Lower Bound on SNARGs in the Random Oracle Model

Iftach Haitner*
iftachh@tauex.tau.ac.il
Tel Aviv University

Daniel Nukrai*
daniel.nukrai@cs.tau.ac.il
Tel Aviv University

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

February 16, 2022

Abstract

Succinct non-interactive arguments (SNARGs) have become a fundamental primitive in the cryptographic community. The focus of this work is constructions of SNARGs in the Random Oracle Model (ROM). Such SNARGs enjoy post-quantum security and can be deployed using lightweight cryptography to heuristically instantiate the random oracle. A ROM-SNARG is (t,ε)-*sound* if no t-query malicious prover can convince the verifier to accept a false statement with probability larger than ε. Recently, Chiesa-Yogev (CRYPTO '21) presented a ROM-SNARG of length $\Theta(\log(t/\varepsilon)\cdot\log t)$ (ignoring $\log n$ factors, for n being the instance size). This improvement, however, is still far from the (folklore) lower bound of $\Omega(\log(t/\varepsilon))$.

Assuming the *randomized exponential-time hypothesis*, we prove a tight lower bound of $\Omega(\log(t/\varepsilon)\cdot\log t)$ for the length of (t,ε)-sound ROM-SNARGs. Our lower bound holds for constructions with non-adaptive verifiers and strong soundness notion called *salted soundness*, restrictions that hold for *all* known constructions (ignoring contrived counterexamples). We prove our lower bound by transforming any short ROM-SNARG (of the considered family) into a same length ROM-SNARG in which the verifier asks only a *few* oracles queries, and then apply the recent lower bound of Chiesa-Yogev (TCC '20) for such SNARGs.

Keywords: Random oracle; SNARGs; high-entropy sets; lower bound

[*] The Blavatnik School of Computer Science, Tel-Aviv University. Member of the Check Point Institute for Information Security.

Contents

1 Introduction **1**
1.1 Our Results . . . 2
1.2 Related Work . . . 5
Paper Organization . . . 6

2 Techniques **6**
2.1 Warmup . . . 6
2.2 Actual Scenario . . . 7
2.3 Completeness . . . 8
2.4 Soundness . . . 9

3 Preliminaries **10**
3.1 Notations . . . 10
3.2 Entropy Measures . . . 10
3.3 Randomized Exponential Time Hypothesis . . . 11
3.4 Random Oracles . . . 11
3.5 Non-Interactive Arguments in the ROM . . . 11

4 Hitting High-Entropy Distribution using Product Sets **13**
4.1 High-Entropy Distributions Have an (Almost) Uniform Large Projection . . . 15
4.2 Hitting almost Full-Entropy Distributions using Product Set . . . 17

5 Lower Bound on the Length of ROM-SNARGs **20**
5.1 Proof of Theorem 5.1 . . . 21
5.2 Short ROM-SNARGs to Low Query ROM-SNARGs, Proving Lemma 5.2 . . . 22

6 Lower Bound on the Length of ROM-SVCs **25**

Acknowledgments **27**

References **27**

A Salted Soundness Amplification **31**

B A Folklore Lower Bound on ROM-SNARG Length **32**

C Proof of Lemma 5.3 **33**

1 Introduction

Constructions in the *random oracle model* (ROM) have shaped our understanding of the cryptographic world. Being a simple information-theoretic model, the ROM was found to be a very useful framework for understating what can be done (sometimes only heuristically), and what is unlikely to be achieved using (merely) *symmetric-key* cryptography. A notable example for the above is *key-agreement* protocols. Merkle [Mer82] has constructed a key-agreement protocol in the ROM with a quadratic gap between the query complexity of the players and the eavesdropper. Barak and Mahmoody-Ghidary [BM17], building on the seminal work of Impagliazzo and Rudich [IR89], proved that the quadratic gap achieved by [Mer82] is optimal, and Haitner, Mazor, Oshman, Reingold, and Yehudayoff [HMORY19], showed that for a large family of constructions, the communication complexity of [Mer82] is optimal.

Another primitive whose constructions in the ROM have high impact is *Succinct Non-interactive Argument systems* (SNARGs): non-interactive computationally sound proofs (arguments) for NP of *succinct* proof length (sublinear in the instance length). The first construction of SNARGs was given by Micali [Mic00] in the ROM. This feasibility result turned out to be very influential both theoretically and practically. In theory, it was shown how to instantiate SNARGs in the *standard model* for many languages of interest by instantiating the Fiat and Shamir [FS86] paradigm with a specific family of hash functions [CCHLRR18]. In practice, the succinctness of the proof is imperative in applications such as cryptocurrency and blockchain, where proofs are broadcast in a peer-to-peer network and (redundantly) stored at every network node, c.f., [BCGGMTV14; Zc14]. As such, improving the concrete efficiency of SNARGs is the focus of long line of work c.f., [Gro16; ZGKPP17; AHIV17; BBHR19; WTSTW18; BBBPWM18; BCRSVW19; CHMMVW20; BFS20; COS20; Sta18; LSTW21; CY21b; CY21a; GNS21].

ROM-SNARGs, like the one of [Mic00], have several attractive features. First, to date, they are the most efficient approach for post-quantum security with public verification (i.e., the verifier has no secrets). Moreover, from a practical perspective, one can heuristically instantiate the random oracle with a suitable cryptographic hash function. The result is a SNARG that uses lightweight cryptography (no need for public-key primitives), is easy to deploy (users only need to agree on a hash function), and has no trusted setup. The best ROM-SNARG appeared in the recent work of Chiesa and Yogev [CY21a], who constructed a (t, ε)-sound ROM-SNARG of proof length of $O(\log(t/\varepsilon) \cdot \log t \cdot \log n)$, where n is the instance length. A ROM-SNARG is (t, ε)-*sound* if no t-queries (malicious) prover can convince the verifier to accept a false statement with probability larger than ε.[1]

Interestingly, and in contrast to other important primitives such as key-agreement protocols [IR89; HMORY19] and digital signatures [GGKT05; BMG07], we are lacking crucial lower bounds on the length of SNARGs in the ROM. Apart from the weak (folklore) lower bound of $\Omega(\log(t/\varepsilon))$ (see Appendix B), the only exception is the recent bound of Chiesa and Yogev [CY20], who proved that the verifier query complexity of SNARGs cannot be too small. However, their bound does not rule out short ROM-SNARGs with verifier query complexity $\Omega(\log 1/\varepsilon)$, which is common for SNARG constructions.

This state-of-affairs naturally leads to the question of finding the shortest ROM-SNARG. Is it $O(\log(t/\varepsilon) \cdot \log t \cdot \log n)$, as the best-known construction achieve, or is it as short as $O(\log(t/\varepsilon) \cdot \log n)$,

[1] We focus on the *bare* ROM— no computational assumptions are made beyond bounding the query complexity to the oracle.

as achieved in other security models (see Section 1.2.2). In this work, we advance our understanding about the existence of short ROM-SNARGs (with *arbitrary* verifier query complexity).

1.1 Our Results

Assuming the *(randomized) exponential time hypothesis (rETH)*, see details below, we prove that for a large family of constructions, the current state-of-the art ROM-SNARG is (essentially) *optimal*. Specifically, we show that, for this family of constructions, a proof of 3SAT over n variables is of length $\widetilde{\Omega}(\log(t/\varepsilon) \cdot \log t)$ (hiding $\log n$ factors). Matching (up to $\log n$ factors) the construction of the [CY21a]. The family of constructions we consider includes all constructions that have: (i) *non-adaptive* verifier and (ii) *salted soundness*. This includes *all types of constructions* we are aware of [Mic00; BCS16; CY21b; CY21a]). See details below.

- **Exponential time hypothesis.** The (randomized) Exponential Time Hypothesis (rETH) is a stronger version P $\neq$ NP that states that solving 3SAT on n variables takes (randomized) time $2^{\Omega(n)}$. Note that some complexity assumption is inevitable for proving lower bounds on a SNARGs length.[2]
- **Non-adaptive verifier.** The oracle queries are asked by a non-adaptive (deterministic) verifier. That is, the queries are a function of the proof and are *independent* of the answers to other queries.[3]
- **Salted soundness.** This is a natural strengthening of the standard soundness of SNARG, which was introduced in Chiesa and Yogev [CY20]. A (t, ε)-salted-soundness ROM-SNARG allows a cheating prover to request the random oracle to re-sample the answer for a chosen query (similar to changing a "salt" for this query). Each re-sampling costs a unit from the total t query budget allowed. The cheating prover can also return to previously sampled query answers at no cost.[4]

  While one can easily construct contrived ROM-SNARGs for which salted soundness does not hold, we are not aware of any ROM-SNARG that exploits the fact that the prover cannot resample some of the oracle answers in a meaningful way. All constructions we are aware of satisfy salted soundness.[5]

With these notions, we are ready to state our main result.

Theorem 1.1 (Conditional lower bound on ROM-SNARG length. Informal)**.** *Let* $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ *be an* s*-length ROM-SNARG for* n*-variable* 3SAT*, with* (t, ε)*-salted-soundness, perfect completeness, and (deterministic) non-adaptive verifier. Let* $\mathsf{q_P}$ *and* $\mathsf{q_V}$ *be the query complexity of* P *and* V*, respectively, and let* λ *denote the random oracle input and output length.*

Assuming rETH, if $\mathsf{q_V} \cdot \lambda \in o(n)$*, and* $\log t \cdot \log \frac{t}{\varepsilon} \cdot \log^{-1} \mathsf{q_P} \in o(n)$ *then* $\mathsf{s} \geq c \cdot \log t \cdot \log \frac{t}{\varepsilon} \cdot \log^{-1} \mathsf{q_P}$*, for some constant* $c > 0$*.*

[2] This follows since P = NP yields trivial SNARGs for all NP.

[3] We mention that SNARGs resulting from applying the Fiat and Shamir [FS86] paradigm on interactive proofs do *not* require an adaptive verifier, as the queries added by the compilation are determined by the proof (i.e., transcript) sent by the non-adaptive prover.

[4] Our notion of salted soundness is a strengthening of the salted-soundness notion considered in Chiesa and Yogev [CY20]. There, the cheating prover has to decide on a salt for a specific query *before* moving to the next one. See details in Section 3.5.1.

[5] See the analysis given in [CY21b] and in [CY21a], which explicitly allowed the adversary to choose a salt for each query in the construction (e.g., see remark 3.2 in [CY21b]).

We argue that the assumptions on the parameters regime in our theorem are reasonable and consider the most interesting settings (see Theorem 5.1 for the precise list of requirements). The goal of a SNARG is to have the proof length and the verifier complexity be much smaller than the instance size n. Usually, proportional to $\mathrm{poly}(\lambda, \log n)$. Thus, our assumption that $\mathsf{q_V} \cdot \lambda$, and $\log t \cdot \log \frac{t}{\varepsilon} / \log \mathsf{q_P}$ are of order $o(n)$ is rather mild. The third requirement of $\mathsf{q_V} \leq t^{1/10}$ is almost trivial. It says that the query complexity of the verifier is much smaller than the query bound t of the *adversary*, which is very much expected from any reasonable SNARG.

The proof of Theorem 1.1 immediately follows by combing the following lemma with the recent lower bound of Chiesa and Yogev [CY20] on the length ROM-SNARG with low query-complexity verifiers.

Lemma 1.2 (Short ROM-SNARG → low query ROM-SNARG. Informal)**.** *Let* $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ *be a ROM-SNARG for a language* $\mathcal{L}$ *with a deterministic non-adaptive verifier and* (t, ε)*-salted-soundness, perfect completeness, and proof length* s*. Then there exist a verier* V' *of query complexity* $\mathsf{s}/\log t$*, running time* $2^{\mathsf{q_V} \cdot \log t}$ *times that of* V*, such that* $(\mathsf{P}, \mathsf{V}')$ *is a ROM-SNARG for* $\mathcal{L}$ *with* (t, ε)*-soundness and completeness* $\omega(\varepsilon)$*.*

That is, the larger the salted-soundness of ARG, the smaller the number of queries made by V', and the better the completeness. While the completeness and verifier running time of the resulting scheme are rather poor, and we do not encourage to use it as an actual proof system, it is still non-trivial for the parameters in consideration: V' running time is $2^{o(n)}$, for n being the instant length, and the completeness is larger than the soundness error. By [CY20], the existence of such ROM-SNARG for 3SAT contradicts rETH.

Using similar means, we can compile ARG into $(\mathsf{P}', \mathsf{V}')$, with (almost) perfect completeness, but with inefficient prover and slightly longer proof (see details in Section 2). Since this transformation does not yield better lower bounds, and the resulting scheme is impractical, we present the simpler transformation above.

Lower bound on the length of ROM subvector commitments. A *subvector commitment* (SVC) [LM19] allows to succinctly commit to a sequence of values, and later open the commitment for a *subset* of positions (an adversary cannot open any location into two different values). Ideally, the commitment string and the opening size of the SVC are *independent* (or at least not strongly related) of the length of the committed vector and the number of positions to open. This generalization of *vector commitments* [CF13] has a variety of applications, including SNARGs, *verifiable databases with efficient updates*, *updatable zero-knowledge databases*, *universal dynamic accumulators*, and more. Since SVCs in the (bare) ROM are the main building blocks in all ROM-SNARGs constructions, finding shorter ROM-SVCs is the obvious approach towards construction shorter ROM-SNARGs. For this very reason, Theorem 1.1 yields a lower bound on ROM-SVCs for an analog family of constructions: non-adapter receiver and salted-binding (i.e., the sender can resample the oracle outputs).

Theorem 1.3 (Conditional lower bound on the length of ROM subvector commitments. Informal)**.** *Let* CM *be a* (t, ε)*-salted-sound, non-adaptive (deterministic) verification ROM-SVC for vectors of length* n*. Let* $\mathsf{q_S}$ *and* $\mathsf{q_R}$ *be the query complexity of the sender and receiver, respectively. Let* α *denote the commitment length, and* $\beta(\ell)$ *denote the opening length for subsets of size* ℓ*.*

Assuming rETH, if $\mathsf{q_R} \cdot \lambda \in o(n)$*, and* $\log t \cdot \log \frac{t}{\varepsilon} \cdot \log^{-1} \mathsf{q_S} \in o(n)$*, then* $\alpha + \beta(\log \frac{t}{\varepsilon}) \in \Omega(\log t \cdot \log \frac{t}{\varepsilon} / \log n)$*.*

That is, unless the commitment itself is large, the opening of subsets of size $\log \frac{t}{\varepsilon}$ must be large: about $\log t/\log n$ bits per element. SVCs are relatively a strong primitive as they imply SNARGs for NP via the Micali construction (the other direction is not known to hold). However, we only know how to derive lower bounds for them by a reduction to SNARGs. An interesting open question is to directly get lower bounds for SVC, presumably for a larger class of constructions. Moreover, we can hope to get a lower bound for SVCs (in the ROM) without assuming rETH (or any complexity assumption). Indeed, even P = NP is not known to yield trivial SVCs in the ROM (which is not the case for SNARGs).

1.1.1 Hitting High-Entropy Distributions

The crux of Lemma 1.2 proof is analyzing the completeness of the resulting low verifier query scheme. We manage to translate this challenge into the following task of hitting high-entropy distributions.

Let $X = (X_1, \ldots, X_m)$ be a random variable uniformly distributed over $(\{0,1\}^\lambda)^m$, let W be an event, and consider the random variable $X|_W$, i.e., X conditioned on W. It is instructive to think of this question as *"How does X appear to an adversary who received* $\log(1/\Pr[W])$ *bits of information about X?"* A long sequence of works have studied the question of how "close" $X|_W$ is to the uniformly distributed (unconditioned) X. In particular, these works considered the question of *indistinguishability*: showing that parts of $X|_W$ are close to being uniform. Some works, see [EIRS01; Raz98; SV10] to name a few, proved that the distribution of $(X|_W)_i$ is close in statistical distance to the uniform one, apart from a size $\log(1/\Pr[W])$ set of bad i's. Other works extended the above to bounded-query adversaries [Unr07; DGK17; CDGS18; GSV18; GLLZ20].

Unlike the above works, the focus of our result is *forgeability*: can we hit/sample from the conditional distribution $X|_W$ using a simple distribution? We show that after putting aside some bad indices, one can hit the support of $X|_W$, conditioned on its value in these bad indices, using a large enough product distribution. Like some of the above works, we state our result for high-entropy distributions, and not only for the uniform distribution conditioned on a high probability event.[6]

Theorem 1.4 (Hitting high-entropy distributions using product sets, informal)**.** *Let $X = (X_1, \ldots, X_m)$ be a random variable over the product set $(\{0,1\}^\lambda)^m$ with $H(X) \geq \lambda m - \ell$, and let $\lceil \log m \rceil \leq \gamma \leq \lambda$. Then with probability at least $1/2$ over $x \leftarrow X$, there exists an $O(\ell/\gamma)$-size set $\mathcal{B} \subseteq [m]$ (of bad indices) such that*

$$\Pr_{S \leftarrow (\{0,1\}^\gamma)^{m-|\mathcal{B}|}} \left[S \cap \mathsf{Supp}\big(X_{[m]\setminus\mathcal{B}} \mid X_{\mathcal{B}} = x_{\mathcal{B}}\big) \neq \emptyset \right] \in \Omega(1/\lambda m).$$

Letting H be the Shannon entropy function, and $v_{\mathcal{I}}$, for a vector v, denote the ordered vector $(v_i)_{i\in\mathcal{I}}$. That is, with high probability over $x \leftarrow X$, and after a few "bad" locations (indexed by $\mathcal{B}$) are exposed, one can hit (i.e., forge a sample from) the conditional distribution $X_{[m]\setminus\mathcal{B}} \mid X_{\mathcal{B}} = x_{\mathcal{B}}$ by sampling a *tiny*, in relative terms, product set.

Note that Theorem 1.4 does *not* state that $X_{[m]\setminus\mathcal{B}} \mid X_{\mathcal{B}} = x_{\mathcal{B}}$ is close to the uniform distribution. Actually, it might be very far from that, e.g., for $X = (U_1, \ldots, U_m) \mid \bigoplus U_i = 0^\lambda$ where the U_i's are uniform and independent random variables over $\{0,1\}^\lambda$, there is no choice of $\mathcal{B}$, apart from the trivial one of $\mathcal{B} = [m]$, that makes $X_{[m]\setminus\mathcal{B}} \mid X_{\mathcal{B}} = x_{\mathcal{B}}$ being close to uniform. It is also worth mentioning that one cannot prove Theorem 1.4 using the simple observation that after fixing some bad indices,

[6] This is a generalization since for uniformly distributed X it holds that $H(X \mid W) \geq \lambda m - \log 1/\Pr[W]$.

the projection of $X' \stackrel{\text{def}}{=} (X \mid X_{\mathcal{B}} = x_{\mathcal{B}})$ on all other coordinates has large support. While the latter guarantees that, with high probability, each random subset $S_i \leftarrow \{0,1\}^\gamma$ intersects the support of X'_i, appending these samples together does not necessarily form an element in X'. Rather, we prove the theorem by showing that the number of points in $S \cap \mathsf{Supp}(X'_{[m]\setminus\mathcal{B}})$ is well-concentrated around its mean.

In our application of Theorem 1.4, the event W is the proof sent by P being a fixed ℓ-bit value π, and the size of the bad set $\mathcal{B}$ translates to the query complexity of the new verier V'. The theorem yields, see Section 2, that if V' makes all queries is $\mathcal{B}$, and samples the potential answers for the other queries by itself, then it will accept (i.e., hitting the support of the accepting distribution) with good probability.

1.2 Related Work

1.2.1 SNARGs in the Random Oracle Model

There are several approaches to construct ROM-SNARGs. Micali [Mic00] (building on [Kil92; FS86]) showed a transformation that compiles a *probabilistically checkable proof* (PCP) and a commitment scheme into ROM-SNARG. Using the best know PCPs, the proof length of Micali's construction, to get (t, ε)-soundness, is $O((\log(t/\varepsilon))^2 \cdot \log n)$, where n is the instance size. Even when using the best-conjectured parameters for PCPs, known as the *Sliding Scale Conjecture* [BGLR93], the proof length remains the same up to the $\log n$ factors (see [CY21b] for a tight analysis of the Micali construction). Ben-Sasson, Chiesa, and Spooner [BCS16] (hereon BCS) transformed a public-coin *interactive oracle proofs* (IOPs) into ROM-SNARG. The benefit of their is approach is that we are much better at constructing IOPs, with good parameters, than PCPs. Still, even when using the best known (or conjectured) IOP, the proof length of the BCS construction remains $O((\log(t/\varepsilon))^2 \cdot \log n)$.

Recently, Chiesa and Yogev [CY21a] have constructed a ROM-SNARG of proof length of $O(\log(t/\varepsilon) \cdot \log t \cdot \log n)$, and hence slightly overcome the above "quadratic" barrier. Yet, the proof length of their construction is still far from the only (folklore) lower bound of $\Omega(\log(t/\varepsilon))$. Thus, the question of how to close this gap remains a major open question in this area.

1.2.2 SNARGs in Other Models

The security of a SNARGs is unlikely to be proven in a non-idealized model (using falsifiable assumptions) Gentry and Wichs [GW11], but if one is willing to rely on "more structured" non-falsifiable assumptions (in addition or instead of the random oracle), much shorter SNARGs become feasible. Treating t as the running time of the adversary, constructions that use *group-based and pairing-based assumptions* achieve the optimal length (or close to optimal) of $O(\log(t/\varepsilon))$ (c.f., [Gro10; GGPR13; BCIOP13; BCCGP16; BBBPWM18; BFS20; PGHR13; MBKM19; CHMMVW20; Set19]). These constructions are *insecure* against quantum adversaries. Lattices based constructions, which are plausibly post-quantum, either achieve *private-verifiability* [BISW17; BISW18; GMNO18; ISW21; Nit19], or are public-verifiability, but with large proof length in practice (moreover, they typically use a random oracle as an additional assumption) [BBCPGL18; BLNS20; BCS21; CMSZ21]. (All of the above works assume a common random or reference string.)

To date, relying on the ROM is the best way to construct SNARGs that overcome all of the drawbacks mentioned above (alas, at the price of larger proofs).

Paper Organization

In Section 2, we give a high-level overview of the techniques for proving Lemma 1.2 (from short ROM-SNARGs to short ROM-SNARGs with low verifier query complexity). A formal definition of our notion of salted soundness, along with notations, definitions, and general statements used throughout the paper are given in Section 3. Theorem 1.4 (hitting high-entropy events using product sets) is proved in Section 4. Theorem 1.1 (lower bound on the length of ROM-SNARGs) and its accompanied Lemma 1.2 are proved in Section 5, and Theorem 1.3 (lower bound on the length of ROM subvector commitments) is proved in Section 6.

2 Techniques

In this section, we give a high-level overview of our proof for Lemma 1.2, explaining how to transform a short salted-soundness, perfect completeness, deterministic non-adaptive verifier ROM-SNARG into a low verifier query ROM-SNARG for the same language.

Fix a deterministic non-adaptive ROM-SNARG $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ for a language $\mathcal{L}$ with (t, ε)-slated-soundness and perfect completeness. Let s denote the proof length ARG, and let q_{P} and q_{V} denote the query complexity of P and V, respectively.

2.1 Warmup

As a warmup, assume that the honestly generated proof π, sent by P, only contains information about outputs of k ("important") queries, whose identity is independent of the oracle. (The proof might contain additional information depending only on the instance $\mathbb{x}$ and the witness $\mathbb{w}$.) For this simple scenario, the construction of a k-query V' is rather straightforward:

Algorithm 2.1 (Low-query verifier V'. Warmup)**.**

```
Oracle: $\zeta\colon \{0,1\}^\lambda \mapsto \{0,1\}^\lambda$.
Input: Instance $\mathbb{x}$ and a proof $\pi$.
Operation:
  1. Emulate $\mathsf{V}(\mathbb{x}, \pi)$ till it produces a list of oracle queries $(w_1, \ldots, w_{\mathsf{q}_{\mathsf{V}}})$. (Recall that $\mathsf{V}$ is non-adaptive.)
  2. Sample a random $k$-size subset $\mathcal{J} \subseteq [\mathsf{q}_{\mathsf{V}}]$.
  3. For $i = 1, \ldots, \mathsf{q}_{\mathsf{V}}$:
       If $i \in \mathcal{J}$, set $y_i = \zeta(w_i)$.
       Otherwise, sample $y_i \leftarrow \{0,1\}^\lambda$.
  4. Accept if $\mathsf{V}$ accepts on the emulation with $(y_1, \ldots, y_{\mathsf{q}_{\mathsf{V}}})$ as the answers to its oracle queries
```

Namely, V' guesses the identity of the important queries, and then uses the oracle ζ to answer them. It samples the answers to the other queries uniformly at random. The query complexity of V' is small if the number of important queries is small. Let us quickly argue about the completeness and soundness of $\mathsf{ARG}' = (\mathsf{P}, \mathsf{V}')$.

- Completeness. If the set $\mathcal{J}$ happens to contain all important queries, then the given the proof π, the instance $\mathbb{x}$, and the witness $\mathbb{w}$, the oracle answers provided to the emulated V have *exactly* the same distribution as in its non-emulated execution. Since we assume ARG has perfect completeness, the completeness of ARG' is at least $1/\left|\binom{\mathsf{q_V}}{k}\right|$—the probability that $\mathcal{J}$ contains all important queries.

- Soundness: Here we rely on the salted soundness of the original SNARG scheme. Assume there exists a $(t-\mathsf{q_V})$-query cheating prover $\widetilde{\mathsf{P}'}$ that makes V' accepts $\mathbb{x} \notin \mathcal{L}$ with probability ε. Consider the following t-query cheating prover $\widetilde{\mathsf{P}}$ for violating the salted-soundness of ARG.[7]

  1. Run $\widetilde{\mathsf{P}'}^{\zeta}$ to generate a proof π.
     Emulate $\mathsf{V}(\mathbb{x}, \pi)$ till it produces a list of oracle queries $(w_1, \ldots, w_{\mathsf{q_V}})$.
  2. For $i = 1, \ldots, \mathsf{q_V}$:
     Query ζ on w_i with a fresh salt. Set $S_i = \{y_i\}$ for y_i be the query answer.
     If w_i was asked by $\widetilde{\mathsf{P}'}$ in Step 1, add the retrieved answer to S_i.
  3. If there exists $(y_1, \ldots, y_{\mathsf{q_V}}) \in S_1 \times \ldots \times S_{\mathsf{q_V}}$ that would make V accept $(\mathbb{x}, \pi)$ with $(y_1, \ldots, y_{\mathsf{q_V}})$ as the answers to its oracle queries, program $\zeta(w_i) = y_i$ for each $i \in [\mathsf{q_V}]$ (this programming is allowed by the salted soundness security game).
  4. Output π.

  By definition, if $\widetilde{\mathsf{P}}$ outputs a proof π then V accepts π on the programmed oracle. In addition, the probability that $\widetilde{\mathsf{P}}$ outputs the proof π generated in Step 1, is at least as large as the probability that V' accepts π on the non-programmed oracle: $\widetilde{\mathsf{P}}$ considers for each query the original output of the oracle, as seen by V' on queries in $\mathcal{J}$, and a uniform output, as sampled by V' on inputs not in $\mathcal{J}$.

2.2 Actual Scenario

Things get way more challenging when the proof π depends on the queries made by P, even in a slightly more complicated way. For instance, suppose π contains the XOR of some k queries, and V verifies that the XOR of these queries is consistent with π. Since k might be arbitrarily large, i.e., much larger than π, there is no low-query verifier that makes all these queries. So the challenge is to design a verifier that does not make all queries that effect the value of π, but still has non-trivial soundness and completeness.

The key observation is that for the general case, where π depends *arbitrarily* on all oracle answers, we can modify the verifier so that the completeness and soundness are not that different from the naïve example considered in the warmup. Very informally, with high probability over the value of π and apart from $k = s/\gamma$ "important" queries, the verification verdict does not depend "too much" on the answer to all other "non-important" queries. That is, there are many possible answers for the non-important queries that lead to acceptance (compared with *all* possible answers in the warmup case). See Section 2.3 for details. It follows that the answers for the non-important queries

[7] Recall that the salted-soundness game allows a cheating prover to *resample* (many times) the output of the random oracle on a query. Each resampling costs the cheating prover a single query call from its query budget. The prover can role-back the oracle on certain queries, to set their answers to a previously answered values. See Section 3.5.1 for exact definition.

can be emulated by the verifier (without querying the oracle). Equipped with this understanding, the low query V' is defined as follows:

Algorithm 2.2 (Low-query verifier V')**.**

```
Oracle: $\zeta\colon \{0,1\}^\lambda \mapsto \{0,1\}^\lambda$.
Paramters: $\gamma < \lambda$.
Input: Instance $\mathbb{x}$ and a proof $\pi$.
Operation:
  1. Emulate $\mathsf{V}(\mathbb{x},\pi)$ till it produces a list of oracle queries $(w_1,\ldots,w_{\mathsf{q_V}})$. (Recall that $\mathsf{V}$ is non-adaptive.)
  2. Sample a random $k = \lceil s/\gamma \rceil$-size subset $\mathcal{J} \subseteq [\mathsf{q_V}]$.
  3. For $i = 1,\ldots,\mathsf{q_V}$:
       If $i \in \mathcal{J}$, set $S_i = \{\zeta(w_i)\}$.
       Otherwise, let $S_i$ be a $2^\gamma$-size *random* subset of $\{0,1\}^\lambda$.
  4. Accept if there exists $(y_1,\ldots,y_{\mathsf{q_V}}) \in S_1 \times \ldots \times S_{\mathsf{q_V}}$ that make $\mathsf{V}$ accepts on the emulation, with $(y_1,\ldots,y_{\mathsf{q_V}})$ as the answers to its oracle queries
```

That is, similar to the warmup scenario, V' only uses the oracle to answer the $k = \lceil s/\gamma \rceil$ queries in the guessed set $\mathcal{J}$. For each other query, V' samples 2^γ candidates answers. It accepts if there is a choice from the candidate answers that jointly with the oracle answers to the queries in $\mathcal{J}$, leads to acceptance. The running-time of V' is (roughly) $2^{\mathsf{q_V}\cdot\gamma}$, and the following claim states the completeness and soundness of $\mathsf{ARG}' = (\mathsf{P},\mathsf{V}')$:

Claim 2.3 (Informal)**.** ARG' *has* $\left(\lambda \cdot \mathsf{q_P} \cdot \binom{\mathsf{q_V}}{s/\gamma}\right)^{-1}$*-completeness and* $(t - \mathsf{q_V} \cdot 2^\gamma, \varepsilon)$*-soundness.*

We argue completeness in Section 2.3, using the observation we made above regarding the small number of important queries, and argue soundness in Section 2.4, by extending the approach we took for proving soundness in the warmup case.

2.3 Completeness

Let Π and $Y = (Y_1,\ldots,Y_{\mathsf{q_P}})$ denote the proof and the random oracle answers to honest prover P queries on instance $\mathbb{x}$ and witness $\mathbb{w}$, respectively. Since the Y_i's are independent uniform values in $\{0,1\}^\lambda$, it holds that

$$H(Y) = \mathsf{q_P} \cdot \log \lambda \tag{1}$$

where $H(Y)$ is the Shannon entropy of Y. A standard entropy argument yields that with probability at least $1/2$ over $\pi \leftarrow \Pi$:

$$H(Y \mid \Pi = \pi) \geq \mathsf{q_P} \cdot \lambda - 2|\pi| \tag{2}$$

In the following, fix $\pi \in \mathsf{Supp}(\Pi)$ for which Equation (3) holds. Applying Theorem 1.4 with respect to $Y|_{\Pi=\pi}$ and $\ell = 2|\pi|$, yields that with probability $1/2$ over the value of $(y_1, \ldots, y_{\mathsf{q}_\mathsf{P}}) \leftarrow Y|_{\Pi=\pi}$, there exists a set $\mathcal{B} \subseteq [\mathsf{q}_\mathsf{P}]$ of size ℓ/γ (omitting constant factors) such that

$$\Pr\left[(S_1 \times \cdots \times S_{\mathsf{q}_\mathsf{P}-|\mathcal{B}|}) \cap \mathsf{Supp}(Y'_{[\mathsf{q}_\mathsf{P}]\setminus\mathcal{B}}) \neq \emptyset\right] \in \Omega(1/\lambda \cdot \mathsf{q}_\mathsf{P}) \tag{3}$$

where each of the S_i's is an independent 2^γ-size subset of $\{0,1\}^\lambda$, $Y' \overset{\text{def}}{=} Y|_{Y_\mathcal{B}=y_\mathcal{B},\Pi=\pi}$, and $Y'_\mathcal{I}$ is the ordered vector $(Y'_i)_{i\in\mathcal{I}}$.

Assume for simplify that V and P make exactly the same queries. By Equation (3), if the random set $\mathcal{J}$ (sampled by V') contains all queries indexed by $\mathcal{B} = \mathcal{B}(\pi)$, then with probability $\Omega(1/\lambda \cdot \mathsf{q}_\mathsf{P})$ over the choice of the sets S_i's sampled by V', exit answers $\{y_j \in S_j\}_{j\notin\mathcal{J}}$ that when combined with the oracle answers $\{y_j \in S_j\}_{j\in\mathcal{J}}$, it holds that $y = (y_1, \ldots, y_{\mathsf{q}_\mathsf{P}}) \in \mathsf{Supp}(Y|_{\Pi=\pi})$. Since such a vector y is *possible* to occur as random oracle answers in an honest execution of P that results in π, the perfect completeness of ARG yields that V accepts on (the answers in) y with probability one. We conclude that V' accepts with probability $\Omega(1/\lambda \cdot \mathsf{q}_\mathsf{P})$ times $\Pr[\mathcal{J} = \mathcal{B}] \geq 1/\binom{\mathsf{q}_\mathsf{V}}{s/\gamma}$.

Improving the completeness.. We note that one could slightly modify the transformation to improve the completeness significantly (at the cost of proof length and prover running time). However, as this does not improve our lower bound, we only sketch the idea here. Instead of having the verifier guess the set $\mathcal{J}$, let the prover find $\mathcal{J}$, and send its description to the verifier. The completeness error now would come only from the error in Equation (3) (i.e., an error of $(\lambda \cdot \mathsf{q}_\mathsf{P})^{-1}$), and not from the probability of choosing the right set $\mathcal{J}$. The proof would be slightly larger (as it needs to contain the description of $\mathcal{J}$), and the running-time of the honest prover would increase, as it needs to find the right set $\mathcal{J}$ (query complexity will stay the same). Even more so, using a prefix salt for all queries (included in the proof), one can make the completeness error exponentially small.

2.4 Soundness

Assume there exists a $(t - \mathsf{q}_\mathsf{V} \cdot 2^\gamma)$-query cheating prover $\widetilde{\mathsf{P}'}$ that makes V' accepts $\mathbb{x} \notin \mathcal{L}$ with probability ε, and consider the following t-query cheating prover $\widetilde{\mathsf{P}'}$ for violating the salted-soundness of ARG.

Algorithm 2.4 ($\widetilde{\mathsf{P}}$)**.**

```
Oracle: $\zeta \colon \{0,1\}^\lambda \mapsto \{0,1\}^\lambda$.

Input: Instance $\mathbb{x}$.

1. Run $\widetilde{\mathsf{P}'}^{\zeta}(\mathbb{x})$ to generate a proof $\pi$.
2. Emulate $\mathsf{V}$ on $(\mathbb{x}, \pi)$ to determine its list of oracle queries $(w_1, \ldots, w_{\mathsf{q}_\mathsf{V}})$.
3. For $i = 1, \ldots, \mathsf{q}_\mathsf{V}$:
    (a) Query $\zeta$ on $w_i$ for $2^\gamma$ times. Let $S_i$ be the set of answers.
    (b) If $w_i$ was asked by $\widetilde{\mathsf{P}'}$ in Step 1, add the retrieved answer to $S_i$.
4. If there exists $(y_1, \ldots, y_{\mathsf{q}_\mathsf{V}}) \in S_1 \times \ldots \times S_{\mathsf{q}_\mathsf{V}}$ that make $\mathsf{V}$ accept $(\mathbb{x}, \pi)$ with $(y_1, \ldots, y_{\mathsf{q}_\mathsf{V}})$ as the
   answers to its oracle queries, program $\zeta(w_i) = y_i$ for each $i \in [\mathsf{q}_\mathsf{V}]$.
```

5. Output π.

The cheating probability of $\widetilde{\mathsf{P}}$ it as least as high as that of $\widetilde{\mathsf{P}'}$. This is shown via a coupling argument, and the precise details are given in Section 5.2.2.

3 Preliminaries

3.1 Notations

We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for values and functions. Let poly stand for the set of all polynomials. Throughout the paper, log is the base 2 logarithm. For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. Given a vector $v \in \Sigma^n$, let v_i denote its ith entry. Similarly, for a set $\mathcal{I} \subseteq [n]$, let $v_{\mathcal{I}}$ be the *ordered sequence* $(v_i)_{i \in \mathcal{I}}$, let $v_{-\mathcal{I}} \stackrel{\text{def}}{=} v_{[n]\setminus\mathcal{I}}$. For a set $\mathcal{S}$ and $k \in \mathbb{N}$, let $\mathcal{P}_k(\mathcal{S})$ denote all k-size subsets of $\mathcal{S}$. The *support* of a random variable X, denoted $\mathsf{Supp}(X)$, is defined as $\{x \colon \Pr[X = x] > 0\}$. For an event E, we write $X|_E$ to denote the random variable X conditioned on E.

The language 3SAT over n variables is the set of all satisfiable formulas in conjunctive normal form where each clause is limited to at most three literals. The class BPTIME$[T]$ refers to all languages that can be decided by a probabilistic TM that runs in time $T(n)$, on inputs of length n.

Some basic inequalities. We use the following well-known facts:

Fact 3.1. $\log(1-x) \leq -x$ *for* $x \in [0,1]$*, and* $\log(1-x) \geq -2x$*, for any* $x \in [0,1/2]$.

Theorem 3.2 (Chebyshev inequality)**.** *For any finite non-negative random variable X it holds that* $\Pr[X > 0] \geq \mathrm{E}[X]^2/\mathrm{E}[X^2]$.

3.2 Entropy Measures

We refer to several measures of entropy. The relation and motivation of these measures are best understood by considering a notion that we will refer to as the sample-entropy: for a random variable X and $x \in \mathsf{Supp}(X)$, the *sample-entropy* of x with respect to X is the quantity

$$H_X(x) \stackrel{\text{def}}{=} \log \tfrac{1}{\Pr[X=x]},$$

letting $H_X(x) = \infty$ for $x \notin \mathsf{Supp}(X)$, and $2^{-\infty} = 0$.

The sample-entropy measures the amount of “randomness" or “surprise" in the specific sample x, assuming that x has been generated according to X. Using this notion, we can define the *Shannon entropy* $H(X)$ and *min-entropy* $\mathrm{H}_\infty(X)$ as follows:

$$H(X) \stackrel{\text{def}}{=} \mathrm{E}_{x \leftarrow X}\left[H_X(x)\right], \quad \mathrm{H}_\infty(X) \stackrel{\text{def}}{=} \min_{x \in \mathsf{Supp}(X)} H_X(x).$$

We will also discuss the *max-entropy* $H_0(X) \stackrel{\text{def}}{=} \log|\mathsf{Supp}(X)|$. The term “max-entropy” and its relation to the sample-entropy will be made apparent below.

It can be shown that $\mathrm{H}_\infty(X) \leq H(X) \leq H_0(X)$ with each inequality being an equality if and only if X is flat (uniform on its support). Thus, saying that $\mathrm{H}_\infty(X) \geq k$ is a strong way of saying that X has “high entropy” and $H_0(X) \leq k$ a strong way of saying that X has “low entropy”.

Conditional entropies. We will also be interested in conditional versions of entropy. For jointly distributed random variables (X, Y) and $(x, y) \in \mathsf{Supp}(X, Y)$, we define the *conditional sample-entropy* to be $H_{X|Y}(x|y) = \log \frac{1}{\Pr_{X|Y}[x|y]} = \log \frac{1}{\Pr[X=x|Y=y]}$. Then the standard *conditional Shannon entropy* can be written as

$$H(X \mid Y) = \mathrm{E}_{(x,y)\leftarrow(X,Y)} \left[H_{X|Y}(x \mid y)\right] = \mathrm{E}_{y\leftarrow Y} \left[H(X|_{Y=y})\right] = H(X, Y) - H(Y).$$

The following known fact bounds the effect conditioning on an event reduce the entropy of uniformly distributed random variable.

Fact 3.3. *Let X be a random variable uniform over a set $\mathcal{S}$ and let W be an event. Then $H(X \mid W) \geq \log(|\mathcal{S}|) - \log 1/\Pr[W]$.*

3.3 Randomized Exponential Time Hypothesis

Definition 3.4 (rETH; [DHMTW14])**.** *The* randomized Exponential Time Hypothesis *(rETH) states that there exist $\varepsilon > 0$ and $c > 1$ such that* 3SAT *on n variables and with $c \cdot n$ clauses cannot be solved by probabilistic algorithms that run in time $2^{\varepsilon \cdot n}$.*

3.4 Random Oracles

We denote by $\mathcal{U}(\lambda)$ the uniform distribution over all functions $\zeta \colon \{0,1\}^* \to \{0,1\}^\lambda$. Given an oracle algorithm A and an oracle $\zeta \in \mathcal{U}(\lambda)$, $\mathsf{queries}(A, \zeta)$ is the set of oracle queries that A^ζ makes. We say that A is *t-query* if $|\mathsf{queries}(A, \zeta)| \leq t$ for every $\zeta \in \mathcal{U}(\lambda)$. We say that A is *non-adaptive* if its queries do not depend on the responses of the random oracle to previous queries. Finally, we consider the length of oracle queries, i.e., the number of bits used to specify the query: we say that A has queries of length λ if for every $\zeta \in \mathcal{U}(\lambda)$ and $x \in \mathsf{queries}(\mathsf{A}, \zeta)$ it holds that $|x| \leq \lambda$.

3.5 Non-Interactive Arguments in the ROM

We consider non-interactive arguments in the ROM, where security holds against query-bounded, yet possibly computationally-unbounded, adversaries. Recall that a non-interactive argument typically consists of a prover algorithm and a verifier algorithm that prove and validate statements for a binary relation, which represents the valid instance-witness pairs.

A pair of polynomial-time oracle algorithms $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ is a ROM-SNARG with α-completeness and (t, ϵ)-soundness, for a relation $\mathcal{R}$, if the following holds.

- **Completeness.** For every $\lambda \in \mathbb{N}$ and $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$:

$$\Pr_{\substack{\zeta\leftarrow\mathcal{U}(\lambda) \\ \pi\leftarrow\mathsf{P}^\zeta(\mathbb{x},\mathbb{w})}} \left[\mathsf{V}^\zeta(\mathbb{x}, \pi) = 1\right] \geq \alpha(|\mathbb{x}|, \lambda) \ .$$

- **Soundness.**[8] For every $\lambda \in \mathbb{N}$, t-query $\widetilde{\mathsf{P}}$ and $\mathbb{x} \notin \mathcal{L}(\mathcal{R})$:

[8] This notion, where $\mathbb{x}$ is set before the oracle, is sometimes refereed to as *non-adaptive soundness*. Clearly, lower bounds on this weaker notion , as we do in this work, apply also for its adaptive variant (where the cheating prover is allowed to choose $\mathbb{x}$ as a function of the oracle).

$$\Pr_{\substack{\zeta \leftarrow \mathcal{U}(\lambda) \\ \pi \leftarrow \widetilde{\mathsf{P}}^{\zeta}}} \left[\mathsf{V}^{\zeta}(\mathbb{x}, \pi) = 1 \right] \geq \epsilon(|\mathbb{x}|, \lambda, t) \ .$$

Complexity measures. We consider several complexity measures beyond soundness error. All of these complexity measures are, implicitly, functions of $\mathbb{x}$ and the security parameter λ.

- *argument length*: $\mathsf{s} := |\pi|$.
- *times*: the prover P runs in time pt; the verifier V runs in time vt.
- *queries*: the prover P is a t-query algorithm the verifier V is a q_{V}-query algorithm.

3.5.1 Salted Soundness

Chiesa and Yogev [CY20] introduced a stronger notion of soundness for ROM-SNARG that they named salted soundness. This notion requires soundness to hold also against a malicious prover that has limited ability to *program* the oracle: it can obtain a set of random, independent strings as candidates for random oracle answers to a specific query. After obtaining such sets to the queries of his choice, the malicious prover can pick an answer of his desire from each set to be the random oracle answer.[9] This notion is formalized via the following *salted soundness game* defined as follows:

Game 3.5 ($\mathsf{SaltedSoundess}_{\mathsf{V},\lambda,t}(\mathsf{A}, \mathbb{x})$)**.**

Parameters: Algorithm V and $\lambda, t \in \mathbb{N}$.

Input: $\mathbb{x} \in \{0,1\}^*$

Player: A.

Operation:

1. Initialize keyed-map S of lists (each entry is initialized with the empty list).
2. Repeat the following t times:
   (a) A sends a query $x \in \{0,1\}^*$.
   (b) Send $y \leftarrow \{0,1\}^{\lambda}$ to A, and add it to the list $S[x]$.
3. A outputs a proof string π and query-answer list $\sigma = [(x_1, y_1), \ldots, (x_n, y_n)]$.
4. Abort if $y_i \notin S[x_i]$ for some $i \in [n]$.
5. Output $\mathsf{V}^{\zeta_\sigma}(\mathbb{x}, \pi)$.

Definition 3.6 (Salted soundness)**.** *We say that ROM-SNARG* (P, V) *has* (t, ε)*-salted-soundness for a language* $\mathcal{L}$*, if for any* λ*,* $\mathbb{x} \notin \mathcal{L}$ *and* $\widetilde{\mathsf{P}}$ *it holds that* $\Pr\left[\mathsf{SaltedSoundess}_{\mathsf{V},\lambda,t}(\widetilde{\mathsf{P}}, \mathbb{x}) = 1\right] \leq \varepsilon(|\mathbb{x}|, \lambda, t)$.

Remark 3.7 (Known constructions satisfy salted soundness)**.** Known constructions of ROM-SNARGs are usually proven to have standard soundness (as opposed to salted soundness). However, we observe that the constructions of [Mic00; BCS16; CY21b; CY21a] actually achieve this stronger notion of security. In particular, the tight analysis given in [CY21b] and in [CY21a] explicitly allowed the adversary to choose a salt for each query in the construction (e.g., see remark 3.2 in [CY21b]).

[9] Our notion slightly strengthens the notion of Chiesa and Yogev [CY20], in which the prover cannot roll back the oracle answer to a previously seen answer.

Amplification. It turns out that salted soundness can be easily amplified (at the expense of the query complexity). Lemma 3.8 is proved in Appendix A.

Lemma 3.8. *Let* ARG *be an ROM-SNARG for a language* $\mathcal{L}$ *with* (t, ε)*-salted-soundness for* $\varepsilon \leq 1/4$. *Then* ARG *has* $(t/k, 2\varepsilon/k)$*-salted-soundness for any* $k \in \mathbb{N}$, .

4 Hitting High-Entropy Distribution using Product Sets

In this section we formally state and prove Theorem 1.4. Recall that for a set $\mathcal{S}$ and $k \in \mathbb{N}$, we let $\mathcal{P}_k(\mathcal{S})$ denote all k-size subsets of $\mathcal{S}$. Thus, a uniform sample from $(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda))^{m-|\mathcal{B}|}$ is a random product in $(\{0,1\}^\lambda)^{m-|\mathcal{B}|}$ of width 2^γ.

Theorem 4.1 (Hitting high-entropy distributions using product sets, restatement of Theorem 1.4)**.** *Let* $\gamma \leq \lambda \in \mathbb{N}$, *and let* $X = (X_1, \ldots, X_m)$ *be a random variable over* $(\{0,1\}^\lambda)^m$. *If* $H(X) \geq \lambda m - \ell$ *and* $\gamma \geq 4\lceil \log m \rceil + 4$, *then with probability at least* $1/2$ *over* $x \leftarrow X$, *then there exists a set* $\mathcal{B} \subseteq [m]$ *of size at most* $8\ell/\gamma + 4$ *such that for any* $\mathcal{B}' \supseteq \mathcal{B}$ *it holds that*

$$\Pr_{S \leftarrow \left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^{m-|\mathcal{B}'|}} \left[S \cap \mathsf{Supp}(X_{[m]\setminus\mathcal{B}'}\,|_{X_{\mathcal{B}'}=x_{\mathcal{B}'}}) \neq \emptyset\right] \geq 1/32\lambda m.$$

Remark 4.2 (Tightness of Theorem 4.1)**.** The size of $\mathcal{B}$ in Theorem 4.1 is tight up to a constant: Let $m, \lambda, \gamma \in \mathbb{N}$ be as in Theorem 4.1, let $X = (X_1, \ldots, X_m)$ be uniform over $(\{0,1\}^\lambda)^m$ and let W be the event that $X_1 = \ldots = X_t = 0^\lambda$, for some $t \in [m]$. Clearly, $H(X|_W) = (m-t)\lambda$. It is also clear that for every x and every set $\mathcal{B} \subseteq [m]$ of size $t' < t$, it holds that

$$\Pr_{S \leftarrow \left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^{m-t'}} \left[S \cap \mathsf{Supp}(X_{m\setminus\mathcal{B}}\,|_{X_{\mathcal{B}}=x_{\mathcal{B}}}) \neq \emptyset\right] \leq 2^{\gamma-\lambda},$$

which is negligible for sufficiently small γ, e.g., $\gamma = \lambda/2$. This matches, up to a constant, Theorem 4.1, which states that with high probability over $x \leftarrow X\,|_W$, there exists a set $\mathcal{B}$ of size at most $16t + 4$ for which that the above event occurs with probability at least $1/32\lambda m$.

Proving Theorem 4.1. We start with describing the high-level approach for the proof for the case $\mathcal{B}' = \mathcal{B}$. We need to prove that with high probability over $x \leftarrow X$, exits a small (i.e., with size at most $8\ell/\gamma + 4$) subset $\mathcal{B} \subseteq [m]$ such that

$$\Pr_{S \leftarrow \left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^{\widehat{m}}} \left[S \cap \mathsf{Supp}(\widehat{X}) \neq \emptyset\right] \geq 1/32\lambda m,$$

for $\widehat{X} = X_{[m]\setminus\mathcal{B}}\,|_{X_{\mathcal{B}}=x_{\mathcal{B}}}$ and $\widehat{m} = m - |\mathcal{B}|$. We assume, without loss of generality, that the elements of each S_i are chosen in a uniform order, and denote the jth element of S_i, according to this order, by $S_i[j]$. For $y = (y_1, \ldots, y_{\widehat{m}}) \in [2^\gamma]^{\widehat{m}}$, let $S^y \in \{0,1\}^{\lambda \times \widehat{m}}$ be the random variable defined by $(S^y)_i = S_i[y_i]$. Let Z^y be the indicator for the event $S^y \in \mathsf{Supp}(\widehat{X})$, and let $Z \overset{\text{def}}{=} \sum_{y \in [2^\gamma]^{\widehat{m}}} Z^y$. That is, Z^y is event that the yth element of S is in $\mathsf{Supp}(\widehat{X})$. Given this notation, we need to prove that $\Pr[Z > 0] \geq 1/32\lambda m$. We start by proving that the expected value of Z is large. By linearity of expectation,

$$\mathrm{E}[Z] = \sum_{y \in [2^\gamma]^{\widehat{m}}} \mathrm{E}[Z^y] = 2^{\gamma\widehat{m}} \cdot |\mathsf{Supp}(\widehat{X})|/2^{\widehat{m}\lambda} = 2^{(\gamma-\lambda)\widehat{m}} \cdot |\mathsf{Supp}(\widehat{X})| \tag{4}$$

To guarantee that $\mathrm{E}[Z]$ is at least one, we chose $\mathcal{B}$ to be a *maximal* subset of $[m]$ with

$$H_{X_{\mathcal{B}}}(x_{\mathcal{B}}) \leq (\lambda - \gamma) \cdot |\mathcal{B}| \tag{5}$$

for $H_Y(y)$ be the *sample entropy of y according to Y* (see Section 3.2). It is rather straightforward to show that with respect to this choice of $\mathcal{B}$, the expected value of Z is indeed at least one. Furthermore, since, by assumption, X has high entropy, the expected size of $\mathcal{B}$, as a function of x, is small, and therefore, with high probability over x the size of $\mathcal{B}$ is also small. (See proof in Lemma 4.3).

The above would suffice for lower-bounding $\Pr[Z > 0]$, if the random variables $\{Z^y\}$ would have been independent. This, however, is clearly not the case since most Z^y are not even pairwise independent: for a pair $y, y' \in [2^\gamma]^{\widehat{m}}$ with $y_{\mathcal{I}} = y'_{\mathcal{I}}$ for some $\mathcal{I} \subseteq [\widehat{m}]$, the event $Z^y = 1$, implying $(S^{y'})_{\mathcal{I}} \in \mathsf{Supp}(\widehat{X}_{\mathcal{I}})$, is likely to increase the probability of $Z^{y'} = 1$. Yet, we manage to show that the expected value of Z^2 is small enough, implying that Z is well concentrated around its mean, and therefore $\Pr[Z > 0]$ is large. To do that, we notice that for the maximal set $\mathcal{B}$ defined above, it holds that

$$H_{X_{\mathcal{I}}|_{X_{\mathcal{B}}=x_{\mathcal{B}}}}(x_{\mathcal{I}}) > (\lambda - \gamma) \cdot |\mathcal{I}| \tag{6}$$

for *every* $\mathcal{I} \subseteq [m] \setminus \mathcal{B}$. This condition implies that for every y, y' with $y_{\mathcal{I}} = y'_{\mathcal{I}}$, the probability of $Z^y \wedge Z^{y'}$ is sufficiently small (quantified by the size of $\mathcal{I}$), implying that $\mathrm{E}\left[Z^2\right]$ is small.

Moving to the formal proof, Theorem 4.1 is an immediate corollary of the following two lemmata: Lemma 4.3 states that with high probability over x, there exists a small set $\mathcal{B}$ for which Equation (6) holds, and Lemma 4.4 completes the job by proving the conclusion of the theorem for the random variable $X_{[m]\setminus\mathcal{B}}\,|_{X_{\mathcal{B}}=x_{\mathcal{B}}}$.

Lemma 4.3 (High-entropy events have an almost full-entropy large projection)**.** *Let $\gamma \leq \lambda \in \mathbb{N}$, and let $X = (X_1, \ldots, X_m)$ be a random variable over $(\{0,1\}^\lambda)^m$. If $H(X) \geq \lambda \cdot m - \ell$ and $\gamma \geq 2 \cdot \lceil \log m \rceil + 2$, then with probability at least $1/2$ over $x \leftarrow X$, exists a set $\mathcal{B} \subseteq [m]$ of size at most $4\ell/\gamma + 4$ such that for every $\mathcal{I} \subseteq [m] \setminus \mathcal{B}$:*

$$H_{X_{\mathcal{I}}|_{X_{\mathcal{B}}=x_{\mathcal{B}}}}(x_{\mathcal{I}}) \geq (\lambda - \gamma)\,|\mathcal{I}|\,.$$

Lemma 4.4 (Hitting almost full-entropy events using product sets)**.** *Let $\gamma \leq \lambda \in \mathbb{N}$, let $X = (X_1, \ldots, X_m)$ be a random variable over $(\{0,1\}^\lambda)^m$. Assume $\gamma \geq 2 \cdot \lceil \log m \rceil + 2$, and that for every $x \in \mathsf{Supp}(X)$ and $\mathcal{I} \subseteq [m]$, it holds that $H_{X_{\mathcal{I}}}(x_{\mathcal{I}}) \geq (\lambda - \gamma/2) \cdot |\mathcal{I}|$. Then*

$$\Pr_{S \leftarrow \left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^m}\left[S \cap \mathsf{Supp}(X) \neq \emptyset\right] \geq 1/32\lambda m.$$

We prove Lemmas 4.3 and 4.4 in Sections 4.1 and 4.2, receptively, but first use them for proving Theorem 4.1.

Proof of Theorem 4.1: Let $t \overset{\text{def}}{=} 8\ell/\gamma + 4$, and let

$$\mathcal{T} \overset{\text{def}}{=} \{x \in \mathsf{Supp}(X) :\ \exists \mathcal{B} \subseteq [m], |\mathcal{B}| \leq t \colon \forall \mathcal{I} \subseteq [m] \setminus \mathcal{B}, H_{X_{\mathcal{I}}|_{X_{\mathcal{B}}=x_{\mathcal{B}}}}(x_{\mathcal{I}}) \geq (\lambda - \gamma/2) \cdot |\mathcal{I}|\}$$

Since, by assumption, $\gamma/2 \geq 2\,\lceil \log m \rceil + 2$, Lemma 4.3 yields that

$$\Pr[X \in \mathcal{T}] \geq 1/2 \tag{7}$$

Fix $x \in \mathcal{T}$, let $\mathcal{B}$ be the set guaranteed by the definition of $\mathcal{T}$ (choose an arbitrary one, if there is more than one), let $\mathcal{B}' \supseteq \mathcal{B}$, let $X' \stackrel{\text{def}}{=} X_{[m]\setminus\mathcal{B}'}|_{X_{\mathcal{B}'}=x_{\mathcal{B}'}}$, and let $m' \stackrel{\text{def}}{=} m - |\mathcal{B}'|$. By Lemma 4.4

$$\Pr_{S\leftarrow\left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^{m'}}\left[S \cap \mathsf{Supp}(X') \neq \emptyset\right] \geq 1/32\lambda m' \geq 1/32\lambda m \tag{8}$$

Combing Equations (7) and (8), concludes the proof. □

4.1 High-Entropy Distributions Have an (Almost) Uniform Large Projection, Proving Lemma 4.3

Proof of Lemma 4.3. Let m, λ, γ and X be as in Lemma 4.3. For $x \in \mathsf{Supp}(X)$, let $\mathcal{B}^x$ be the (lex. first) *maximal*[10] subset of $[m]$ with

$$H_{X_{\mathcal{B}^x}}(x_{\mathcal{B}^x}) \leq (\lambda - \gamma)\,|\mathcal{B}^x| \tag{9}$$

Since Equation (9) holds for the empty set, $\mathcal{B}^x$ is always defined. We prove Lemma 4.3 using the following two claims, proven below.

Claim 4.5. *For every $x \in \mathsf{Supp}(X)$ and $\mathcal{I} \subseteq [m] \setminus \mathcal{B}^x$, it holds that $H_{X_{\mathcal{I}}|X_{\mathcal{B}^x}=x_{\mathcal{B}^x}}(x_I) \geq (\lambda - \gamma)\cdot|\mathcal{I}|$.*

Claim 4.6. *If $H(X) \geq \lambda \cdot m - \ell$, then for every random variable $I \subseteq [m]$ it holds that $H(X_I \mid I) \geq (\lambda - \lceil\log m\rceil) \cdot \mathrm{E}\left[|I|\right] - \ell - \lceil \log m\rceil$.*

By Claim 4.5, for every $x \in \mathsf{Supp}(X)$ and $\mathcal{I} \subseteq [m] \setminus \mathcal{B}^x$, it holds that

$$H_{X_{\mathcal{I}}|X_{\mathcal{B}^x}=x_{\mathcal{B}^x}}(x_I) \geq (\lambda - \gamma)\,|\mathcal{I}| \tag{10}$$

Hence, to conclude the proof, it is left to argue that with high probability over $x \leftarrow X$, the size of $\mathcal{B}^x$ is small. For $\mathcal{I} \subseteq [m]$, let $f_{\mathcal{I}}(x) = x_{\mathcal{I}}$ if $\mathcal{B}^x = \mathcal{I}$, and $f_{\mathcal{I}}(x) = \perp$ otherwise, and let $p_{\mathcal{I}} = \Pr\left[f_{\mathcal{I}}(X) = \perp\right]$. Compute

[10] Maximal means relative to inclusion—there is no $\mathcal{I}$ strictly containing $\mathcal{B}^x$ with $H_{X_{\mathcal{I}}}(x_{\mathcal{I}}) \leq (\lambda - \gamma)\cdot|\mathcal{I}|$.

$$
\begin{aligned}
H(X_{\mathcal{B}^X} \mid \mathcal{B}^X) &= \mathrm{E}_{\mathcal{B} \leftarrow \mathcal{B}^X} \left[H(X_{\mathcal{B}} \mid \mathcal{B}^X = \mathcal{B}) \right] \\
&= \mathrm{E}_{\mathcal{B} \leftarrow \mathcal{B}^X} \left[H(f_{\mathcal{B}}(X) \mid \mathcal{B}^X = \mathcal{B}) \right] \\
&\leq \sum_{\mathcal{I}} \mathrm{E}_{\mathcal{B} \leftarrow \mathcal{B}^X} \left[H(f_{\mathcal{I}}(X) \mid \mathcal{B}^X = \mathcal{B}) \right] \\
&= \sum_{\mathcal{I}} H(f_{\mathcal{I}}(X) \mid \mathcal{B}^X) \\
&\leq \sum_{\mathcal{I}} H(f_{\mathcal{I}}(X)) \\
&= \sum_{\mathcal{I}} \Big(\sum_{x \colon \mathcal{B}^x = \mathcal{I}} \Pr\left[X = x\right] \cdot H_{X_{\mathcal{I}}}(x_{\mathcal{I}}) \Big) + p_{\mathcal{I}} \cdot \log(1/p_{\mathcal{I}}) \\
&\leq \sum_{\mathcal{I}} \Pr\left[\mathcal{B}^X = \mathcal{I} \right] \cdot (\lambda - \gamma) \cdot |\mathcal{I}| + p_{\mathcal{I}} \cdot \log(1/p_{\mathcal{I}}) \\
&= (\lambda - \gamma) \mathrm{E}\left[\left| \mathcal{B}^X \right| \right] + \sum_{\mathcal{I}} p_{\mathcal{I}} \cdot \log(1/p_{\mathcal{I}}) \\
&\leq (\lambda - \gamma) \mathrm{E}\left[\left| \mathcal{B}^X \right| \right] + \sum_{\mathcal{I}} 2 \cdot (1 - p_{\mathcal{I}}) \\
&\leq (\lambda - \gamma) \mathrm{E}\left[\left| \mathcal{B}^X \right| \right] + 2.
\end{aligned} \tag{11}
$$

The third equality and the second inequality, hold by the chain rule of Shannon entropy. The third inequality holds by the definition of $\mathcal{B}^x$.

On the other hand since, by assumption, $H(X) \geq \lambda \cdot m - \ell$, Claim 4.6 yields that

$$
H(X_{\mathcal{B}^X} \mid \mathcal{B}^X) \geq (\lambda - \lceil \log m \rceil) \cdot \mathrm{E}\left[\left| \mathcal{B}^X \right| \right] - \ell - \lceil \log m \rceil \tag{12}
$$

Combining Equations (11) and (12), we conclude that $\mathrm{E}\left[\left| \mathcal{B}^X \right| \right] \leq \frac{\ell + \lceil \log m \rceil + 2}{\gamma - \lceil \log m \rceil} \leq 2\ell/\gamma + 2$, where the 2nd inequality follows from the fact that $\gamma \geq 2 \cdot \lceil \log m \rceil + 2$. The proof follows by Markov inequality. □

Proving Claim 4.5.

Proof of Claim 4.5. Let $\mathcal{B} = \mathcal{B}^x$. Since for every disjoint sets $\mathcal{A}, \mathcal{C} \subseteq [m]$ and $x \in \mathsf{Supp}(X)$

$$
\Pr[X_{\mathcal{A}} = x_{\mathcal{A}}] \cdot \Pr[X_{\mathcal{C}} = x_{\mathcal{C}} \mid X_{\mathcal{A}} = x_{\mathcal{A}}] = \Pr[X_{\mathcal{A} \cup \mathcal{C}} = x_{\mathcal{A} \cup \mathcal{C}}],
$$

for every $\mathcal{I} \subseteq [m] \setminus \mathcal{B}$

$$
H_{X_{\mathcal{B}}}(x_{\mathcal{B}}) + H_{X_{\mathcal{I}} | X_{\mathcal{B}} = x_{\mathcal{B}}}(x_{\mathcal{I}}) = H_{X_{\mathcal{I} \cup \mathcal{B}}}(x_{\mathcal{I} \cup \mathcal{B}}).
$$

Assume towards a contradiction that $H_{X_{\mathcal{I}} | X_{\mathcal{B}} = x_{\mathcal{B}}}(x_{\mathcal{I}}) < (\lambda - \gamma) \left| \mathcal{I} \right|$. Since, by definition, $H_{X_{\mathcal{B}}}(x_{\mathcal{B}}) \leq (\lambda - \gamma) \left| \mathcal{B} \right|$, it follows that

$$
H_{X_{\mathcal{I} \cup \mathcal{B}}}(x_{\mathcal{I} \cup \mathcal{B}}) < (\lambda - \gamma) \cdot (|\mathcal{B}| + |\mathcal{I}|) = (\lambda - \gamma) \cdot \left| \mathcal{B} \cup \mathcal{I} \right|,
$$

in contradiction to the maximality of $\mathcal{B}$. □

Proving Claim 4.6.

Proof. Since, by assumption, $H(X) \geq \lambda m - \ell$, and since

$$H(I) = H(I, |I|) \leq \lceil \log m \rceil + H(I \mid |I|) \leq \lceil \log m \rceil + \mathrm{E}\left[|I|\right] \cdot \lceil \log m \rceil = \lceil \log m \rceil \left(\mathrm{E}\left[|I|\right] + 1\right),$$

we conclude that

$$H(X \mid I) \geq \lambda m - \ell - \left(\mathrm{E}_{x \leftarrow X}[|I|] + 1\right) \lceil \log m \rceil \tag{13}$$

By data-processing inequality,

$$H(X \mid I) \leq H(X_I, X_{[m]\setminus I} \mid I) \leq H(X_I \mid I) + H(X_{[m]\setminus I} \mid I) \tag{14}$$

Finally, since $H(X_{[m]\setminus I} \mid I) \leq H_0(X_{[m]\setminus I}) \mid I) \leq \lambda \cdot (m - \mathrm{E}_{x \leftarrow X}[|I|])$, we conclude that

$$\begin{aligned} H(X_I \mid I) \geq & \lambda \cdot m - \ell - \lceil \log m \rceil \left(\mathrm{E}\left[|I|\right] + 1\right) - \lambda \cdot \left(m - \mathrm{E}\left[|I|\right]\right) \\ = & (\lambda - \lceil \log m \rceil) \cdot \mathrm{E}\left[|I|\right] - \ell - \lceil \log m \rceil . \end{aligned}$$

□

4.2 Hitting almost Full-Entropy Distributions using Product Set, Proving Lemma 4.4

We start by proving the following variant of Lemma 4.4, stated for *flat* distributions, i.e., X is uniform over a set. In Section 4.2.1, we use this variant for proving Lemma 4.4.

Lemma 4.7 (Hitting flat distributions)**.** *Let $m, \gamma \leq \lambda \in \mathbb{N}$ be such that $\gamma \geq 2 \cdot \lceil \log m \rceil + 2$, let $\delta > 0$, and let $\mathcal{T} \subseteq \{0,1\}^{\lambda \cdot m}$ be a non empty set. If for all $\mathcal{I} \subseteq [m]$ and $a \in \{0,1\}^{\lambda \cdot |\mathcal{I}|}$, it holds that*

$$|\{x \in \mathcal{T} \colon x_{\mathcal{I}} = a\}| \leq |\mathcal{T}| \cdot 2^{(\gamma/2 - \lambda)|\mathcal{I}|}/\delta \ , \tag{15}$$

then

$$\Pr_{S \leftarrow \left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^m}\left[S \cap \mathcal{T} \neq \emptyset\right] \geq \delta/2 \ .$$

Proof. Let $S = (S_1, \ldots, S_m)$ be as in the lemma statement, i.e., uniformly distributed over $\left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^m$. We assume, without loss of generality, that the elements of each S_i are chosen in a uniform order and denote the jth element of S_i, according to this order, by $S_i[j]$. For $y = (y_1, \ldots, y_m) \in [2^\gamma]^m$, let $S^y \in \{0,1\}^{\lambda \times m}$ be the random variable defined by $(S^y)_i \overset{\text{def}}{=} S_i[y_i]$. Let Z^y be the indicator for the event $S^y \in \mathcal{T}$, and let $Z \overset{\text{def}}{=} \sum_{y \in [2^\gamma]^m} Z^y$. By Chebyshev inequality, Theorem 3.2, it holds that

$$\Pr_{S \leftarrow \left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^m}\left[S \cap \mathcal{T} \neq \emptyset\right] = \Pr[Z > 0] \geq \mathrm{E}[Z]^2/\mathrm{E}[Z^2] \tag{16}$$

Thus, we prove Lemma 4.7 by properly bounding $\mathrm{E}[Z]$ and $\mathrm{E}[Z^2]$. Let $\rho \overset{\text{def}}{=} \frac{|\mathcal{T}|}{2^{m\lambda}}$. Since we associate a random order with the elements of each S_i, for every $y \in [2^\gamma]^m$ it holds that $\mathrm{E}\left[Z^y\right] = \rho$. Hence,

$$\mathrm{E}\left[Z\right] = \sum_{y \in [2^\gamma]^m} \mathrm{E}\left[Z^y\right] = 2^{\gamma m} \rho \tag{17}$$

For upperbounding $\mathrm{E}[Z^2]$, we use the following claim (proved in Section 4.2). In the following for $y, y' \in [2^\gamma]^m$, let $\mathcal{K}_{y,y'} \overset{\text{def}}{=} \{i \in [m] : y_i = y'_i\}$.

Claim 4.8. *For every $y, y' \in [2^\gamma]^m$ it holds that* $\Pr[Z^y \wedge Z^{y'}] \le 2^{\gamma \cdot |\mathcal{K}_{y,y'}|/2} \cdot \rho^2/\delta$.

For $\mathcal{K} \subseteq [m]$, let $\mathcal{A}_\mathcal{K} \overset{\text{def}}{=} \{(y, y') \in [2^\gamma]^m \colon \mathcal{K}_{y,y'} = \mathcal{K}\}$. Using Claim 4.8, we deuce that

$$\begin{aligned}
\mathrm{E}\left[Z^2\right] &= \sum_{y,y' \in [2^\gamma]^m} \Pr[Z^y \wedge Z^{y'}] \qquad (18)\\
&= \sum_{\mathcal{K} \subseteq [m]} \sum_{y,y' \in \mathcal{A}_\mathcal{K}} \Pr[Z^y \wedge Z^{y'}]\\
&\le \sum_{\mathcal{K} \subseteq [m]} \sum_{y,y' \in \mathcal{A}_\mathcal{K}} 2^{\gamma|\mathcal{K}|/2} \cdot \rho^2/\delta\\
&\le \frac{\rho^2}{\delta} \cdot \sum_{k=0}^{m} \sum_{\mathcal{K} \subseteq [m], |\mathcal{K}|=k} 2^{\gamma k} \cdot (2^{2\gamma})^{m-k} \cdot 2^{\gamma k/2}\\
&= \frac{\rho^2}{\delta} \cdot 2^{2\gamma m} \cdot \sum_{k=0}^{m} \binom{m}{k} \cdot 2^{-\gamma k/2}\\
&\le \frac{\rho^2}{\delta} \cdot 2^{2\gamma m} \cdot \sum_{k=0}^{m} 2^{-k \cdot (\gamma/2 - \log m)} \le 2 \cdot \frac{\rho^2}{\delta} \cdot 2^{2\gamma m}.
\end{aligned}$$

The first inequality holds by Claim 4.8, and the last one by holds since, by assumption, $\gamma \ge 2 \cdot \lceil \log m \rceil + 2$. Combining Equations (16) to (18), prove the lemma by deducing that

$$\Pr[Z > 0] \ge \frac{\mathrm{E}[Z]^2}{\mathrm{E}[Z^2]} \ge \frac{(2^{\gamma m} \cdot \rho)^2}{2 \cdot \frac{\rho^2}{\delta} \cdot 2^{2\gamma m}} = \delta/2.$$

□

Proving Claim 4.8.

Proof of Claim 4.8. Let $\mathcal{K} = \mathcal{K}_{y,y'}$, and for $a \in \{0,1\}^{\lambda|\mathcal{K}|}$ let $\mathcal{T}_a = \{x \in \mathcal{T} \colon x_\mathcal{I} = a\}$. Compute

$$\begin{aligned}
\Pr\left[Z^y \wedge Z^{y'}\right] &= \sum_{a \in \{0,1\}^{\lambda \cdot |\mathcal{K}|}} \Pr\left[S^y_\mathcal{K} = a\right] \cdot \Pr\left[Z^y \wedge Z^{y'} \mid S^y_\mathcal{K} = a\right]\\
&= \sum_{a \in \{0,1\}^{\lambda \cdot |\mathcal{K}|}} \Pr\left[S^y_\mathcal{K} = a\right] \cdot \left(\frac{|\mathcal{T}_a| \cdot (|\mathcal{T}_a| - 1)}{2^{2\lambda(m-|\mathcal{K}|)}}\right)\\
&\le \sum_{a \in \{0,1\}^{\lambda \cdot |\mathcal{K}|}} 2^{-\lambda|\mathcal{K}|} \cdot \left(\frac{|\mathcal{T}|}{2^{\lambda(m-|\mathcal{K}|)}}\right)^2 \cdot \left(\frac{|\mathcal{T}_a|}{|\mathcal{T}|}\right)^2\\
&\le \sum_{a \in \{0,1\}^{\lambda|\mathcal{K}|}} 2^{-\lambda|\mathcal{K}|} \cdot \left(\frac{|\mathcal{T}|}{2^{\lambda(m-|\mathcal{K}|)}}\right)^2 \cdot \frac{|\mathcal{T}_a|}{|\mathcal{T}|} \cdot 2^{(\gamma/2-\lambda)\cdot|\mathcal{K}|}/\delta\\
&= \frac{1}{\delta} \cdot \left(\frac{|\mathcal{T}|}{2^{\lambda m}}\right)^2 \cdot 2^{\gamma|\mathcal{K}|/2} \cdot \sum_{a \in \{0,1\}^{\lambda|\mathcal{K}|}} \frac{|\mathcal{T}_a|}{|\mathcal{T}|}\\
&= \frac{1}{\delta} \cdot \rho^2 \cdot 2^{\gamma|\mathcal{K}|/2}.
\end{aligned}$$

The second inequality holds by the assumption of the lemma (Equation (15)). □

4.2.1 Proving Lemma 4.4

Proof of Lemma 4.4. Define

$$\mathcal{T} \overset{\text{def}}{=} \{x \in \mathsf{Supp}(X) \colon \forall \mathcal{I} \subseteq [m], H_{X_{\mathcal{I}}}(x_{\mathcal{I}}) \geq (\lambda - \gamma/2) \cdot |\mathcal{I}|\}$$

We partition the set $\mathcal{T}$ into $2\lambda m$ subsets, such that the elements of each part have roughly the same probability under X. Specifically, for $i \in [2\lambda m]$ let

$$\mathcal{T}^i \overset{\text{def}}{=} \{x \in \mathcal{T} \colon H_X(x) \in [i-1, i)\},$$

and let $\mathcal{T}^0 \overset{\text{def}}{=} \{x \in \mathcal{T} \colon H_X(x) \geq 2\lambda m\}$. By definition,

$$\Pr[X \in \mathcal{T}^0] = \sum_{x \in \mathcal{T}^0} \Pr[X = x] \leq 2^{\lambda \cdot m} \cdot 2^{-2 \cdot \lambda \cdot m} = 2^{-\lambda \cdot m},$$

and therefore $2^{-\lambda \cdot m} + \sum_{i \in [2 \cdot \lambda \cdot m]} \Pr[X \in \mathcal{T}^i] \geq 1$. Hence, by averaging argument, exists $i \in [2\lambda m]$ such that

$$\Pr[X \in \mathcal{T}^i] \geq \frac{1 - 2^{-\lambda \cdot m}}{2\lambda m} \geq \frac{1}{4\lambda m} \tag{19}$$

The second inequality hold since, by assumption, $\lambda \geq \gamma \geq 2$. In the rest of the proof we use Lemma 4.7 to prove that $\Pr_{S \leftarrow \mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)} \left[S \cap \mathcal{T}^i \neq \emptyset\right]$. Let $X^i = X \mid_{X \in \mathcal{T}^i}$, and for $\mathcal{I} \subseteq [m]$ and $a \in \mathsf{Supp}(X^i_{\mathcal{I}})$, let $\mathcal{T}^i_{\mathcal{I},a} \overset{\text{def}}{=} \{x \in \mathcal{T}^i \colon x_{\mathcal{I}} = a\}$. Since X^i is almost flat, for every $a \in \mathsf{Supp}(X^i_{\mathcal{I}})$ and $x \in \mathcal{T}^i_{\mathcal{I},a}$:

$$\Pr[X^i_{\mathcal{I}} = a] = \sum_{x' \in \mathcal{T}^i_{\mathcal{I},a}} \Pr[X^i = x'] \geq \left|\mathcal{T}^i_{\mathcal{I},a}\right| \cdot \Pr[X^i = x]/2.$$

Similarly,

$$\begin{aligned} 1 = \sum_{a \in \mathsf{Supp}(X^i_{\mathcal{I}})} \Pr[X^i_{\mathcal{I}} = a] &= \sum_{a \in \mathsf{Supp}(X^i_{\mathcal{I}})} \sum_{x' \in \mathcal{T}^i_{\mathcal{I},a}} \Pr[X^i = x'] \\ \leq \sum_{a \in \mathsf{Supp}(X^i_{\mathcal{I}})} \left|\mathcal{T}^i_{\mathcal{I},a}\right| \cdot 2 \cdot \Pr[X^i = x] &= 2 \cdot \left|\mathcal{T}^i\right| \cdot \Pr[X^i = x]. \end{aligned}$$

Combing the above two inequalities, we get that

$$\Pr[X^i_{\mathcal{I}} = a] \geq \frac{1/2 \cdot \left|\mathcal{T}^i_{\mathcal{I},a}\right| \cdot \Pr[X^i = x]}{2 \cdot |\mathcal{T}^i| \cdot \Pr[X^i = x]} = \frac{\left|\mathcal{T}^i_{\mathcal{I},a}\right|}{4 \cdot |\mathcal{T}^i|} \tag{20}$$

By assumption, for every $x \in \mathcal{T}$ and $\mathcal{I} \subseteq [m]$:

$$\Pr[X_{\mathcal{I}} = x_{\mathcal{I}}] \leq 2^{(\gamma/2 - \lambda)|\mathcal{I}|} \tag{21}$$

Therefore, for every $a \in \mathsf{Supp}(X^i_{\mathcal{I}})$:

$$\frac{\left|\mathcal{T}^i_{\mathcal{I},a}\right|}{|\mathcal{T}^i|} \leq 4 \cdot \Pr[X^i_{\mathcal{I}} = a] = 4 \cdot \frac{\Pr[X_{\mathcal{I}} = a]}{\Pr[X \in \mathcal{T}^i]} \leq 16\lambda m \cdot 2^{(\gamma/2-\lambda)|\mathcal{I}|} \tag{22}$$

The first inequality holds by Equation (20), and the second by Equation (21). Applying Lemma 4.7 for the set $\mathcal{T}^i$ with parameter $\delta = 1/16\lambda m$, yields that

$$\Pr_{S \leftarrow \mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)}\left[S \cap \mathcal{T}^i \neq \emptyset\right] \geq \frac{1}{32\lambda m},$$

and we deduce that $\Pr_{S \leftarrow \mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)}\left[S \cap \mathsf{Supp}(X) \neq \emptyset\right] \geq \frac{1}{32\lambda m}$. □

5 Lower Bound on the Length of ROM-SNARGs

In this section, we present our lower bound on the proof length of ROM-SNARGs, formally stated below (see Definition 3.4 for the formal definition of rETH, and Section 3.5 for that of salted-soundness ROM-SNARGs).

Theorem 5.1 (Conditional lower bound on ROM-SNARGs length)**.** *Let* $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ *be an* s*-length ROM-SNARG for* n*-variable* 3SAT*, with* (t, ε)*-salted-soundness, perfect completeness, and deterministic non-adaptive verifier. Let* $\mathsf{q_P}$ *and* $\mathsf{q_V}$ *be the query complexity of* P *and* V*, respectively, let* v *denotes* V*'s running time, and let* λ *denote the random oracle input and output length. Assuming rETH, if*

1. $\varepsilon \leq 1/4$*;*
2. $t^{1/2}/\varepsilon \geq n$*.*
3. $\mathsf{q_V} \cdot \lambda \in o(n)$*;*
4. $\log t \cdot \log \frac{t}{\varepsilon} / \log \mathsf{q_P} \in o(n)$*;*
5. $\mathsf{q_V} \leq t^{1/10}$*; and*
6. $v \in 2^{o(n)}$*,*

then $\mathsf{s} \geq 2^{-15} \cdot \log t \cdot \log \frac{t}{\varepsilon} / \log \mathsf{q_P}$*.*

Theorem 5.1 is proved using the following two lemmata. Lemma 5.2 states that the verifier query complexity of a short ROM-SNARG can be significantly reduced, and Lemma 5.3, taken from [CY20], states that the existence of a low verifier query complexity ROM-SNARGs contradicts rETH.

Lemma 5.2 (Short ROM-SNARGs → Low Query ROM-SNARGs)**.** *Let* $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ *be as in Theorem 5.1, then for any* $\gamma \in \mathbb{N}$*, there exists a verifier* V' *such that* $\mathsf{ARG}' \overset{\text{def}}{=} (\mathsf{P}, \mathsf{V}')$ *is a ROM-SNARG for* $\mathcal{L}$ *with the following properties:*

1. *completeness* $\left(128 \cdot \lambda \cdot \mathsf{q_P} \cdot \binom{\mathsf{q_V}}{20 \cdot \lceil \mathsf{s}/\gamma \rceil}\right)^{-1}$*,*
2. $(t - \mathsf{q_V} \cdot 2^\gamma, \varepsilon)$*-soundness,*
3. *verifier query complexity* $O(\lceil \mathsf{s}/\gamma \rceil)$*, and*
4. *verifier running time* $O(2^{\mathsf{q_V} \cdot \log t} \cdot v)$*.*

Furthermore, the transformation from V *to* V' *is efficient (in the description length of* V*).*

In words, Lemma 5.2 states that there exists a generic transformation from short ROM-SNARGs into the same length ROM-SNARGs with low verifier query complexity (but worse completeness and soundness). Lemma 5.2 is proven in Section 5.2.

While not explicit in their work, the following lemma follows by similar arguments to the main proof in [CY20] (see details Appendix C).

Lemma 5.3 (Follows from [CY20])**.** *Let* $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ *be a* (t, ε)*-sound ROM-SNARG for* n*-variable* 3SAT *with random oracle (input and output) length* λ*, argument length* s*, and let* $\mathsf{q_V}$ *and* $\mathsf{q_P}$ *denote* P*'s and* V*'s query complexity, respectively. Assume*

1. $\mathsf{s} + \lambda \cdot \mathsf{q_V} \in o(n)$*;*
2. $\mathsf{q_V} \leq 1/10 \cdot \log(1/\varepsilon) \cdot \log^{-1} \mathsf{q_P}$*;*
3. *completeness* $\geq \varepsilon^{1/2}$*;*
4. $\log(1/\varepsilon) \leq o(n^{1/2})$*; and*
5. V*'s running time* $2^{o(n)}$*,*

then $3\mathrm{SAT} \in \mathrm{BPTIME}[2^{o(n)}]$.

Note that Lemma 5.3 does not require V to be deterministic or non adaptive.

5.1 Proof of Theorem 5.1

Proof of Theorem 5.1. Suppose we are given a SNARG ARG for 3SAT that satisfies the conditions of the theorem, and assume without loss of generality that $\mathsf{q_P} \leq t^{1/10}$. (Otherwise, for $\mathsf{q_P} > t^{1/10}$), the lower bound we need to prove can be written as $\mathsf{s} \geq 2^{-10} \cdot \log \frac{t}{\varepsilon}$, which follows by Theorem B.1.) Assume towards contradiction that $\mathsf{s} \leq 2^{-15} \cdot \log t \cdot \log \frac{t}{\varepsilon} / \log \mathsf{q_P}$. Theorem 5.1 si proved via the following steps:

1. Apply Lemma 3.8 with $k = t^{0.5}$, to get that ARG has (t', ε')-salted-soundness, for $t' = t^{1/2}$, and $\varepsilon' = 2\varepsilon/t^{1/2}$.

   Set $\gamma =$ for a large (universal) constant c' that will be defined later.
2. Apply Lemma 5.2 with $\gamma = c' \log t$, c' will be defined later, to get a ROM-SNARG ARG' for 3SAT with the following parameters:
   (a) completeness $\left(128 \cdot \lambda \cdot \mathsf{q_P} \cdot \binom{\mathsf{q_V}}{c \cdot \lceil \mathsf{s}/\gamma \rceil}\right)^{-1}$ for a universal constant $c > 0$;
   (b) $(t' - \mathsf{q_V} \cdot 2^{\gamma}, \varepsilon')$-soundness.
   (c) verifier query complexity $16 \cdot \lceil \mathsf{s}/\gamma \rceil + 4$; and
   (d) verifier running time $O(2^{\mathsf{q_V} \cdot \log t} \cdot v)$.
3. Apply Lemma 5.3 on ARG' to contradict rETH. For this, we need to verify that all of the conditions of the lemma apply. Indeed,
   (a) $\mathsf{s} + \lambda \cdot \mathsf{q_V} \in o(n)$: Holds since $\mathsf{s} \leq 2^{-15} \cdot \log t \cdot \log \frac{t}{\varepsilon} / \log \mathsf{q_P} \in o(n)$, and we assumed that $\lambda \cdot \mathsf{q_V} \in o(n)$.

(b) $\mathsf{q}_{\mathsf{V}} \leq 1/10 \cdot \log(1/\varepsilon') \cdot \log^{-1} \mathsf{q}_{\mathsf{P}}$: the query complexity of the verifier of ARG' is

$$\begin{aligned}\mathsf{q}_{\mathsf{V}} = 16\left\lceil \mathsf{s}/\gamma \right\rceil + 4 &= 16 \cdot \left(\frac{2^{-15} \cdot \log t \cdot \log \frac{t}{\varepsilon} / \log \mathsf{q}_{\mathsf{P}}}{1/10 \cdot \log t}\right) + 4 \\ &\leq 1/10 \cdot \log \frac{t}{\varepsilon} \cdot \log^{-1} \mathsf{q}_{\mathsf{P}} = 1/10 \cdot \log \frac{1}{\varepsilon'} \cdot \log^{-1} \mathsf{q}_{\mathsf{P}} \ .\end{aligned}$$

(c) completeness $\geq \varepsilon'^{1/2}$: Observe that $16\left\lceil \mathsf{s}/\gamma \right\rceil + 4 \leq 1/10 \cdot \log(t/\varepsilon) \cdot \log^{-1} \mathsf{q}_{\mathsf{P}}$. Thus, the completeness of our scheme satisfies:

$$\begin{aligned}&\left(128 \cdot \lambda \cdot \mathsf{q}_{\mathsf{P}} \cdot \binom{\mathsf{q}_{\mathsf{V}}}{c \cdot \left\lceil \mathsf{s}/\gamma \right\rceil}\right)^{-1} \\ &\geq 2^{-7-\log\lambda-\log \mathsf{q}_{\mathsf{P}}-\log \mathsf{q}_{\mathsf{V}} \cdot 1/10 \cdot \log(t/\varepsilon) \cdot \log^{-1} \mathsf{q}_{\mathsf{P}}} \\ &\geq 2^{-7-1/10\cdot\log t-1/10\cdot\log t-1/10\cdot\log(t/\varepsilon)} \\ &\geq 2^{-1/2\cdot\log(2t^{1/2}/\varepsilon)} \\ &= 2^{-1/2\cdot\log(1/\varepsilon')} \\ &\geq \varepsilon'^{1/2} \ .\end{aligned}$$

(d) $\log(1/\varepsilon') \leq o(n^{1/2})$: we observe that $\log(1/\varepsilon') = 1/2\log(2t/\varepsilon) \geq \log n$.

(e) V's running time $2^{o(n)}$: The verifier running time of the scheme is $O(2^{\mathsf{q}_{\mathsf{V}} \cdot \log t} \cdot v)$. Since $\mathsf{q}_{\mathsf{V}} \cdot \log t = o(n)$ and $v = 2^{o(n)}$, its the total running time is $2^{o(n)}$.

4. We conclude that 3SAT $\in$ BPTIME$[2^{o(n)}]$, contradicting rETH.

□

5.2 Short ROM-SNARGs to Low Query ROM-SNARGs, Proving Lemma 5.2

In this section, we prove Lemma 5.2 (see Section 2 for a high-level overview of the proof). Let $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ be ROM-SNARG with (t, ε)-salted soundness, random oracle of length λ, a non-adaptive deterministic verifier, prover query complexity q_{P}, and verifier query complexity q_{V}. The low query verifier V' is defined as follows:

Algorithm 5.4 (Low-query verifier V')**.**

```
Oracle: $\zeta \colon \{0,1\}^\lambda \mapsto \{0,1\}^\lambda$.
Parameter: $\gamma \leq \lambda$. Let $k = 20\left\lceil \mathsf{s}/\gamma \right\rceil$.
Input: Instance $\mathbb{x}$ and proof $\pi$.
Operation:

 1. Emulate $\mathsf{V}$ on $(\mathbb{x}, \pi)$ to get a list of queries $w = (w_1, \ldots, w_{\mathsf{q}_{\mathsf{V}}})$.

 2. Uniformly sample a $k$-size subset $\mathcal{J} \subseteq [\mathsf{q}_{\mathsf{V}}]$.

 3. For each $i \in [\mathsf{q}_{\mathsf{V}}]$:
        If $i \in \mathcal{J}$, set $S_i = \{\zeta(w_i)\}$.
        Otherwise, let $S_i$ be a $2^\gamma$-size random subset of $\{0,1\}^\lambda$.
```

4. Accept if there exists $(y_1, \ldots, y_{\mathsf{q_V}}) \in S_1 \times \ldots \times S_{\mathsf{q_V}}$ that make V accepts given $(y_1, \ldots, y_{\mathsf{q_V}})$ as answers to its oracle queries.

It is easy to observe that V' has the desired query complexity and running time. Thus, it is left to prove that $\mathsf{ARG}' = (\mathsf{P}, \mathsf{V}')$ has the desired completeness and soundness. The completeness of ARG' is analyzed in Section 5.2.1 and its soundness in Section 5.2.2. We put things together in Section 5.2.3

5.2.1 Completeness

We prove the following lower bound on the completeness of ARG'.

Claim 5.5. ARG' *has completeness* $\geq \left(128 \cdot \lambda \cdot \mathsf{q_P} \cdot \binom{\mathsf{q_V}}{20 \cdot \lceil \mathsf{s}/\gamma \rceil}\right)^{-1}$.

In the following, we assume for simplicity that the V's queries are (always) a subset of the P's queries. (The proof without this assumption follows very similar lines, though with more complicated notation. Also, one could always modify the honest prover to perform all the verifier's queries, this comes with a negligible cost that has no effect on our results.)

Proof. We associate the following random variable with the probability space defined by the choice of ζ over the (honest) execution of $(\mathsf{P}^{\zeta}(\mathbb{w}), \mathsf{V}'^{\zeta})(\mathbb{x})$: denote P's queries by $X = (X_1, \ldots, X_{\mathsf{q_P}})$, define $Z = (Z_1, \ldots, Z_{\mathsf{q_P}})$ by $Z_i = \zeta(X_i)$, and let Π denote the proof sent by P. We assume for ease of notation that the queries that V would have made on the proof Π are just $X_1, \ldots, X_{\mathsf{q_V}}$.

The length of Π is s, thus a standard argument yields that

$$H(\Pi) \leq H_0(\Pi) \leq \mathsf{s} \ .$$

Since each Z_i is a bit string of length λ (recall that λ is the output length of ζ), it holds that

$$H(Z \mid \Pi) \geq H(Z) - H(\Pi) \geq \lambda \cdot \mathsf{q_P} - \mathsf{s} \ .$$

Since (by definition) $H(Z \mid \Pi) = \mathrm{E}_{\pi \leftarrow \Pi}[H(Z \mid \Pi = \pi)]$, with probability at least $1/2$ over $\pi \leftarrow \Pi$, it holds that $H(Z \mid \Pi = \pi) \geq \lambda \cdot \mathsf{q_P} - 2 \cdot \mathsf{s}$. Fix any such proof π, and let $Y = (Y_1, \ldots, Y_{\mathsf{q_P}}) = Z \mid_{\Pi = \pi}$. For $\ell = 2 \cdot \mathsf{s}$, it holds that

$$H(Y) \geq \lambda \cdot \mathsf{q_P} - \ell \ .$$

Applying Theorem 4.1 on Y yields that with probability $1/2$ over $y \leftarrow Y$ there exists a subset $\mathcal{B} \subseteq [\mathsf{q_P}]$ with $|\mathcal{B}| \leq \lfloor 8\ell/\gamma \rfloor + 4$ such that for any subset $\mathcal{B}' \supseteq \mathcal{B}$:

$$\Pr_{S \leftarrow \left(\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda)\right)^{\mathsf{q_P} - |\mathcal{B}'|}} \left[S \cap \mathsf{Supp}(Y \mid_{Y_{\mathcal{B}'} = y_{\mathcal{B}'}}) \neq \emptyset\right] \geq \frac{1}{32 \cdot \lambda \cdot \mathsf{q_P}} \tag{23}$$

An immediate corollary of Equation (23) is that with probability at least $1/2$ over the choice of $y \leftarrow Y$, the following holds the following process outputs one with probability at least $\frac{1}{32 \cdot \lambda \cdot \mathsf{q_P}}$:

1. For each $i \in [\mathsf{q_V}]$:

   If $i \in \mathcal{B}'$, set $S_i = \{y_i\}$.

   Otherwise, let S_i be a 2^γ-size *random* subset of $\{0,1\}^\lambda$.

2. Output 1 if $(S_1 \times \ldots \times S_{\mathsf{q_V}}) \cap \mathsf{Supp}((Y \mid_{Y_{\mathcal{B}'}=y_{\mathcal{B}'}})_{[\mathsf{q_V}]}) \neq \emptyset$.

The perfect completeness of ARG yields that for any $\pi \in \mathsf{Supp}(\Pi)$, $\mathsf{V}(\mathbb{x}, \pi)$ accepts on any value of $z \in \mathsf{Supp}((Y = Z_{\Pi=\pi})_{[\mathsf{q_V}]})$ given as oracle answers. Thus, it accepts any value of $z \in \mathsf{Supp}((Y \mid_{Y_{\mathcal{B}'}=y_{\mathcal{B}'}})_{[\mathsf{q_V}]})$ for any $y \in \mathsf{Supp}(Y)$.

We deduce that V' accepts with this probability, assuming that the set $\mathcal{J}$ sampled contains $\mathcal{B} \cap [\mathsf{q_V}]$. Noting that $|\mathcal{B}| \leq \lfloor 8\ell/\gamma \rfloor + 4 = \lfloor 16\mathsf{s}/\gamma \rfloor + 4 \leq 20 \lceil \mathsf{s}/\gamma \rceil = k$, the latter happens with probability at least $1/\binom{\mathsf{q_V}}{k}$. We conclude that V' accepts with probability at least

$$\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{32 \cdot \lambda \cdot \mathsf{q_P}} \cdot \frac{1}{\binom{\mathsf{q_V}}{k}} \geq \left(128 \cdot \lambda \cdot \mathsf{q_P} \cdot \binom{\mathsf{q_V}}{20 \cdot \lceil \mathsf{s}/\gamma \rceil}\right)^{-1} .$$

□

5.2.2 Soundness

We prove the following upper bound on the soundness error of ARG'.

Claim 5.6. ARG' *has* $(t - \mathsf{q_V} \cdot 2^{\gamma}, \varepsilon)$*-soundness.*

Proof. Let $\widetilde{\mathsf{P}'}$ be a $t' := t - 2^{\gamma \cdot \mathsf{q_V}}$-query cheating prover such that

$$\Pr\left[\langle \widetilde{\mathsf{P}'}, \mathsf{V}'(\mathbb{x}) \rangle = 1\right] > \varepsilon$$

for some $\mathbb{x} \notin \mathcal{L}$. We show how to use $\widetilde{\mathsf{P}'}$ to construct the following t-query cheating prover $\widetilde{\mathsf{P}}$ such that

$$\Pr\left[\mathsf{SaltedSoundess}_{\mathsf{V},\lambda,t'}(\widetilde{\mathsf{P}}, \mathbb{x}) = 1\right] > \varepsilon \ ,$$

violating the assumed salted-soundness of (P, V).

We assume without loss of generality that $\widetilde{\mathsf{P}'}$ is deterministic. Indeed, since $\widetilde{\mathsf{P}}$ is computationally unbounded (it is only bounded by its query complexity to the random oracle), it has sufficient time to enumerate all random strings and choose the best one.

Algorithm 5.7 ($\widetilde{\mathsf{P}}$)**.**

Oracle: $\zeta \colon \{0,1\}^{\lambda} \mapsto \{0,1\}^{\lambda}$.

Input: Instance $\mathbb{x}$.

1. Run $\widetilde{\mathsf{P}'}^{\zeta}(\mathbb{x})$ to generate a proof π.
2. Emulate V on $(\mathbb{x}, \pi)$ to determine its list of oracle queries $(w_1, \ldots, w_{\mathsf{q_V}})$.
3. For $i = 1, \ldots, \mathsf{q_V}$:
   (a) Iterate in the salted soundness loop with query w_i for 2^{γ} times. Let $\widetilde{S}_i$ be the set of obtained answers.
   (b) If w_i was asked by $\widetilde{\mathsf{P}'}$ in Step 1, add the retrieved answer to $\widetilde{S}_i$.
4. If there exists $(y_1, \ldots, y_{\mathsf{q_V}}) \in \widetilde{S}_1 \times \ldots \times \widetilde{S}_{\mathsf{q_V}}$ that make V accept $(\mathbb{x}, \pi)$ with $(y_1, \ldots, y_{\mathsf{q_V}})$ as the answers to its oracle queries, output $(\pi, \sigma = [(w_1, y_1), \ldots, (w_{\mathsf{q_V}}, y_{\mathsf{q_V}})])$.

Recall that for $i \in \mathcal{J}$, the verifier V' sets S_i to be the output of a single call to the oracle, and for $i \notin \mathcal{J}$, it sets S_i to 2^γ random strings in $\{0,1\}^\lambda$. Hence, for every choice of ζ, there exists a coupling between the sets S_i sampled by V' to the sets $\widetilde{S}_i$ sampled by $\widetilde{\mathsf{P}}$ with $\widetilde{S}_i \supseteq S_i$ for every i. It follows that the probability that $\widetilde{\mathsf{P}}$ makes V accept $\mathbb{x}$ is at least as high as the probability that $\widetilde{\mathsf{P}'}$ makes P' accept $\mathbb{x}$, which by assumption is at least ε. This concludes the proof since by construction, $\widetilde{\mathsf{P}'}$ makes t' queries. □

5.2.3 Putting it Together

Proof of Lemma 5.2. Immediately follows by Claims 5.5 and 5.6. □

6 Lower Bound on the Length of ROM-SVCs

In this section, we present our lower bound on the length of ROM-SVCs (subvector commitments in the random oracle model).

We begin with a formal definition of ROM-SVCs. A ROM-SVC is a triplet of oracle-aided polynomial-time algorithms $\mathsf{CM} = (\mathsf{CM.Commit}, \mathsf{CM.Open}, \mathsf{CM.Verify})$. For a security parameter $\lambda \in \mathbb{N}$, a message space $\mathcal{M}$ the algorithms are given (query) access to the random oracle $\zeta \colon \{0,1\}^\lambda \to \{0,1\}^\lambda$:

- $\mathsf{CM.Commit}(\mathsf{m}_1, \ldots, \mathsf{m}_q)$. The algorithm $\mathsf{CM.Commit}$ gets as input a sequence of values $\mathsf{m}_1, \ldots, \mathsf{m}_q \in \mathcal{M}$, and outputs a commitment cm and auxiliary information aux.
- $\mathsf{CM.Open}(\mathsf{cm}, I, \mathsf{aux})$. The algorithm $\mathsf{CM.Open}$ gets as input a commitment cm, a subset $I \subseteq [q]$, and auxiliary information aux, and outputs a proof π.
- $\mathsf{CM.Verify}(\mathsf{cm}, I, \mathsf{M}_I, \pi)$. The algorithm $\mathsf{CM.Open}$ gets as input a commitment cm, a subset $I \subseteq [q]$, a vector M_I of length I and accepts only if π is a valid proof that $\mathsf{CM.Commit}$ was created with a sequence $\mathsf{m}_1, \ldots, \mathsf{m}_q$ such that $\mathsf{m}_I = \mathsf{M}_I$ (where m_I is the subset of $\mathsf{m}_1, \ldots, \mathsf{m}_q$ corresponding to the indices in I).

For correctness, we require that for any $\mathsf{M} = \mathsf{m}_1, \ldots, \mathsf{m}_q \in \mathcal{M}^q$, for any $I \subseteq [q]$ we have that

$$\Pr_{\substack{\zeta \leftarrow \mathcal{U}(\lambda) \\ (\mathsf{cm},\mathsf{aux}) \leftarrow \mathsf{CM.Commit}^\zeta(\mathsf{m}_1,\ldots,\mathsf{m}_q) \\ \pi \leftarrow \mathsf{CM.Open}^\zeta(\mathsf{cm}, I, \mathsf{aux})}} \left[\mathsf{CM.Verify}^\zeta(\mathsf{cm}, I, \mathsf{M}_I, \pi) = 1 \right] = 1 \ .$$

The main complexity measure for a vector commitment is the size of the commitment cm and its opening π (the subvector itself M_I is *not* included in the size). Thus, we say that the size of the commitment is bounded by s if for any $\mathsf{M} = \mathsf{m}_1, \ldots, \mathsf{m}_q \in \mathcal{M}^q$, $I \subseteq [q]$, and $\zeta \leftarrow \mathcal{U}(\lambda)$, we have that

- *Size*: $|\mathsf{cm}| + |\pi| \le \mathsf{s}$, where $(\mathsf{cm}, \mathsf{aux}) \leftarrow \mathsf{CM.Commit}^\zeta(\mathsf{m}_1, \ldots, \mathsf{m}_q)$, and $\pi \leftarrow \mathsf{CM.Open}^\zeta(\mathsf{cm}, I, \mathsf{aux})$.

For binding, we give a security definition with "salts", in the style of the salted soundness security for SNARGs (Section 3.5.1). Thus, we begin with a salted binding game, in which the cheating committer is allowed to re-sample elements of the random oracle (e.g., using different salts) and only when done giving out a commitment. For player A, we define

Game 6.1 ($\mathsf{SaltedBinding}_{\lambda,t}(\mathcal{A})$)**.**

Player: $\mathcal{A}$.

Paramters: $\lambda, t \in \mathbb{N}$.

1. Initialize keyed-map S of lists (each entry is initialized with the empty list).
2. Repeat the following t times (or until $\mathcal{A}$ decides to exit the loop):
   (a) $\mathcal{A}$ sends a query $x \in \{0,1\}^*$.
   (b) Sample $y \leftarrow \{0,1\}^\lambda$, and add it to the list $S[x]$.
3. $\mathcal{A}$ chooses a query-answer list $\sigma = [(x_1, y_1), \ldots, (x_n, y_n)]$, commitment cm and two openings $d_1 = (I_0, \mathsf{M}_0, \pi_0), d_2 = (I_1, \mathsf{M}_1, \pi_1)$.
4. If $y_i \notin S[x_i]$ for some $i \in [n]$, set $\sigma = \emptyset$.
5. Output 1 if and only if:
   (a) $\mathsf{M}_0[I] \neq \mathsf{M}_1[I]$;
   (b) $\mathsf{CM.Verify}^{\zeta_\sigma}(\mathsf{cm}, I_0, \mathsf{M}_0, \pi_0) = 1$; and
   (c) $\mathsf{CM.Verify}^{\zeta_\sigma}(\mathsf{cm}, I_1, \mathsf{M}_1, \pi_1) = 1$.

We say that $\mathsf{CM} = (\mathsf{CM.Commit}, \mathsf{CM.Open}, \mathsf{CM.Verify})$ has (t, ε)-salted binding if the following holds.

Definition 6.2 (Salted binding)**.** *For every security parameter $\lambda \in \mathbb{N}$, query bound $t \in \mathbb{N}$, and t-query algorithm A,*

$$\Pr[\mathsf{SaltedBinding}_{\lambda,t}(\mathsf{A}) = 1] \leq \varepsilon \ .$$

Given the above definition of vector commitment, with salted binding, and an amortized definition of opening size, we can state our lower bound on size of vector commitments in the ROM. In the theorem below, we require the scheme to have a *non-adaptive* verification algorithm, that is, one where all queries to the random oracle are performed in a single round.

Theorem 6.3. *Let CM be a (t, ε)-salted-sound, non-adaptive (deterministic) verification ROM-SVC for vectors of length n. Let $\mathsf{q_S}$ and $\mathsf{q_R}$ be the query complexity of the sender and receiver, respectively, let v be the running-time of the receiver. Let α denote the commitment length, and $\beta(\ell)$ denote the opening length for subsets of size ℓ.*

Assuming rETH, if

1. *$\varepsilon \leq 1/4$;*
2. *$t^{1/2}/\varepsilon \geq n$.*
3. *$\mathsf{q_V} \cdot \lambda \in o(n)$;*
4. *$\log t \cdot \log \frac{t}{\varepsilon} / \log \mathsf{q_P} \in o(n)$;*
5. *$\mathsf{q_V} \leq t^{1/10}$; and*
6. *$v \in 2^{o(n)}$,*

then $\alpha + \beta(\log \frac{t}{\varepsilon}) \in \Omega(\log t \cdot \log \frac{t}{\varepsilon} / \log n)$.

Proof. The proof from the Micali construction of SNARGs in the ROM [Mic00]. Micali's construction is merely a vector commitment to an appropriate PCP string and an opening to a subset indices that correspond to the location the PCP verifier reads. That is, the proof size in Micali's construction is precisely the vector commitment size for a specific subset I.

If the underlying PCP has soundness error ε_{PCP}, then the Micali construction has a soundness error that can be bounded by $t \cdot \varepsilon_{PCP} + 4 \cdot \frac{t^2}{2^\lambda}$ (see [BCS16]). This tells us how to set parameters for security: the Micali construction is (t, ε_{PCP})-salted sound secure, e.g., if each term is bounded by $\frac{\varepsilon}{2}$. This yields two requirements: (i) $\varepsilon_{PCP} \leq \frac{1}{2} \cdot \varepsilon / t$ (the PCP has small-enough soundness error); and (ii) $\lambda \geq \log(8\frac{t^2}{\varepsilon})$ (the random oracle has large-enough output size).

Using the PCP theorem, we can get a polynomial-size PCP with soundness error $\varepsilon_{PCP} \leq \frac{1}{2} \cdot \varepsilon / t$ where the verifier reads $q = O(\log t/\varepsilon)$ bits from the PCP. Thus, proof size corresponds to the vector commitment with an opening to a set of size $|I| = \Theta(\log t/\varepsilon)$ (or equivalently, opening $O(1)$ sets each of size exactly $\log t/\varepsilon$). Our lower bound on SNARGs, Theorem 5.1 tells us that the overall proof size (and thus also the total subvector commitment size) is at least $\Omega(\log t \cdot \log t/\varepsilon \cdot \log^{-1} n)$. □

Acknowledgments

We thank Gal Arnon and Alessandro Chiesa for many invaluable discussions. Iftach Haitner and Daniel Nukrai are supported by the Israel Science Foundation grant 666/19 and the Blavatnik Interdisciplinary Cyber Research Center at Tel-Aviv University. Eylon Yogev is supported by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office, and by the Alter Family Foundation.

References

[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 (cit. on p. 1).

[BBBPWM18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *Proceedings of the 39th IEEE Symposium on Security and Privacy*. S&P '18. 2018, pp. 315–334 (cit. on pp. 1, 5).

[BBCPGL18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits". In: *Proceedings of the 38th Annual International Cryptology Conference*. CRYPTO '18. 2018, pp. 669–699 (cit. on p. 5).

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Scalable Zero Knowledge with No Trusted Setup". In: *Proceedings of the 39th Annual International Cryptology Conference*. CRYPTO '19. 2019, pp. 733–764 (cit. on p. 1).

[BCCGP16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting". In: *Proceedings of the 35th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '16. 2016, pp. 327–357 (cit. on p. 5).

[BCGGMTV14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. “Zerocash: Decentralized Anonymous Payments from Bitcoin”. In: *Proceedings of the 2014 IEEE Symposium on Security and Privacy*. SP ’14. 2014, pp. 459–474 (cit. on p. 1).

[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 (cit. on p. 5).

[BCRSVW19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. “Aurora: Transparent Succinct Arguments for R1CS”. In: *Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT ’19. Full version available at `https://eprint.iacr.org/2018/828`. 2019, pp. 103–128 (cit. on p. 1).

[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 (cit. on pp. 2, 5, 12, 27).

[BCS21] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. “Sumcheck Arguments and Their Applications”. In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*. 2021, pp. 742–773 (cit. on p. 5).

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. “Transparent SNARKs from DARK Compilers”. In: *Proceedings of the 39th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT ’20. 2020, pp. 677–706 (cit. on pp. 1, 5).

[BGLR93] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. “Efficient Probabilistically Checkable Proofs and Applications to Approximations”. In: *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*. STOC ?93. 1993, pp. 294–304 (cit. on p. 5).

[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 (cit. on p. 5).

[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 (cit. on p. 5).

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. “A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge”. In: *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*. 2020, pp. 441–469 (cit. on p. 5).

[BM17] Boaz Barak and Mohammad Mahmoody-Ghidary. “Merkle’s Key Agreement Protocol is Optimal: An $\mathrm{O}(\mathrm{n}^2)$ Attack on Any Key Agreement from Random Oracles”. In: *J. Cryptol.* 30.3 (2017), pp. 699–734 (cit. on p. 1).

[BMG07] Boaz Barak and Mohammad Mahmoody-Ghidary. “Lower bounds on signatures from symmetric primitives”. In: *FOCS*. 2007, pp. 680–688 (cit. on p. 1).

[CCHLRR18] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum. *Fiat–Shamir From Simpler Assumptions*. Cryptology ePrint Archive, Report 2018/1004. 2018 (cit. on p. 1).

[CDGS18] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John Steinberger. "Random oracles and non-uniformity". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques.* 2018, pp. 227–258 (cit. on p. 4).

[CF13] Dario Catalano and Dario Fiore. "Vector Commitments and Their Applications". In: *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings.* 2013, pp. 55–72 (cit. on p. 3).

[CHMMVW20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Noah Vesely, and Nicholas Ward. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques.* EUROCRYPT '20. 2020 (cit. on pp. 1, 5).

[CMSZ21] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. "Post-Quantum Succinct Arguments". In: *IACR Cryptol. ePrint Arch.* (2021), p. 334 (cit. on p. 5).

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. "Fractal: Post-Quantum and Transparent Recursive Proofs from Holography". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques.* EUROCRYPT '20. 2020 (cit. on p. 1).

[CY20] Alessandro Chiesa and Eylon Yogev. "Barriers for Succinct Arguments in the Random Oracle Model". In: *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II.* 2020, pp. 47–76 (cit. on pp. 1–3, 12, 20, 21, 33, 34).

[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 (cit. on pp. 1, 2, 5, 12).

[CY21b] Alessandro Chiesa and Eylon Yogev. "Tight Security Bounds for Micali's SNARGs". In: *Theory of Cryptography - 19th International Conference, TCC.* 2021, pp. 401–434 (cit. on pp. 1, 2, 5, 12).

[DGK17] Yevgeniy Dodis, Siyao Guo, and Jonathan Katz. "Fixing cracks in the concrete: Random oracles with auxiliary input, revisited". In: *EUROCRYPT.* 2017, pp. 473–495 (cit. on p. 4).

[DHMTW14] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. "Exponential time complexity of the permanent and the Tutte polynomial". In: *ACM Transactions on Algorithms* 10.4 (2014), Art. 21, 32 (cit. on p. 11).

[EIRS01] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiri Sgall. "Communication complexity towards lower bounds on circuit depth". In: *Computational Complexity* 10.3 (2001), pp. 210–246 (cit. on p. 4).

[FS86] Amos Fiat and Adi Shamir. "How to prove yourself: practical solutions to identification and signature problems". In: *Proceedings of the 6th Annual International Cryptology Conference.* CRYPTO '86. 1986, pp. 186–194 (cit. on pp. 1, 2, 5).

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. "Bounds on the Efficiency of Generic Cryptographic Constructions". In: *SICOMP* 35.1 (2005), pp. 217–246 (cit. on p. 1).

[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 (cit. on p. 5).

[GH98] Oded Goldreich and Johan Håstad. “On the complexity of interactive proofs with bounded communication”. In: *Information Processing Letters* 67.4 (1998), pp. 205–214 (cit. on p. 34).

[GLLZ20] Siyao Guo, Qian Li, Qipeng Liu, and Jiapeng Zhang. “Unifying Presampling via Concentration Bounds.” In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 1589 (cit. on p. 4).

[GMNO18] Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù. “Lattice-Based zk-SNARKs from Square Span Programs”. In: *Proceedings of the 25th ACM Conference on Computer and Communications Security.* CCS ’18. 2018, pp. 556–573 (cit. on p. 5).

[GNS21] Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. “Rinocchio: SNARKs for Ring Arithmetic”. In: *IACR Cryptol. ePrint Arch.* (2021), p. 322 (cit. on p. 1).

[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.* ASIACRYPT ’10. 2010, pp. 321–340 (cit. on p. 5).

[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 (cit. on p. 1).

[GSV18] Aryeh Grinberg, Ronen Shaltiel, and Emanuele Viola. “Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs”. In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS).* 2018, pp. 956–966 (cit. on p. 4).

[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 (cit. on p. 5).

[HMORY19] Iftach Haitner, Noam Mazor, Rotem Oshman, Omer Reingold, and Amir Yehudayoff. “On the Communication Complexity of Key-Agreement Protocols”. In: *ITCS.* 2019, 40:1–40:16 (cit. on p. 1).

[IR89] Russell Impagliazzo and Steven Rudich. “Limits on the provable consequences of one-way permutations”. In: *STOC.* 1989, pp. 44–61 (cit. on p. 1).

[ISW21] Yuval Ishai, Hang Su, and David J. Wu. “Shorter and Faster Post-Quantum Designated-Verifier zkSNARKs from Lattices”. In: *IACR Cryptol. ePrint Arch.* (2021), p. 977 (cit. on p. 5).

[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 (cit. on p. 5).

[LM19] Russell W. F. Lai and Giulio Malavolta. “Subvector Commitments with Application to Succinct Arguments”. In: *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I.* 2019, pp. 530–560 (cit. on p. 3).

[LSTW21] Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. “Linear-time zero-knowledge SNARKs for R1CS”. In: *IACR Cryptol. ePrint Arch.* (2021), p. 30 (cit. on p. 1).

[MBKM19] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. *Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updateable Structured Reference Strings.* Cryptology ePrint Archive, Report 2019/099. 2019 (cit. on p. 5).

[Mer82] Ralph C. Merkle. “Secure Communications over Insecure Channels”. In: *SIMMONS: Secure Communications and Asymmetric Cryptosystems.* 1982 (cit. on p. 1).

[Mic00] Silvio Micali. "Computationally Sound Proofs". In: *SIAM Journal on Computing* 30.4 (2000). Preliminary version appeared in FOCS '94., pp. 1253–1298 (cit. on pp. 1, 2, 5, 12, 27).

[Nit19] Anca Nitulescu. "Lattice-Based Zero-Knowledge SNARGs for Arithmetic Circuits". In: *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings.* 2019, pp. 217–236 (cit. on p. 5).

[PGHR13] Brian Parno, Craig Gentry, Jon Howell, and Mariana Raykova. "Pinocchio: Nearly Practical Verifiable Computation". In: *Proceedings of the 34th IEEE Symposium on Security and Privacy.* Oakland '13. 2013, pp. 238–252 (cit. on p. 5).

[Raz98] Ran Raz. "A parallel repetition theorem". In: *SIAM Journal on Computing* 27.3 (1998), pp. 763–803 (cit. on p. 4).

[Set19] Srinath Setty. *Spartan: Efficient and general-purpose zkSNARKs without trusted setup.* Cryptology ePrint Archive, Report 2019/550. 2019 (cit. on p. 5).

[Sta18] libstark. *libstark: a C++ library for zkSTARK systems.* 2018. URL: `https://github.com/elibensasson/libSTARK` (cit. on p. 1).

[SV10] Ronen Shaltiel and Emanuele Viola. "Hardness amplification proofs require majority". In: *SIAM Journal on Computing* 39.7 (2010), pp. 3122–3154 (cit. on p. 4).

[Unr07] Dominique Unruh. "Random oracles and auxiliary input". In: *Annual International Cryptology Conference.* 2007, pp. 205–223 (cit. on p. 4).

[WTSTW18] Riad S. Wahby, Ioanna Tzialla, Abhi Shelat, Justin Thaler, and Michael Walfish. "Doubly-efficient zkSNARKs without trusted setup". In: *Proceedings of the 39th IEEE Symposium on Security and Privacy.* S&P '18. 2018, pp. 926–943 (cit. on p. 1).

[Zc14] Electric Coin Company. *Zcash Cryptocurrency.* `https://z.cash/`. 2014 (cit. on p. 1).

[ZGKPP17] Yupeng Zhang, Daniel Genkin, Jonathan Katz, Dimitrios Papadopoulos, and Charalampos Papamanthou. *A Zero-Knowledge Version of vSQL.* Cryptology ePrint Archive, Report 2017/1146. 2017 (cit. on p. 1).

A Salted Soundness Amplification

In this section we prove Lemma 3.8, restated below.

Lemma A.1 (Restatement of Lemma 3.8). *Let* ARG *be an ROM-SNARG for a language* $\mathcal{L}$ *with* (t, ε)*-salted-soundness for* $\varepsilon \leq 1/4$*. Then* ARG *has* $(t/k, 2\varepsilon/k)$*-salted-soundness for any* $k \in \mathbb{N}$*, .*

Proof. Assume towards a contradiction that there exists a t/k-query cheating prover $\widetilde{\mathsf{P}}$ that wins the salted soundness game with probability more than $\varepsilon' := 2\varepsilon/k$. Then, we construct a t-query cheating prover $\widetilde{\mathsf{P}}_k$ that will succeed with probability more than ε, contradicting the soundness condition.

Roughly speaking, $\widetilde{\mathsf{P}}_k$ emulates $\widetilde{\mathsf{P}}$ for k times, each time it will uses new salts, so that $\widetilde{\mathsf{P}}$ gets fresh and independent randomness (for the random oracle) between its runs. Thus, $\widetilde{\mathsf{P}}_k$ will succeed if any of the k iterations of $\widetilde{\mathsf{P}}$ where successful. More formally, $\widetilde{\mathsf{P}}_k$ acts as follows in the salted soundness game:

Algorithm A.2 ($\widetilde{\mathsf{P}}_k$)**.**

1. For all $i \in [k]$:

(a) Emulate the play of $\widetilde{\mathsf{P}}$ in the salted soundness game (as described in Section 3.5.1), as follows:

i. When $\widetilde{\mathsf{P}}$ choose a query $x \in \{0,1\}^*$, $\widetilde{\mathsf{P}}_k$ chooses the same query x.

ii. $\widetilde{\mathsf{P}}_k$ gets a response y which is given to $\widetilde{\mathsf{P}}$.

(b) The emulation end with $\widetilde{\mathsf{P}}$ outputting a proof π_i and a list σ_i.

(c) Emulate V on π_i with the random oracle ζ, while overriding queries with σ_i (i.e., if $(x, y) \in \sigma_i$ then query x will get response y, otherwise respond with $\zeta(x)$).

(d) If the verifier accepts, then output π_i and a list σ_i and halt.

Note that if $\widetilde{\mathsf{P}}_k$ outputs $\sigma = \sigma_i$ for some $i \in [k]$, we have all the queries in σ asked in the i^{th} emulation. Therefore, if $(x, y) \in \sigma$, this means that $y \in S[x]$, as required in the *salted soundness game*. Since we emulate the salted soundness game in each iteration, we know that for all $i \in [k]$ the probability that (π_i, σ_i) will lead V to accept is at least $\varepsilon' = 2\varepsilon/k$. As the simulations of $\widetilde{\mathsf{P}}$ are independent, the probability that all of the emulations will end up in V rejecting is at most

$$(1 - \varepsilon')^k = \left(1 - \frac{2\varepsilon}{k}\right)^k \leq e^{-2\varepsilon} \leq 1 - 2\varepsilon + 4\varepsilon^2 \leq 1 - \varepsilon \ .$$

This proves that $\widetilde{\mathsf{P}}_k$ wins with probability more than $1 - (1 - \varepsilon) = \varepsilon$, which is a contradiction to the salted soundness of ARG. □

B A Folklore Lower Bound on ROM-SNARG Length

We observe that one can derive a simple lower bound of $\Omega(\log \frac{t}{\varepsilon})$ for any non-trivial SNARGs with (t, ε)-soundness, where the verifier performs $\mathsf{q_V}$ to the random oracle. The terms non-trivial means that there exists an instance $\mathbb{x} \notin \mathcal{L}$ and at least one "wrong proof", i.e., a proof for a false statement that the verifier accepts.[11]

Theorem B.1 (Lower bound on ROM-SNARG length. Folklore)**.** *Let* $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ *be a non-trivial* s*-length ROM-SNARG, with* (t, ε)*-soundness. Let* $\mathsf{q_V}$ *be the query complexity of* V*. If* $\mathsf{q_V} \leq t^{1/2}$*, then* $\mathsf{s} \geq \frac{1}{2} \cdot \log \frac{t}{\varepsilon}$*.*

Proof sketch. Fix an instance $\mathbb{x} \notin \mathcal{L}$ for which a false proof exists. We first claim a lower bound of $\Omega(\log t)$. If the SNARG has a proof of size $1/2 \cdot \log t$, a cheating prover can enumerate all possible proofs (there are $t^{1/2}$ such proofs), and for each run the verifier to check if it accepts. This would take the cheating prover at most $t^{1/2} \cdot \mathsf{q_V} \leq t$ queries, where at the end it will find a false proof with probability 1 (as we assumed that such a proof exists).

Next, we show a lower bound of $1/2 \cdot \log(1/\varepsilon)$. If a SNARG has proof length at most $1/2 \cdot \log(1/\varepsilon)$ then, given a false statement, a cheating prover can simply guess a wrong proof. Since at least one such proof exists, then it will succeed with probability at least $\varepsilon^{1/2}$, contradicting the (t, ε) soundness guarantee (note that cheating prover in this case makes no queries at all).

Combining these two lower bounds together, we get our lower bound of $\frac{1}{2} \cdot \log \frac{t}{\varepsilon}$. □

[11] Since the existence of trivial short ROM-SNARG for 3SAT contradicts rETH, one can replace the non-triviality requirement in the following theorem with the rETH assumption.

C Proof of Lemma 5.3

In this section, we give details about the proof of Lemma 5.3, restated below, that follows from similar arguments to main proof appearing in [CY20].

Lemma C.1 (Lemma 5.3, restated). *Let* $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ *be a* (t, ε)*-sound ROM-SNARG for* n*-variable* 3SAT *with random oracle (input and output) length* λ*, argument length* s*, and let* $\mathsf{q_V}$ *and* $\mathsf{q_P}$ *denote* P*'s and* V*'s query complexity, respectively. Assume*

1. $\mathsf{s} + \lambda \cdot \mathsf{q_V} \in o(n)$*;*
2. $\mathsf{q_V} \leq 1/10 \cdot \log(1/\varepsilon) \cdot \log^{-1} \mathsf{q_P}$*;*
3. *completeness* $\geq \varepsilon^{1/2}$*;*
4. $\log(1/\varepsilon) \leq o(n^{1/2})$*; and*
5. V*'s running time* $2^{o(n)}$*,*

then $3\mathrm{SAT} \in \mathrm{BPTIME}[2^{o(n)}]$.

Proof. We transform the given SNARG to a laconic IP (i.e., one where there is small prover-to-verifier communication).

Construction C.2. Let $\mathsf{ARG} = (\mathsf{P}, \mathsf{V})$ be a non-interactive argument in the ROM. We construct a public-coin interactive oracle proof $\mathsf{IP} = (\mathsf{P}', \mathsf{V}')$, parametrized by a choice of security parameter $\lambda \in \mathbb{N}$. The IP prover P' takes as input an instance $\mathbb{x}$ and a witness $\mathbb{w}$, and will internally simulate the argument prover P on input $(\mathbb{x}, \mathbb{w})$, answering P's queries to the random oracle as described below. The IP verifier V' takes as input only the instance $\mathbb{x}$, and will simulate the argument verifier V on input $\mathbb{x}$, answering V's queries to the random oracle as described below.

The interactive phase of the IP protocol proceeds as follows:

- P guesses a subset $I \subseteq [\mathsf{q_P}]$ of size V.
- For round $j = 1, \ldots, \mathsf{q_P}$:
  1. P' simulates P to get its j-th query X_j.
  2. If $j \in I$ then:
     (a) P' sends X_j to the verifier.
     (b) The verifier responds with y_j.
  3. Otherwise, P' samples y_j at random.
- P' simulates P until it outputs the non-interactive argument π, which is sent to the verifier.
- The verifier, given π, simulates $\mathsf{V}(\mathbb{x}, \pi)$ while answering query w as follows:
  1. If w was sent to the verifier during step Item 2a then answer with the corresponding y given in Item 2b.
  2. Otherwise, sample y at random.

For the simplicity of the analysis, we assume that all of the SNARG verifier queries are a subset of the SNARG prover queries. (This can always be achieved by having the honest prover simulate the verifier at the end of its execution. This slightly increases the query complexity of the honest prover but does not effect on our results.)

We now argue completeness and soundness. Let α be the completeness of the SNARG, and let ε be the soundness error.

- *The completeness of* $(\mathsf{P}', \mathsf{V}')$ *is at least* $(1-\alpha) \cdot \binom{\mathsf{q_P}}{\mathsf{q_V}}^{-1}$.

  The IP prover P' simulates the SNARG prover P, where P' sends only the queries in I to the verifier and the rest are self simulated. If the IP prover P' guessed all queries of the verifier (i.e., we never get to Item 2), then the prover and verifier agree on all queries, and thus we get that the verifier will accept with probability at least α, where α is the completeness of the SNARG.

  The probability of the prover guessing I to include all the verifier queries is $\binom{\mathsf{q_P}}{\mathsf{q_V}}^{-1}$. Notice that these two events are independent, and therefore the probability of a correct prediction and that the verifier accepts is at least $(1-\alpha) \cdot \binom{\mathsf{q_P}}{\mathsf{q_V}}^{-1}$.

- *The soundness error of* $(\mathsf{P}', \mathsf{V}')$ *is at most* ε.

  Suppose that there exists a malicious IP prover $\widetilde{\mathsf{P}'}_{\mathsf{IP}IP}$ that convinces V' to accept with probability at least ε. We construct a malicious SNARG prover $\widetilde{\mathsf{P}}$ that convinces the SNARG verifier V to accept with the same probability. The SNARG prover $\widetilde{\mathsf{P}}$ runs $\widetilde{\mathsf{P}'}_{\mathsf{IP}IP}$ while replacing the (public-coin) IP verifier with the random oracle. If the IP verifier accepts, then the SNARG verifier accepts as well since the IP verifier makes its decision according to the SNARG verifier, with the same distribution of queries.

Observe that the number of rounds of the IP is at most $\mathsf{q_V}$, and its communication complexity is bounded by $\mathsf{s} + O(\mathsf{q_V} \cdot \lambda)$. To finish the proof, we plug in the above IP into the following lemma, which is proven in [CY20], as a refinement of [GH98].

Lemma C.3 (IP to algorithm)**.** *Suppose that a language $\mathcal{L}$ has a public-coin IP with completeness error α, soundness error β, round complexity k, prover-to-verifier communication c, and verifier running time $v(n)$. Then, for $d(n) := c(n) + \mathsf{k}(n) \cdot \log \frac{\mathsf{k}(n)}{1-\alpha(n)-\beta(n)}$ the language $\mathcal{L}$ is in*

$$\mathrm{BPTIME}\left[2^{O(d)} \cdot v(n) \cdot \mathrm{poly}(n)\right] \quad .$$

Notice that, since $v = 2^{o(n)}$, to conclude the proof, we must show that $d(n) = o(n)$, which will imply that $\mathcal{L}$ is in

$$\mathrm{BPTIME}\left[2^{O(d)} \cdot v(n) \cdot \mathrm{poly}(n)\right] = \mathrm{BPTIME}\left[2^{o(n)}\right] \quad .$$

Recall that $d(n) := c(n) + \mathsf{k}(n) \cdot \log \frac{\mathsf{k}(n)}{1-\alpha(n)-\beta(n)}$. To show that $d(n) = o(n)$ we make the following observations:

1. $c(n) = o(n)$: This is since we have argued that $c(n) = \mathsf{s} + O(\mathsf{q_V} \cdot \lambda) = o(n)$, by our assumptions in the lemma statement.
2. $1 - \alpha(n) - \beta(n) \geq 1/2 \cdot \varepsilon^{1/5}$: since the argument system has completeness at least $\varepsilon^{1/10}$, we get that the completeness is at least

$$1 - \alpha(n) \geq \varepsilon^{1/10} \cdot \binom{\mathsf{q_P}}{\mathsf{q_V}}^{-1} \geq \varepsilon^{1/10} \cdot 2^{-\mathsf{q_V} \cdot \log \mathsf{q_P}} \geq \varepsilon^{1/10} \cdot 2^{-1/10 \cdot \log(1/\varepsilon)} = \varepsilon^{1/5} \quad .$$

   Thus, we get that $1 - \alpha(n) - \beta(n) \geq \varepsilon^{1/5} - \varepsilon \geq 1/2 \cdot \varepsilon^{1/5}$.

3. $\mathsf{k}(n) \cdot \log \frac{\mathsf{k}(n)}{1-\alpha(n)-\beta(n)} = o(n)$: Given the previous item, we can bound:

$$
\begin{aligned}
& \mathsf{k}(n) \cdot \log \frac{\mathsf{k}(n)}{1-\alpha(n)-\beta(n)} \\
& \leq \mathsf{q}_{\mathsf{V}} \cdot \log \frac{\mathsf{q}_{\mathsf{V}}}{1/2 \cdot \varepsilon^{1/5}} \\
& \leq O(\log(1/\varepsilon) \cdot \log^{-1} \mathsf{q}_{\mathsf{P}} \cdot (\log 1/\varepsilon + \log \mathsf{q}_{\mathsf{P}})) \\
& \leq O(\log^2(1/\varepsilon)) \\
& \leq o(n) \ .
\end{aligned}
$$

4. $d(n) = o(n)$: this follows since $d(n) = c(n) + \mathsf{k}(n) \cdot \log \frac{\mathsf{k}(n)}{1-\alpha(n)-\beta(n)} \leq o(n) + o(n) \leq o(n)$.

This concludes the proof. □