construct VC for circuits as follows:

- $\mathsf{VC.Gen}(1^\lambda, f)$:
  1. Compute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Gen}(1^{\lambda_{\mathsf{FHE}}}, 1^D)$.
  2. Compute $\mathsf{ct}_f \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, f)$.
  3. Compute $\mathsf{pp}_{\mathsf{SNARG}} \leftarrow \mathsf{Setup}(1^{\lambda_{\mathsf{SNARG}}}, n_R)$.
  4. Output $(\mathsf{pp} \coloneqq (\mathsf{pk}, \mathsf{ct}_f, \mathsf{pp}_{\mathsf{SNARG}}), \mathsf{td} \coloneqq \mathsf{sk})$.
- $\mathsf{VC.Commit}(\mathsf{pp}, m)$:
  1. Parse pp as $(\mathsf{pk}, \mathsf{ct}_f, \mathsf{pp}_{\mathsf{SNARG}})$.
  2. Compute $\mathsf{cm} \coloneqq \mathsf{FHE.Eval}(\mathsf{pk}, \mathcal{U}_m, \mathsf{ct}_f)$.
  3. Output $(\mathsf{cm}, \mathsf{aux} \coloneqq (\mathsf{cm}, m))$.
- $\mathsf{VC.Open}(\mathsf{pp}, \mathsf{aux}, \mathsf{Q})$:
  1. Parse pp as $(\mathsf{pk}, \mathsf{ct}_f, \mathsf{pp}_{\mathsf{SNARG}})$.
  2. Parse aux as (cm, m).
  3. Set $\mathsf{a} \coloneqq m[\mathsf{Q}]$.
  4. Set $\mathbb{x} \coloneqq ((\mathsf{pk}, \mathsf{ct}_f), \mathsf{cm}, \mathsf{Q}, \mathsf{a})$.
  5. Output $\mathsf{pf} \leftarrow \mathcal{P}(\mathsf{pp}_{\mathsf{SNARG}}, \mathbb{x}, m)$.
- $\mathsf{VC.Check}(\mathsf{pp}, \mathsf{cm}, \mathsf{Q}, \mathsf{a}, \mathsf{pf})$:
  1. Parse pp as $(\mathsf{pk}, \mathsf{ct}_f, \mathsf{pp}_{\mathsf{SNARG}})$; if parsing fails, output 0.
  2. Set $\mathbb{x} \coloneqq ((\mathsf{pk}, \mathsf{ct}_f), \mathsf{cm}, \mathsf{Q}, \mathsf{a})$.
  3. If $\mathbb{x}$ is not well formed or $|\mathbb{x}| > n_R$, output 0.
  4. Output $\mathcal{V}(\mathsf{pp}_{\mathsf{SNARG}}, \mathbb{x}, \mathsf{pf})$.
- $\mathsf{VC.Extract}(\mathsf{td}, \mathsf{cm})$:
  1. Parse td as sk.
  2. Output $\mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{cm})$.

Completeness. For honestly generated values, the corresponding instance is well formed and has length at most n_R, while $m[\mathsf{Q}] = \mathsf{a}$ and $\mathsf{FHE.Eval}(\mathsf{pk}, \mathcal{U}_m, \mathsf{ct}_f) = \mathsf{cm}$. Thus, m is a valid witness for the corresponding instance, and completeness follows from perfect completeness of SNARG.

Function hiding. $z_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \le \epsilon_{\mathsf{FHE}}(\lambda_{\mathsf{FHE}}, D, t_{\mathsf{VC}})$ from semantic security of FHE.

Function statistical binding. We bound the following probability:

$$
\begin{aligned}
&\Pr\left[\begin{array}{l|l}
\mathsf{VC.Check}(\mathsf{pp},\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf})=1 & (\mathsf{pp},\mathsf{td})\leftarrow\mathsf{VC.Gen}(1^\lambda,f)\\
\wedge\forall\,\widetilde{m}\in\Sigma^\ell:(\widetilde{m}[\mathsf{Q}]\neq\mathsf{a}\ \vee\ f(\widetilde{m})\neq y) & (\mathsf{pk},\mathsf{ct}_f,\mathsf{pp}_{\mathsf{SNARG}}):=\mathsf{pp}\\
& (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}):=A_{\mathsf{VC}}(\mathsf{pp})\\
& \mathbb{x}:=((\mathsf{pk},\mathsf{ct}_f),\mathsf{cm},\mathsf{Q},\mathsf{a})\\
& y:=\mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})
\end{array}\right]\\
&=\Pr\left[\begin{array}{l|l}
\mathbb{x}\text{ is well formed}\ \wedge\ |\mathbb{x}|\leq n_R & (\mathsf{pp},\mathsf{td})\leftarrow\mathsf{VC.Gen}(1^\lambda,f)\\
\wedge\,\mathcal{V}(\mathsf{pp}_{\mathsf{SNARG}},\mathbb{x},\mathsf{pf})=1 & (\mathsf{pk},\mathsf{ct}_f,\mathsf{pp}_{\mathsf{SNARG}}):=\mathsf{pp}\\
\wedge\,\forall\,\widetilde{m}\in\Sigma^\ell: & (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}):=A_{\mathsf{VC}}(\mathsf{pp})\\
\quad(\widetilde{m}[\mathsf{Q}]\neq\mathsf{a}\ \vee\ f(\widetilde{m})\neq y) & \mathbb{x}:=((\mathsf{pk},\mathsf{ct}_f),\mathsf{cm},\mathsf{Q},\mathsf{a})\\
& y:=\mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})
\end{array}\right]\\
&=\Pr\left[\begin{array}{l|l}
\mathbb{x}\text{ is well formed}\ \wedge\ |\mathbb{x}|\leq n_R & (\mathsf{pp},\mathsf{td})\leftarrow\mathsf{VC.Gen}(1^\lambda,f)\\
\wedge\,\mathcal{V}(\mathsf{pp}_{\mathsf{SNARG}},\mathbb{x},\mathsf{pf})=1 & (\mathsf{pk},\mathsf{ct}_f,\mathsf{pp}_{\mathsf{SNARG}}):=\mathsf{pp}\\
\wedge\,\forall\,\widetilde{m}\in\Sigma^\ell: & (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}):=A_{\mathsf{VC}}(\mathsf{pp})\\
\quad(\widetilde{m}[\mathsf{Q}]\neq\mathsf{a}\ \vee\ f(\widetilde{m})\neq y) & \mathbb{x}:=((\mathsf{pk},\mathsf{ct}_f),\mathsf{cm},\mathsf{Q},\mathsf{a})\\
\wedge\,\exists\,m\in\Sigma^\ell: & y:=\mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\\
\quad(m[\mathsf{Q}]=\mathsf{a}\ \wedge\ \mathsf{FHE.Eval}(\mathsf{pk},\mathcal{U}_m,\mathsf{ct}_f)=\mathsf{cm}) &
\end{array}\right]\\
&\quad+\Pr\left[\begin{array}{l|l}
\mathbb{x}\text{ is well formed}\ \wedge\ |\mathbb{x}|\leq n_R & (\mathsf{pp},\mathsf{td})\leftarrow\mathsf{VC.Gen}(1^\lambda,f)\\
\wedge\,\mathcal{V}(\mathsf{pp}_{\mathsf{SNARG}},\mathbb{x},\mathsf{pf})=1 & (\mathsf{pk},\mathsf{ct}_f,\mathsf{pp}_{\mathsf{SNARG}}):=\mathsf{pp}\\
\wedge\,\forall\,\widetilde{m}\in\Sigma^\ell: & (\mathsf{cm},\mathsf{Q},\mathsf{a},\mathsf{pf}):=A_{\mathsf{VC}}(\mathsf{pp})\\
\quad(\widetilde{m}[\mathsf{Q}]\neq\mathsf{a}\ \vee\ f(\widetilde{m})\neq y) & \mathbb{x}:=((\mathsf{pk},\mathsf{ct}_f),\mathsf{cm},\mathsf{Q},\mathsf{a})\\
\wedge\,\forall\,m\in\Sigma^\ell: & y:=\mathsf{VC.Extract}(\mathsf{td},\mathsf{cm})\\
\quad(m[\mathsf{Q}]\neq\mathsf{a}\ \vee\ \mathsf{FHE.Eval}(\mathsf{pk},\mathcal{U}_m,\mathsf{ct}_f)\neq\mathsf{cm}) &
\end{array}\right]\\
&\leq\mathrm{negl}(\lambda_{\mathsf{SNARG}})\,.
\end{aligned}
$$

Efficiency. The padded circuit descriptions, the circuit deciding R, and n_R have size polynomial in $\lambda_{\mathsf{FHE}}, D, \ell, s, t, d_{\mathsf{circ}}$. Thus, $\mathsf{VC.Gen}$, which runs FHE key generation and encryption and the SNARG setup, has the claimed polynomial running time. FHE compactness gives commitment size $s\cdot\mathsf{poly}(\lambda_{\mathsf{FHE}},D)$. The public parameter consists of the FHE public key, the encrypted auxiliary function, and the SNARG public parameter, so its size is polynomial in the parameters stated in Theorem A.1; the trapdoor is the FHE secret key and has size $\mathsf{poly}(\lambda_{\mathsf{FHE}},D)$; and the opening proof is a SNARG proof and has size $\mathsf{poly}(\lambda_{\mathsf{SNARG}})$.

□

Acknowledgments

We are grateful to Ignacio Manzur and Zhengzhong Jin for their meticulous feedback on an earlier draft of this paper. We additionally thank Zvika Brakerski and Vinod Vaikuntanathan for insightful discussions on fully homomorphic encryption, and Karthik C.S. and Dor Minzer for valuable discussions regarding the notion of shadow PCPs. We thank Gal Arnon and Ariel Gabizon for invaluable feedback. We thank David Wu for pointing out missing citations regarding function vector commitments.

Eylon Yogev is supported by the Israel Science Foundation (Grant No. 2302/22) and the European Research Union (ERC, CRYPTOPROOF, 101164375). 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.

References

[ACFY24] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. "STIR: Reed–Solomon Proximity Testing with Fewer Queries". In: *Proceedings of the 44th Annual International Cryptology Conference*. CRYPTO '24. 2024, pp. 380–413.

[ACFY25] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. "WHIR: Reed-Solomon Proximity Testing with Super-Fast Verification". In: *Proceedings of the 44th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '25. 2025, pp. 214–243.

[ACLMT23] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri AravindaKrishnan Thyagarajan. "Lattice-Based Succinct Arguments from Vanishing Polynomials - (Extended Abstract)". In: *Proceedings of the 43rd Annual International Cryptology Conference*. CRYPTO '23. 2023, pp. 72–105.

[ACY23] Gal Arnon, Alessandro Chiesa, and Eylon Yogev. "IOPs with Inverse Polynomial Soundness Error". In: *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '23. 2023, pp. 752–761.

[AFLN24] Martin R. Albrecht, Giacomo Fenzi, Oleksandra Lapiha, and Ngoc Khanh Nguyen. "SLAP: Succinct Lattice-Based Polynomial Commitments from Standard Assumptions". In: *Proceedings of the 43rd Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '24. 2024, pp. 90–119.

[AGLMS23] Arasu Arun, Chaya Ganesh, Satya V. Lokam, Tushar Mopuri, and Sriram Sridhar. "Dew: A Transparent Constant-Sized Polynomial Commitment Scheme". In: *Proceedings of the 26th International Conference on Practice and Theory in Public Key Cryptography*. PKC '23. 2023, pp. 542–571.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. "Ligero: Lightweight Sublinear Arguments Without a Trusted Setup". In: *Proceedings of the 24th ACM Conference on Computer and Communications Security*. CCS '17. 2017, pp. 2087–2104.

[AY25] Gal Arnon and Eylon Yogev. "Towards a White-Box Secure Fiat-Shamir Transformation". In: *Proceedings of the 45th Annual International Cryptology Conference*. CRYPTO '25. 2025, pp. 27–56.

[BBBPWM18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *Proceedings of the 39th IEEE Symposium on Security and Privacy*. Oakland '18. 2018, pp. 315–334.

[BBHMR19] James Bartusek, Liron Bronfman, Justin Holmgren, Fermi Ma, and Ron D. Rothblum. "On the (In)security of Kilian-Based SNARGs". In: *Proceedings of the 17th Theory of Cryptography Conference*. TCC '19. 2019, pp. 522–551.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed–Solomon Interactive Oracle Proofs of Proximity". In: *Proceedings of the 45th International Colloquium on Automata, Languages and Programming*. ICALP '18. 2018, 14:1–14:17.

[BBKLP23] Zvika Brakerski, Maya Farber Brodsky, Yael Tauman Kalai, Alex Lombardi, and Omer Paneth. "SNARGs for Monotone Policy Batch NP". In: *Proceedings of the 43rd Annual International Cryptology Conference*. CRYPTO '23. 2023, 252–283.

[BC25] Dan Boneh and Binyi Chen. "LatticeFold+: Faster, Simpler, Shorter Lattice-Based Folding for Succinct Proof Systems". In: *Proceedings of the 45th Annual International Cryptology Conference*. CRYPTO '25. 2025, pp. 327–361.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. "From Extractable Collision Resistance to Succinct Non-Interactive Arguments of Knowledge, and Back Again". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS '12. 2012, pp. 326–349.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. "Recursive composition and bootstrapping for SNARKS and proof-carrying data". In: *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*. STOC '13. 2013, pp. 111–120.

[BCFL23] David Balbás, Dario Catalano, Dario Fiore, and Russell W. F. Lai. "Chainable Functional Commitments for Unbounded-Depth Circuits". In: *Proceedings of the 21st Theory of Cryptography Conference*. TCC '23. 2023, pp. 363–393.

[BCIOP13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. "Succinct Non-Interactive Arguments via Linear Interactive Proofs". In: *Proceedings of the 10th Theory of Cryptography Conference*. TCC '13. 2013, pp. 315–333.

[BCP13] Nir Bitansky, Ran Canetti, and Omer Paneth. *How To Construct Extractable One-Way Functions Against Uniform Adversaries*. Cryptology ePrint Archive, Report 2013/468. 2013.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *Proceedings of the 14th Theory of Cryptography Conference*. TCC '16-B. 2016, pp. 31–60.

[BDGM19] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. "Leveraging Linear Decryption: Rate-1 Fully-Homomorphic Encryption and Time-Lock Puzzles". In: *Proceedings of the 17th Theory of Cryptography Conference*. TCC '19. 2019, pp. 407–437.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS '12. 2012, 309–325.

[BISW17] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. "Lattice-Based SNARGs and Their Application to More Efficient Obfuscation". In: *Proceedings of the 36th Annual International Conference on Theory and Applications of Cryptographic Techniques*. EUROCRYPT '17. 2017, pp. 247–277.

[BISW18] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. "Quasi-Optimal SNARGs via Linear Multi-Prover Interactive Proofs". In: *Proceedings of the 37th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '18. 2018, pp. 222–255.

[BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. "NIZK from LPN and Trapdoor Hash via Correlation Intractability for Approximable Relations". In: *Proceedings of the 40th Annual International Cryptology Conference*. CRYPTO '20. 2020, pp. 738–767.

[BMVY25] Mitali Bafna, Dor Minzer, Nikhil Vyas, and Zhiwei Yun. "Quasi-Linear Size PCPs with Small Soundness from HDX". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. STOC '25. 2025, pp. 45–53.

[BS05] Eli Ben-Sasson and Madhu Sudan. "Simple PCPs with poly-log rate and query complexity". In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. STOC '05. 2005, pp. 266–275.

[Bar01] Boaz Barak. "How to Go Beyond the Black-Box Simulation Barrier". In: *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*. FOCS '11. 2001, pp. 106–115.

[Bit+14] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. *The Hunting of the SNARK*. ePrint 2014/580. 2014.

[Bra12] Zvika Brakerski. "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP". In: *Proceedings of the 32nd Annual International Cryptology Conference*. CRYPTO '12. 2012, pp. 868–886.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin. "On the Correlation Intractability of Obfuscated Pseudorandom Functions". In: *Proceedings of the 13th Theory of Cryptography Conference*. TCC '16. 2016, pp. 389–415.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. "Fiat-Shamir and Correlation Intractability from Strong KDM-Secure Encryption". In: *Proceedings of the 37th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '18. 2018, pp. 91–122.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. "The random oracle methodology, revisited". In: *Journal of the ACM* 51.4 (2004), pp. 557–594.

[CGJJZ23] Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. "Correlation Intractability and SNARGs from Sub-exponential DDH". In: *Proceedings of the 43rd Annual International Cryptology Conference*. CRYPTO '23. 2023, 635–668.

[CGKS23] Matteo Campanelli, Chaya Ganesh, Hamidreza Khoshakhlagh, and Janno Siim. "Impossibilities in Succinct Arguments: Black-Box Extraction and More". In: *Proceedings of the 14th International Conference on Cryptology in Africa*. AFRICACRYPT' 23. 2023, pp. 465–489.

[CH85] Stephen A. Cook and H. James Hoover. "A Depth-Universal Circuit". In: *SIAM Journal on Computing* 14.4 (1985), pp. 833–839. DOI: 10.1137/0214058.

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. "Non-interactive Batch Arguments for NP from Standard Assumptions". In: *Proceedings of the 41st Annual International Cryptology Conference*. CRYPTO '21. 2021, pp. 394–423.

[CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. "SNARGs for $\mathcal{P}$ from LWE". In: *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science*. FOCS '22. 2022, pp. 68–79.

[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. "Lattice-Based Succinct Arguments from Vanishing Polynomials - (Extended Abstract)". In: *Proceedings of the 43rd Annual International Cryptology Conference*. CRYPTO '23. 2023, pp. 72–105.

[CY21a] Alessandro Chiesa and Eylon Yogev. "Subquadratic SNARGs in the Random Oracle Model". In: *Proceedings of the 41st Annual International Cryptology Conference*. CRYPTO '21. 2021, pp. 711–741.

[CY21b] Alessandro Chiesa and Eylon Yogev. "Tight Security Bounds for Micali's SNARGs". In: *Proceedings of the 19th Theory of Cryptography Conference*. TCC '21. 2021, pp. 401–434.

[CY24] Alessandro Chiesa and Eylon Yogev. *Building Cryptographic Proofs from Hash Functions*. 2024. URL: `https://github.com/hash-based-snargs-book`.

[Can+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. "Fiat-Shamir: from practice to theory". In: *Proceedings of the 51st Annual ACM Symposium on Theory of Computing*. STOC '19. 2019, pp. 1082–1090.

[DFH12] Ivan Damgård, Sebastian Faust, and Carmit Hazay. "Secure Two-Party Computation with Low Communication". In: *Proceedings of the 9th Theory of Cryptography Conference*. TCC '12. 2012, pp. 54–74.

[DGKV22] Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. "Rate-1 Non-Interactive Arguments for Batch-NP and Applications". In: *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science*. FOCS '22. 2022, pp. 1057–1068.

[DHKKLM26] Lalita Devadas, Samuel B. Hopkins, Yael Tauman Kalai, Pravesh K. Kothari, Alex Lombardi, and Surya Mathialagan. *SNARGs for NP and Non-Signaling PCPs, Revisited*. Cryptology ePrint Archive, Paper 2026/006. 2026. URL: `https://eprint.iacr.org/2026/006`.

[DWW24] Lalita Devadas, Brent Waters, and David J. Wu. "Batching Adaptively-Sound SNARGs for NP". In: *Proceedings of the 22nd Theory of Cryptography Conference*. TCC '24. 2024, 339–370.

[Din07] Irit Dinur. "The PCP theorem by gap amplification". In: *Journal of the ACM* 54.3 (2007), p. 12.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. "Quadratic Span Programs and Succinct NIZKs without PCPs". In: *Proceedings of the 32nd Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '13. 2013, pp. 626–645.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. "On the (In)security of the Fiat-Shamir Paradigm". In: *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '03. 2003, pp. 102–113.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based". In: *Proceedings of the 33rd Annual International Cryptology Conference*. CRYPTO '13. 2013, pp. 75–92.

[GSWW22] Rachit Garg, Kristin Sheridan, Brent Waters, and David J. Wu. "Fully Succinct Batch Arguments for sfNP from Indistinguishability Obfuscation". In: *Proceedings of the 20th Theory of Cryptography Conference*. TCC '22. 2022, pp. 526–555.

[GW11] Craig Gentry and Daniel Wichs. "Separating Succinct Non-Interactive Arguments From All Falsifiable Assumptions". In: *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*. STOC '11. 2011, pp. 99–108.

[GZ21] Alonso González and Alexandros Zacharakis. "Fully-Succinct Publicly Verifiable Delegation from Constant-Size Assumptions". In: *Proceedings of the 19th Theory of Cryptography Conference*. TCC '21. 2021, pp. 529–557.

[Gro10] Jens Groth. "Short Pairing-Based Non-interactive Zero-Knowledge Arguments". In: *Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security*. Ed. by Masayuki Abe. ASIACRYPT '10. 2010, pp. 321–340.

[Gro16] Jens Groth. "On the Size of Pairing-Based Non-interactive Arguments". In: *Proceedings of the 35th Annual International Conference on Theory and Applications of Cryptographic Techniques*. EUROCRYPT '16. 2016, pp. 305–326.

[HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. "SNARGs for P from Sub-exponential DDH and QR". In: *Proceedings of the 41st Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '22. 2022, pp. 520–549.

[HL18] Justin Holmgren and Alex Lombardi. "Cryptographic Hashing from Strong One-Way Functions (Or: One-Way Product Functions and Their Applications)". In: *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '18. 2018, pp. 850–858.

[HLR21] Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. "Fiat-Shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge)". In: *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing*. STOC '21. 2021, pp. 750–760.

[HW15] Pavel Hubácek and Daniel Wichs. "On the Communication Complexity of Secure Function Evaluation with Long Output". In: *Proceedings of the 6th Innovations in Theoretical Computer Science Conference*. ITCS '15. 2015, pp. 163–172.

[JJ21] Abhishek Jain and Zhengzhong Jin. "Non-interactive Zero Knowledge from Sub-exponential DDH". In: *Proceedings of the 40th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '21. 2021, pp. 3–32.

[JKLM25] Zhengzhong Jin, Yael Tauman Kalai, Alex Lombardi, and Surya Mathialagan. "Universal SNARGs for NP from Proofs of Correctness". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. STOC '25. 2025, 933–943.

[Jin26] Zhengzhong Jin. *Low-Norm Nullstellensatz Hypothesis for the AND Code is False*. Cryptology ePrint Archive, Paper 2026/1087. 2026. URL: `https://eprint.iacr.org/2026/1087`.

[KLV23] Yael Tauman Kalai, Alex Lombardi, and Vinod Vaikuntanathan. "SNARGs and PPAD Hardness from the Decisional Diffie-Hellman Assumption". In: *Proceedings of the 42nd Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '23. 2023, pp. 470–498.

[KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. "Boosting Batch Arguments and RAM Delegation". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. STOC '23. 2023, pp. 1545–1552.

[KMNPRY22] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. "One-Way Functions and (Im)perfect Obfuscation". In: *SIAM J. Comput.* 51.6 (2022), pp. 1769–1795.

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev. "Collision Resistant Hashing for Paranoids: Dealing with Multiple Collisions". In: *Proceedings of the 37th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '18. 2018, pp. 162–194.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. "From Obfuscation to the Security of Fiat-Shamir for Proofs". In: *Proceedings of the 37th Annual International Cryptology Conference*. CRYPTO '17. 2017, pp. 224–251.

[KRS25] Dmitry Khovratovich, Ron D. Rothblum, and Lev Soukhanov. "How to Prove False Statements: Practical Attacks on Fiat-Shamir". In: *Proceedings of the 45th Annual International Cryptology Conference*. CRYPTO '25. 2025, pp. 3–26.

[KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. "Somewhere Statistical Soundness, Post-Quantum Security, and SNARGs". In: *Proceedings of the 19th Theory of Cryptography Conference*. TCC '21. 2021, pp. 330–368.

[Kil92] Joe Kilian. "A note on efficient zero-knowledge proofs and arguments". In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. STOC '92. 1992, pp. 723–732.

[Lip13] Helger Lipmaa. "Succinct Non-interactive Zero Knowledge Arguments from Span Programs and Linear Error-Correcting Codes". In: *Proceedings of the 19th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '13. 2013, pp. 41–60.

[Lom26] Alex Lombardi. *Counterexamples to the Low-Norm Nullstellensatz Hypothesis*. Cryptology ePrint Archive, Paper 2026/1114. 2026. URL: https://eprint.iacr.org/2026/1114.

[Mic00] Silvio Micali. "Computationally Sound Proofs". In: *SIAM Journal on Computing* 30.4 (2000). Preliminary version appeared in FOCS '94., pp. 1253–1298.

[NWW24] Shafik Nassar, Brent Waters, and David J. Wu. "Monotone Policy BARGs from BARGs and Additively Homomorphic Encryption". In: *Proceedings of the 22nd Theory of Cryptography Conference*. TCC '24. 2024, pp. 399–430.

[NWW25] Shafik Nassar, Brent Waters, and David J. Wu. "Monotone-Policy BARGs and More from BARGs and Quadratic Residuosity". In: *Proceedings of the 28th International Conference on Practice and Theory in Public Key Cryptography*. PKC '25. 2025, pp. 283–313.

[OPWW15] Tatsuaki Okamoto, Krzysztof Pietrzak, Brent Waters, and Daniel Wichs. "New Realizations of Somewhere Statistically Binding Hashing and Positional Accumulators". In: *Proceedings of the 21st International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '15. 2015, pp. 121–145.

[PP22] Omer Paneth and Rafael Pass. "Incrementally Verifiable Computation via Rate-1 Batch Arguments". In: *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science*. FOCS '22. 2022, pp. 1045–1056.

[PS19] Chris Peikert and Sina Shiehian. "Noninteractive Zero Knowledge for NP from (Plain) Learning with Errors". In: *Proceedings of the 39th Annual International Cryptology Conference*. CRYPTO '19. 2019, pp. 89–114.

[SW14] Amit Sahai and Brent Waters. "How to use indistinguishability obfuscation: deniable encryption, and more". In: *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. STOC '14. 2014, pp. 475–484.

[WW22] Brent Waters and David J. Wu. "Batch Arguments for NP and More from Standard Bilinear Group Assumptions". In: *Proceedings of the 42nd Annual International Cryptology Conference*. CRYPTO '22. 2022, 433–463.

[WW23a] Hoeteck Wee and David J. Wu. "Lattice-Based Functional Commitments: Fast Verification and Cryptanalysis". In: *Proceedings of the 29th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '23. 2023, pp. 201–235.

[WW23b] Hoeteck Wee and David J. Wu. "Succinct Vector, Polynomial, and Functional Commitments from Lattices". In: *Proceedings of the 42nd Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '23. 2023, pp. 385–416.

[WW24a] Brent Waters and David J. Wu. "Boosting Batch Arguments and RAM Delegation". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. STOC '24. 2024, 387–398.

[WW24b] Hoeteck Wee and David J. Wu. "Succinct Functional Commitments for Circuits from k-sfLin". In: *Proceedings of the 43rd Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '24. 2024, pp. 280–310.

[WW25] Brent Waters and David J. Wu. "A Pure Indistinguishability Obfuscation Approach to Adaptively-Sound SNARGs for NP". In: *Proceedings of the 45th Annual International Cryptology Conference*. CRYPTO '21. 2025, pp. 394–423.

[WZ24] Brent Waters and Mark Zhandry. "Adaptive Security in SNARGs via iO and Lossy Functions". In: *Proceedings of the 44th Annual International Cryptology Conference*. CRYPTO '24. 2024, pp. 72–104.

[Wee25a] Hoeteck Wee. "Almost Optimal KP and CP-ABE for Circuits from Succinct LWE". In: *Proceedings of the 44th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '25. 2025, pp. 34–62.

[Wee25b] Hoeteck Wee. "Functional Commitments and SNARGs for P/poly from SIS". In: *Proceedings of the 45th Annual International Cryptology Conference*. CRYPTO '25. 2025, 617–640.

[ZCF24] Hadas Zeilberger, Binyi Chen, and Ben Fisch. "BaseFold: Efficient Field-Agnostic Polynomial Commitment Schemes from Foldable Codes". In: *Proceedings of the 44th Annual International Cryptology Conference*. CRYPTO '24. 2024, pp. 138–169.