

Lower Bound on SNARGs in the Random Oracle Model

Iftach Haitner¹[0000-0003-3167-3294], Daniel Nukrai¹ and Eylon Yogev²[0000-0001-8599-2472]

¹ Tel-Aviv University
iftachh@tauex.tau.ac.il
daniel.nukrai@cs.tau.ac.il

² Bar-Ilan University
eylon.yogev@biu.ac.il

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

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 .³

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 $\Omega(\log(t/\varepsilon))$ (which appears in the full version of the paper), 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 $O(\log(t/\varepsilon) \cdot \log n)$, as achieved in other security

³ We focus on the *bare* ROM— no computational assumptions are made beyond bounding the query complexity to the oracle.

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 $\tilde{\Omega}(\log(t/\varepsilon) \cdot \log t)$ (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.⁴
- **Non-adaptive verifier.** The oracle queries are asked by a non-adaptive (deterministic⁵) verifier. That is, the queries are a function of the proof and are *independent* of the answers to other queries.⁶
- **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.⁷

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.⁸

⁴ This follows since $P = NP$ yields trivial SNARGs for all NP.

⁵ If the verifier is “public-coin” then it can be made deterministic by extracting randomness from the random oracle. However, this makes the verifier *adaptive* and thus cannot be used for our lower bound.

⁶ 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.

⁷ 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.

⁸ 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]).

With these notions, we are ready to state our main result. (The precise statements of the following results are given in the main body of the paper, see Paper Organization for references.)

Theorem 1 (Conditional lower bound on ROM-SNARG length. Informal). *Let $\text{ARG} = (\text{P}, \text{V})$ be an s -length ROM-SNARG for n -variable 3SAT, with (t, ε) -salted-soundness, perfect completeness, and (deterministic) non-adaptive verifier. Let q_{P} and 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 $q_{\text{V}} \cdot \lambda \in o(n)$, and $\log^2(t/\varepsilon) \cdot \log^{-1} q_{\text{P}} \in o(n)$ then $s \geq c \cdot \log t \cdot \log \frac{t}{\varepsilon} \cdot \log^{-1} q_{\text{P}}$, for some universal constant $c > 0$.

We argue that the assumptions on the parameters regime in our theorem are reasonable and consider the most interesting settings (see Theorem 13 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 $\text{poly}(\lambda, \log n)$. Thus, our assumption that $q_{\text{V}} \cdot \lambda$, and $\log t \cdot \log \frac{t}{\varepsilon} / \log q_{\text{P}}$ are of order $o(n)$ is rather mild. The third requirement of $q_{\text{V}} \leq t^{1/10}$ 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 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 (Short ROM-SNARG \rightarrow low query ROM-SNARG. Informal). *Let $\text{ARG} = (\text{P}, \text{V})$ be a ROM-SNARG for a language \mathcal{L} with a deterministic non-adaptive verifier and (t, ε) -salted-soundness, perfect completeness, proof length s , and verifier query complexity q_{V} . Then there exists a verifier V' of query complexity $s / \log t$, running time $2^{q_{\text{V}} \cdot \log t}$ times that of V , such that (P, 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 instance 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 (P', 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 yields a lower bound on ROM-SVCs for an analog family of constructions: non-adaptor receiver and salted-binding (i.e., the sender can resample the oracle outputs).

Theorem 2 (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 q_S and 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 $q_R \cdot \lambda \in o(n)$, and $\log^2(t/\varepsilon) \cdot \log^{-1} q_S \in o(n)$, 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 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 $X = (X_1, \dots, X_m)$ 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.⁹

Theorem 3 (Hitting high-entropy distributions using product sets, informal). *Let $X = (X_1, \dots, X_m)$ 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*

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

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 $X_{[m] \setminus \mathcal{B}} | X_{\mathcal{B}} = x_{\mathcal{B}}$ by sampling a *tiny*, in relative terms, product set.

Note that Theorem 3 does *not* state that $X_{[m] \setminus \mathcal{B}} | X_{\mathcal{B}} = x_{\mathcal{B}}$ is close to the uniform distribution. Actually, it might be very far from that, e.g., for $X = (U_1, \dots, U_m) | \bigoplus U_i = 0^\lambda$ 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 $X_{[m] \setminus \mathcal{B}} | X_{\mathcal{B}} = x_{\mathcal{B}}$ being close to uniform. (And this example demonstrates why the “pre-sampling” approach and alike, c.f., [Unr07], do not seem to be relevant for proving bounds of the type stated in the theorem.) It is also worth mentioning that one cannot prove Theorem 3 using the simple observation that after fixing some bad indices, the projection of $X' \stackrel{\text{def}}{=} (X | X_{\mathcal{B}} = x_{\mathcal{B}})$ 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 $S \cap \text{Supp}(X'_{[m] \setminus \mathcal{B}})$ is well-concentrated around its mean.

In our application of Theorem 3, 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 verifier 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*

⁹ This is a generalization since for uniformly distributed X it holds that $H(X | W) \geq \lambda m - \log 1/\Pr[W]$.

checkable proof (PCP) and a commitment scheme into ROM-SNARG. Using the best known 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 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 Yegorov [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 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. Lattice based constructions, which are plausibly post-quantum, either achieve *private-verifiability* [BISW17; BISW18; GMNO18; ISW21; Nit19], or are public-verifiable, 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 (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 3 (hitting high-entropy events using product sets) is proved in Section 4. Theorem 1 (lower bound on the length of ROM-SNARGs) and its accompanied Lemma 1 are proved in Section 5, and Theorem 2 (lower bound on the length of ROM subvector commitments) is proved in the full version of the paper.

2 Techniques

In this section, we give a high-level overview of our proof for Lemma 1, 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 $\text{ARG} = (\text{P}, \text{V})$ for a language \mathcal{L} with (t, ε) -salted-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 \mathbf{x} and the witness \mathbf{w} .) For this simple scenario, the construction of a k -query V' is rather straightforward:

Algorithm 4 (Low-query verifier V' . Warmup).

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

Input: Instance \mathbf{x} and a proof π .

Operation:

1. Emulate $\text{V}(\mathbf{x}, \pi)$ till it produces a list of oracle queries $(w_1, \dots, w_{q_{\text{V}}})$. (Recall that V is non-adaptive.)
2. Sample a random k -size subset $\mathcal{J} \subseteq [q_{\text{V}}]$.
3. For $i = 1, \dots, q_{\text{V}}$:
 - If $i \in \mathcal{J}$, set $y_i = \zeta(w_i)$.
 - Otherwise, sample $y_i \leftarrow \{0, 1\}^\lambda$.
4. Accept if V accepts on the emulation with $(y_1, \dots, y_{q_{\text{V}}})$ 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 $\text{ARG}' = (\text{P}, \text{V}')$.

- Completeness. If the set \mathcal{J} happens to contain all important queries, then the given proof π , the instance \mathbf{x} , and the witness \mathbf{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 $1/|\binom{[q_{\text{V}}]}{k}|$ —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 $(t - \mathbf{q}_V)$ -query cheating prover \tilde{P}' that makes V' accept $\mathbf{x} \notin \mathcal{L}$ with probability ε . Consider the following t -query cheating prover \tilde{P} for violating the salted-soundness of ARG.¹⁰

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

By definition, if \tilde{P} outputs a proof π then V accepts π on the programmed oracle. In addition, the probability that \tilde{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: \tilde{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

¹⁰ 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.

details. It follows that the answers for the non-important queries can be emulated by the verifier (without querying the oracle). Equipped with this understanding, the low query V' is defined as follows:

Algorithm 5 (Low-query verifier V').

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

Parameters: $\gamma < \lambda$.

Input: Instance \mathbb{x} and a proof π .

Operation:

1. Emulate $V(\mathbb{x}, \pi)$ till it produces a list of oracle queries (w_1, \dots, w_{q_V}) . (Recall that V is non-adaptive.)
2. Sample $k' \in [k]$ at random and sample, a random $k' = \lceil s/\gamma \rceil$ -size subset $\mathcal{J} \subseteq [q_V]$.
3. For $i = 1, \dots, q_V$:
 - If $i \in \mathcal{J}$, set $S_i = \{\zeta(w_i)\}$.
 - Otherwise, let S_i be a 2^γ -size *random* subset of $\{0, 1\}^\lambda$.
4. Accept if there exists $(y_1, \dots, y_{q_V}) \in S_1 \times \dots \times S_{q_V}$ that make V accepts on the emulation, with (y_1, \dots, y_{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 candidate 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) $2^{q_V \cdot \gamma}$, and the following claim states the completeness and soundness of $\text{ARG}' = (\text{P}, V')$:

Claim (Informal). ARG' has $(\lambda \cdot q_P \cdot k \cdot \binom{q_V}{\lceil s/\gamma \rceil})^{-1}$ -completeness and $(t - q_V \cdot 2^\gamma, \varepsilon)$ -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 $Y = (Y_1, \dots, Y_{q_P})$ 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

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

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

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

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

$$\Pr \left[(S_1 \times \dots \times S_{q_P - |\mathcal{B}|}) \cap \text{Supp}(Y'_{[q_P] \setminus \mathcal{B}}) \neq \emptyset \right] \in \Omega(1/\lambda \cdot q_P) \quad (3)$$

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

Assume for simplicity that V and P make exactly the same queries. By Equation (3), if the random set \mathcal{J} (sampled by V') is exactly $\mathcal{B} = \mathcal{B}(\pi)$, then with probability $\Omega(1/\lambda \cdot q_P)$ 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 $y = (y_1, \dots, y_{q_P}) \in \text{Supp}(Y|_{\Pi=\pi})$. 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 $\Omega(1/\lambda \cdot q_P)$ times $\Pr[\mathcal{J} = \mathcal{B}] \geq 1/k \cdot 1/\binom{q_P}{s^\gamma}$.

Remark 1 (Improved 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 (2) (i.e., an error of $(\lambda \cdot q_P)^{-1}$), 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 $(t - q_V \cdot 2^\gamma)$ -query cheating prover \tilde{P}' that makes V' accepts $\mathbb{x} \notin \mathcal{L}$ with probability ε , and consider the following t -query cheating prover \tilde{P} for violating the salted-soundness of ARG.

Algorithm 6 (\tilde{P}).

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

Input: Instance \mathbb{x} .

1. Run $\tilde{P}'^\zeta(\mathbb{x})$ to generate a proof π .
2. Emulate V on (\mathbb{x}, π) to determine its list of oracle queries (w_1, \dots, w_{q_V}) .
3. For $i = 1, \dots, q_V$:

- (a) Query ζ on w_i for 2^γ times. Let S_i be the set of answers.
 - (b) If w_i was asked by \tilde{P}' in Step 1, add the retrieved answer to S_i .
4. If there exists $(y_1, \dots, y_{q_V}) \in S_1 \times \dots \times S_{q_V}$ that make V accept (x, π) with (y_1, \dots, y_{q_V}) as the answers to its oracle queries, program $\zeta(w_i) = y_i$ for each $i \in [q_V]$.
 5. Output π .

The cheating probability of \tilde{P} is at least as high as that of \tilde{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 $[n] = \{1, \dots, n\}$. Given a vector $v \in \Sigma^n$, let v_i denote its i th 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 $\text{Supp}(X)$, is defined as $\{x: \Pr[X = x] > 0\}$. 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 $\text{BPTIME}[T]$ 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 7. $\log(1-x) \leq -x$ for $x \in [0, 1]$, and $\log(1-x) \geq -2x$, for any $x \in [0, 1/2]$.

Theorem 8 (Paley–Zygmund inequality). *For any finite non-negative random variable X it holds that $\Pr[X > 0] \geq \text{E}[X]^2 / \text{E}[X^2]$.*

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 $x \in \text{Supp}(X)$, the *sample-entropy* of x with respect to X is the quantity

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

letting $H_X(x) = \infty$ for $x \notin \text{Supp}(X)$, 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* $H_\infty(X)$ as follows:

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

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

It can be shown that $H_\infty(X) \leq H(X) \leq H_0(X)$ with each inequality being an equality if and only if X is flat (uniform on its support). Thus, saying that $H_\infty(X) \geq k$ 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 $(x, y) \in \text{Supp}(X, Y)$, 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

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

The following fact gives a bound on the amount of entropy that is reduced when conditioning on an event for uniformly distributed random variables.

Fact 9. *Let X be a random variable uniform over a set \mathcal{S} and let W be an event. Then $H(X | W) \geq \log(|\mathcal{S}|) - \log 1/\Pr[W]$.*

3.3 Randomized Exponential Time Hypothesis

Definition 1 (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 $\zeta: \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$. Given an oracle algorithm A and an oracle $\zeta \in \mathcal{U}(\lambda)$, $\text{queries}(A, \zeta)$ is the set of oracle queries that A^ζ makes. We say that A is *t-query* if $|\text{queries}(A, \zeta)| \leq t$ 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 $x \in \text{queries}(A, \zeta)$ 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 $\text{ARG} = (\text{P}, \text{V})$ 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 $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$:

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

- **Soundness.**¹¹ For every $\lambda \in \mathbb{N}$, t -query $\tilde{\text{P}}$ and $\mathbb{x} \notin \mathcal{L}(\mathcal{R})$:

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

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:* $s := |\pi|$.
- *times:* the prover P runs in time pt ; the verifier V runs in time vt .
- *queries:* the prover P is a q_{P} -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.¹² This notion is formalized via the following *salted soundness game* defined as follows:

Game 10 ($\text{SaltedSoundness}_{\text{V}, \lambda, t}(\text{A}, \mathbb{x})$).

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

¹¹ This notion, where \mathbb{x} is set before the oracle, is sometimes referred 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).

¹² 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.

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

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 $\sigma = [(x_1, y_1), \dots, (x_n, y_n)]$.
4. Abort if $y_i \notin S[x_i]$ for some $i \in [n]$.
5. Output $V^{\zeta^\sigma}(\mathbf{x}, \pi)$.

Definition 2 (Salted soundness). We say that ROM-SNARG $(\tilde{\mathcal{P}}, V)$ has (t, ε) -salted-soundness for a language \mathcal{L} , if for any λ , $\mathbf{x} \notin \mathcal{L}$ and $\tilde{\mathcal{P}}$ it holds that $\Pr \left[\text{SaltedSoundness}_{V, \lambda, t}(\tilde{\mathcal{P}}, \mathbf{x}) = 1 \right] \leq \varepsilon(|\mathbf{x}|, \lambda, t)$.

Remark 2 (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]).

Amplification. It turns out that salted soundness can be easily amplified (at the expense of the query complexity). The proof of Lemma 2 is proved in the full version of the paper.

Lemma 2. 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 3. 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 11 (Hitting high-entropy distributions using product sets, restatement of Theorem 3). Let $\gamma \leq \lambda \in \mathbb{N}$, and let $X = (X_1, \dots, X_m)$ 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

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

Remark 3 (Tightness of Theorem 11). The size of \mathcal{B} in Theorem 11 is tight up to a constant: Let $m, \lambda, \gamma \in \mathbb{N}$ be as in Theorem 11, let $X = (X_1, \dots, X_m)$ be uniform over $(\{0, 1\}^\lambda)^m$ and let W be the event that $X_1 = \dots = X_t = 0^\lambda$, 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

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

which is negligible for sufficiently small γ , e.g., $\gamma = \lambda/2$. This matches, up to a constant, Theorem 11, 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 11. We start with describing the high-level approach of the proof. We need to prove that with high probability over $x \leftarrow X$, there exists a small (i.e., with size at most $8\ell/\gamma + 4$) subset $\mathcal{B} \subseteq [m]$ such that

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

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 j th element of S_i , according to this order, by $S_i[j]$. For $y = (y_1, \dots, y_{\widehat{m}}) \in [2^\gamma]^{\widehat{m}}$, 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 $S^y \in \text{Supp}(\widehat{X})$, and let $Z \stackrel{\text{def}}{=} \sum_{y \in [2^\gamma]^{\widehat{m}}} Z^y$. That is, Z^y is event that the y th element of S is in $\text{Supp}(\widehat{X})$. 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,

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

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

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

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 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 $(S^y)_{\mathcal{I}} \in \text{Supp}(\widehat{X}_{\mathcal{I}})$, 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

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

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 $\mathbb{E}[Z^2]$ is small.

Moving to the formal proof, Theorem 11 is an immediate corollary of the following two lemmata: Lemma 3 states that with high probability over x , there exists a small set \mathcal{B} for which Equation (6) holds, and Lemma 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 3 (High-entropy events have an almost full-entropy large projection). *Let $\gamma \leq \lambda \in \mathbb{N}$, and let $X = (X_1, \dots, X_m)$ 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 (Hitting almost full-entropy events using product sets). *Let $\gamma \leq \lambda \in \mathbb{N}$, let $X = (X_1, \dots, X_m)$ be a random variable over $(\{0, 1\}^\lambda)^m$. Assume $\gamma \geq 2 \cdot \lceil \log m \rceil + 3$, and that for every $x \in \text{Supp}(X)$ and $\mathcal{I} \subseteq [m]$, it holds that $H_{X_{\mathcal{I}}}(x_{\mathcal{I}}) \geq (\lambda - \gamma/2) \cdot |\mathcal{I}|$. Then*

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

We prove Lemmas 3 and 4 in Sections 4.1 and 4.2, respectively, but first use them for proving Theorem 11.

Proof of Theorem 11: Let $t \stackrel{\text{def}}{=} 8\ell/\gamma + 4$, and let

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

Since, by assumption, $\gamma/2 \geq 2 \lceil \log m \rceil + 2$, Lemma 3 yields that

$$\Pr[X \in \mathcal{T}] \geq 1/2 . \quad (7)$$

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), and 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

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

Combining Equations (7) and (8), concludes the proof. \square

**4.1 High-Entropy Distributions Have an (Almost) Uniform Large Projection,
Proving Lemma 3**

Proof of Lemma 3. Let m, λ, γ and X be as in Lemma 3. For $x \in \text{Supp}(X)$, let \mathcal{B}^x be the (lex. first) *maximal*¹³ subset of $[m]$ with

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

Since Equation (9) holds for the empty set, \mathcal{B}^x is always defined. We prove Lemma 3 using the following two claims, proven below.

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

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

By Section 4.1, for every $x \in \text{Supp}(X)$ and $\mathcal{I} \subseteq [m] \setminus \mathcal{B}^x$, it holds that

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

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 $p_{\mathcal{I}} = \Pr[f_{\mathcal{I}}(X) = \perp]$. Compute

$$H(X_{\mathcal{B}^x} | \mathcal{B}^x) = \mathbb{E}_{\mathcal{B}^x \leftarrow \mathcal{B}^x} [H(X_{\mathcal{B}^x} | \mathcal{B}^x = \mathcal{B})] \quad (11)$$

$$\begin{aligned} &= \mathbb{E}_{\mathcal{B}^x \leftarrow \mathcal{B}^x} [H(f_{\mathcal{B}^x}(X) | \mathcal{B}^x = \mathcal{B})] \leq \sum_{\mathcal{I}} \mathbb{E}_{\mathcal{B}^x \leftarrow \mathcal{B}^x} [H(f_{\mathcal{I}}(X) | \mathcal{B}^x = \mathcal{B})] \\ &= \sum_{\mathcal{I}} H(f_{\mathcal{I}}(X) | \mathcal{B}^x) \leq \sum_{\mathcal{I}} H(f_{\mathcal{I}}(X)) \\ &= \sum_{\mathcal{I}} \left(\sum_{x: \mathcal{B}^x = \mathcal{I}} \Pr[X = x] \cdot H_{X_{\mathcal{I}}}(x_{\mathcal{I}}) \right) + p_{\mathcal{I}} \cdot \log(1/p_{\mathcal{I}}) \\ &\leq \sum_{\mathcal{I}} \Pr[\mathcal{B}^x = \mathcal{I}] \cdot (\lambda - \gamma) \cdot |\mathcal{I}| + p_{\mathcal{I}} \cdot \log(1/p_{\mathcal{I}}) \quad (12) \\ &= (\lambda - \gamma) \mathbb{E}[|\mathcal{B}^x|] + \sum_{\mathcal{I}} p_{\mathcal{I}} \cdot \log(1/p_{\mathcal{I}}) \\ &\leq (\lambda - \gamma) \mathbb{E}[|\mathcal{B}^x|] + 1 + \sum_{\mathcal{I}, p_{\mathcal{I}} \geq 1/2} -p_{\mathcal{I}} \cdot \log(p_{\mathcal{I}}) \\ &\leq (\lambda - \gamma) \mathbb{E}[|\mathcal{B}^x|] + 1 + \sum_{\mathcal{I}, p_{\mathcal{I}} \geq 1/2} p_{\mathcal{I}} \cdot 2(1 - p_{\mathcal{I}}) \quad (13) \\ &= (\lambda - \gamma) \mathbb{E}[|\mathcal{B}^x|] + 1 + 2 \cdot \sum_{\mathcal{I}, p_{\mathcal{I}} \geq 1/2} p_{\mathcal{I}} \cdot \Pr[\mathcal{B}^x = \mathcal{I}] \\ &\leq (\lambda - \gamma) \mathbb{E}[|\mathcal{B}^x|] + 3. \end{aligned}$$

¹³ Maximal means relative to inclusion—there is no \mathcal{I} strictly containing \mathcal{B}^x with $H_{X_{\mathcal{I}}}(x_{\mathcal{I}}) \leq (\lambda - \gamma) \cdot |\mathcal{I}|$.

Inequality 12 holds by the definition of \mathcal{B}^x , and Inequality 13 holds since $\log(1-x) \geq -2x$ for $x \in [0, 1/2]$.

On the other hand since, by assumption, $H(X) \geq \lambda \cdot m - \ell$, Section 4.1 yields that

$$H(X_{\mathcal{B}^x} | \mathcal{B}^x) \geq (\lambda - \lceil \log m \rceil) \cdot \mathbb{E}[\lceil \mathcal{B}^x \rceil] - \ell - \lceil \log m \rceil \quad (14)$$

Combining Equations (11) and (14), we conclude that $\mathbb{E}[\lceil \mathcal{B}^x \rceil] \leq \frac{\ell + \lceil \log m \rceil + 3}{\gamma - \lceil \log m \rceil} \leq 2\ell/\gamma + 2$, where the 2nd inequality follows from the fact that $\gamma \geq 2 \cdot \lceil \log m \rceil + 3$. The proof follows by Markov inequality. \square

Proving Section 4.1.

Proof of Section 4.1. Let $\mathcal{B} = \mathcal{B}^x$. Since for every disjoint sets $\mathcal{A}, \mathcal{C} \subseteq [m]$ and $x \in \text{Supp}(X)$

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

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

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

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

Proving Section 4.1.

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

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

we conclude that

$$H(X | I) \geq \lambda m - \ell - (\mathbb{E}_{x \leftarrow X}[|I|] + 1) \lceil \log m \rceil \quad (15)$$

Therefore,

$$H(X | I) = H(X_I, X_{[m] \setminus I} | I) \leq H(X_I | I) + H(X_{[m] \setminus I} | I) \quad (16)$$

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

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

\square

4.2 Hitting almost Full-Entropy Distributions using Product Set, Proving Lemma 4

We start by proving the following variant of Lemma 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.

Lemma 5 (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*

$$|\{x \in \mathcal{T} : x_{\mathcal{I}} = a\}| \leq |\mathcal{T}| \cdot 2^{(\gamma/2 - \lambda)|\mathcal{I}|} / \delta , \quad (17)$$

then

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

Proof. Let $S = (S_1, \dots, S_m)$ be as in the lemma statement, i.e., uniformly distributed over $(\mathcal{P}_{2^\gamma}(\{0, 1\}^\lambda))^m$. We assume, without loss of generality, that the elements of each S_i are chosen in a uniform order and denote the j th element of S_i , according to this order, by $S_i[j]$. For $y = (y_1, \dots, y_m) \in [2^\gamma]^m$, let $S^y \in \{0, 1\}^{\lambda \times m}$ be the random variable defined by $(S^y)_i \stackrel{\text{def}}{=} S_i[y_i]$. Let Z^y be the indicator for the event $S^y \in \mathcal{T}$, and let $Z \stackrel{\text{def}}{=} \sum_{y \in [2^\gamma]^m} Z^y$. By the Paley–Zygmund inequality, Theorem 8, it holds that

$$\Pr_{S \leftarrow (\mathcal{P}_{2^\gamma}(\{0, 1\}^\lambda))^m} [S \cap \mathcal{T} \neq \emptyset] = \Pr[Z > 0] \geq \mathbb{E}[Z]^2 / \mathbb{E}[Z^2] . \quad (18)$$

Thus, we prove Lemma 5 by properly bounding $\mathbb{E}[Z]$ and $\mathbb{E}[Z^2]$. Let $\rho \stackrel{\text{def}}{=} \frac{|\mathcal{T}|}{2^{m\lambda}}$. Since we associate a random order with the elements of each S_i , for every $y \in [2^\gamma]^m$ it holds that $\mathbb{E}[Z^y] = \rho$. Hence,

$$\mathbb{E}[Z] = \sum_{y \in [2^\gamma]^m} \mathbb{E}[Z^y] = 2^{\gamma m} \rho . \quad (19)$$

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

Claim 12. *For every $y, y' \in [2^\gamma]^m$ it holds that $\Pr[Z^y \wedge Z^{y'}] \leq 2^{\gamma \cdot |\mathcal{K}_{y, y'}|/2} \cdot \rho^2 / \delta$.*

For $\mathcal{K} \subseteq [m]$, let $\mathcal{A}_{\mathcal{K}} \stackrel{\text{def}}{=} \{(y, y') \in [2^\gamma]^m : \mathcal{K}_{y, y'} = \mathcal{K}\}$. Using Claim 12, we deduce that

$$\begin{aligned}
\mathbb{E}[Z^2] &= \sum_{y, y' \in [2^\gamma]^m} \Pr[Z^y \wedge Z^{y'}] \tag{20} \\
&= \sum_{\mathcal{K} \subseteq [m]} \sum_{y, y' \in \mathcal{A}_{\mathcal{K}}} \Pr[Z^y \wedge Z^{y'}] \\
&\leq \sum_{\mathcal{K} \subseteq [m]} \sum_{y, y' \in \mathcal{A}_{\mathcal{K}}} 2^{\gamma|\mathcal{K}|/2} \cdot \rho^2 / \delta \\
&\leq \frac{\rho^2}{\delta} \cdot \sum_{k=0}^m \sum_{\mathcal{K} \subseteq [m], |\mathcal{K}|=k} 2^{\gamma k} \cdot (2^{2\gamma})^{m-k} \cdot 2^{\gamma k/2} \\
&= \frac{\rho^2}{\delta} \cdot 2^{2\gamma m} \cdot \sum_{k=0}^m \binom{m}{k} \cdot 2^{-\gamma k/2} \\
&\leq \frac{\rho^2}{\delta} \cdot 2^{2\gamma m} \cdot \sum_{k=0}^m 2^{-k \cdot (\gamma/2 - \log m)} \leq 2 \cdot \frac{\rho^2}{\delta} \cdot 2^{2\gamma m}.
\end{aligned}$$

The first inequality holds by Claim 12, and the last one by holds since, by assumption, $\gamma \geq 2 \cdot \lceil \log m \rceil + 2$. Combining Equations (18) to (20), prove the lemma by deducing that

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

□

Proving Claim 12.

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

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

The second inequality holds by the assumption of the lemma (Equation (17)).

□

4.2.1 Proving Lemma 4

Proof of Lemma 4. Define

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

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

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

and let $\mathcal{T}^0 \stackrel{\text{def}}{=} \{x \in \mathcal{T} : H_X(x) \geq 2\lambda m\}$. 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

$$\Pr[X \in \mathcal{T}^i] \geq \frac{1 - 2^{-\lambda \cdot m}}{2\lambda m} \geq \frac{1}{4\lambda m} \quad (21)$$

The second inequality hold since, by assumption, $\lambda \geq \gamma \geq 2$. In the rest of the proof we use Lemma 5 to prove that $\Pr_{S \leftarrow \mathcal{P}_{2\gamma}(\{0,1\}^\lambda)} [S \cap \mathcal{T}^i \neq \emptyset]$. Let $X^i = X|_{X \in \mathcal{T}^i}$, and for $\mathcal{I} \subseteq [m]$ and $a \in \text{Supp}(X_{\mathcal{I}}^i)$, let $\mathcal{T}_{\mathcal{I},a}^i \stackrel{\text{def}}{=} \{x \in \mathcal{T}^i : x_{\mathcal{I}} = a\}$. Since X^i is almost flat, for every $a \in \text{Supp}(X_{\mathcal{I}}^i)$ and $x \in \mathcal{T}_{\mathcal{I},a}^i$:

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

Similarly,

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

Combing the above two inequalities, we get that

$$\Pr[X_{\mathcal{I}}^i = a] \geq \frac{1/2 \cdot |\mathcal{T}_{\mathcal{I},a}^i| \cdot \Pr[X^i = x]}{2 \cdot |\mathcal{T}^i| \cdot \Pr[X^i = x]} = \frac{|\mathcal{T}_{\mathcal{I},a}^i|}{4 \cdot |\mathcal{T}^i|} \quad (22)$$

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

$$\Pr[X_{\mathcal{I}} = x_{\mathcal{I}}] \leq 2^{(\gamma/2 - \lambda)|\mathcal{I}|} \quad (23)$$

Therefore, for every $a \in \text{Supp}(X_{\mathcal{I}}^i)$:

$$\frac{|\mathcal{T}_{\mathcal{I},a}^i|}{|\mathcal{T}^i|} \leq 4 \cdot \Pr[X_{\mathcal{I}}^i = a] \leq 4 \cdot \frac{\Pr[X_{\mathcal{I}} = a]}{\Pr[X \in \mathcal{T}^i]} \leq 16\lambda m \cdot 2^{(\gamma/2-\lambda)|\mathcal{I}|} \quad (24)$$

The first inequality holds by Equation (22), and the third by Equation (23). Applying Lemma 5 for the set \mathcal{T}^i with parameter $\delta = 1/16\lambda m$, yields that

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

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

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 1 for the formal definition of rETH, and Section 3.5 for that of salted-soundness ROM-SNARGs).

Theorem 13 (Conditional lower bound on ROM-SNARGs length). *Let $\text{ARG} = (\mathsf{P}, \mathsf{V})$ be an s -length ROM-SNARG for n -variable 3SAT, with (t, ε) -salted-soundness, perfect completeness, and deterministic non-adaptive verifier. Let q_{P} and 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. $q_{\mathsf{V}} \cdot \lambda \in o(n)$, $q_{\mathsf{V}} + \lambda \leq t^{1/10}$;
3. $\log^2(t/\varepsilon) \cdot \log^{-1} q_{\mathsf{P}} \in o(n)$; and
4. $v \in 2^{o(n)}$,

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

Theorem 13 is proved using the following two lemmata. Lemma 6 states that the verifier query complexity of a short ROM-SNARG can be significantly reduced, and Lemma 7, taken from [CY20], states that the existence of a low verifier query complexity ROM-SNARGs contradicts rETH.

Lemma 6 (Short ROM-SNARGs \rightarrow Low Query ROM-SNARGs). *Let $\text{ARG} = (\mathsf{P}, \mathsf{V})$ be as in Theorem 13, then for any $\gamma \in \mathbb{N}$, there exists a verifier V' such that $\text{ARG}' \stackrel{\text{def}}{=} (\mathsf{P}, \mathsf{V}')$ is a ROM-SNARG for \mathcal{L} with the following properties:*

1. completeness $(\lambda \cdot q_{\mathsf{P}} \cdot q_{\mathsf{V}}^{20 \cdot \lceil s/\gamma \rceil})^{-1}$;
2. $(t - q_{\mathsf{V}} \cdot 2^\gamma, \varepsilon)$ -soundness;
3. verifier query complexity $20 \cdot \lceil s/\gamma \rceil$; and
4. verifier running time $O(2^{q_{\mathsf{V}} \cdot \log t} \cdot v)$.

Furthermore, the transformation from \mathcal{V} to \mathcal{V}' is efficient (in the description length of \mathcal{V}).

In words, Lemma 6 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 6 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]. A formal proof is given in the full version of the paper.

Lemma 7 (Follows from [CY20]). *Let $\text{ARG} = (\mathcal{P}, \mathcal{V})$ be a (t, ε) -sound ROM-SNARG for n -variable 3SAT with random oracle (input and output) length λ , argument length s , and let $q_{\mathcal{V}}$ and $q_{\mathcal{P}}$ denote \mathcal{P} 's and \mathcal{V} 's query complexity, respectively. Assume*

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

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

Note that Lemma 7 does not require \mathcal{V} to be deterministic or non adaptive.

5.1 Proof of Theorem 13

Proof of Theorem 13. Suppose we are given a SNARG ARG for 3SAT that satisfies the conditions of the theorem, and assume without loss of generality that $q_{\mathcal{P}} \leq t^{1/10}$. (Otherwise, for $q_{\mathcal{P}} > t^{1/10}$, the lower bound we need to prove can be written as $s \geq 2^{-15} \cdot \log \frac{t}{\varepsilon}$, which follows by the folklore lower bound¹⁴). Assume towards contradiction that $s \leq 2^{-15} \cdot \log t \cdot \log \frac{t}{\varepsilon} / \log q_{\mathcal{P}}$. Theorem 13 is proved via the following steps:

1. Apply Lemma 2 with parameter $k = t^{0.5}$ which yields a scheme ARG that has (t', ε') -salted-soundness, where $t' = t^{1/2}$, and $\varepsilon' = 2\varepsilon/t^{1/2}$.
2. Apply Lemma 6 with $\gamma = 1/10 \cdot \log t$, to get a ROM-SNARG ARG' for 3SAT with the following parameters:
 - (a) completeness $(\lambda \cdot q_{\mathcal{P}} \cdot q_{\mathcal{V}}^{20 \cdot \lceil s/\gamma \rceil})^{-1}$;
 - (b) $(t' - q_{\mathcal{V}} \cdot 2^\gamma, \varepsilon')$ -soundness.
 - (c) verifier query complexity $q_{\mathcal{V}'} = 20 \cdot \lceil s/\gamma \rceil$; and
 - (d) verifier running time $v' = O(2^{q_{\mathcal{V}} \cdot \log t} \cdot v)$.

¹⁴ The proof of the folklore lower bound appears in the full version of the paper.

3. Apply Lemma 7 on ARG' to contradict rETH . For this, we need to verify that all five conditions of the lemma apply. Indeed,

- (i) $\mathfrak{s} + \lambda \cdot \mathfrak{q}_{\mathcal{V}'} \in o(n)$: First, observe that $\mathfrak{s} \leq 2^{-15} \cdot \log t \cdot \log \frac{t}{\varepsilon} / \log \mathfrak{q}_{\mathcal{P}} \in o(n)$. Then, since $\lambda \cdot \mathfrak{q}_{\mathcal{V}} \in o(n)$, we get that $\lambda \cdot \mathfrak{q}_{\mathcal{V}'} = O(\lambda \cdot \mathfrak{s} / \gamma) = O(\log t \cdot \mathfrak{s} / \log t) = o(n)$. Together, we have that $\mathfrak{s} + \lambda \cdot \mathfrak{q}_{\mathcal{V}'} \leq o(n) + o(n) = o(n)$.
- (ii) $\mathfrak{q}_{\mathcal{V}'} \leq 1/4 \cdot \log(1/\varepsilon') \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}}$: the query complexity of the verifier of ARG' is

$$\begin{aligned} \mathfrak{q}_{\mathcal{V}'} &\leq 20 \cdot \lceil \mathfrak{s} / \gamma \rceil \leq 20 \cdot \left\lceil \frac{2^{-15} \cdot \log t \cdot \log \frac{t}{\varepsilon} / \log \mathfrak{q}_{\mathcal{P}}}{1/10 \cdot \log t} \right\rceil \leq 1/8 \cdot \log \frac{t}{\varepsilon} \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}} \\ &\leq 1/4 \cdot \log \frac{t^{1/2}}{2\varepsilon} \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}} = 1/4 \cdot \log \frac{1}{\varepsilon'} \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}} . \end{aligned}$$

- (iii) completeness $\geq \varepsilon'^{2/3}$: Observe that $20 \lceil \mathfrak{s} / \gamma \rceil \leq 2^{-10} \cdot \log(t/\varepsilon) \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}}$. Thus, the completeness of our scheme satisfies:

$$\begin{aligned} \left(\lambda \cdot \mathfrak{q}_{\mathcal{P}} \cdot \mathfrak{q}_{\mathcal{V}'}^{20 \lceil \mathfrak{s} / \gamma \rceil} \right)^{-1} &\geq \left(t^{1/10} \cdot t^{1/10} \cdot \mathfrak{q}_{\mathcal{V}}^{2^{-10} \cdot \log(t/\varepsilon) \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}}} \right)^{-1} \\ &\geq 2^{-2/10 \log t - 2^{-10} \cdot \log(t/\varepsilon)} \geq 2^{-2/10 \log t - 2^{-9} \cdot \log(t^{1/2}/2\varepsilon)} \\ &\geq 2^{-3/10 \cdot \log(t^{1/2}/2\varepsilon)} = 2^{3/10 \cdot \log(\varepsilon')} \geq \varepsilon'^{2/3} . \end{aligned}$$

- (iv) $\log^2(1/\varepsilon') \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}} \leq o(n)$: By the definition of ε' and the conditions of the theorem we get that $\log^2(1/\varepsilon') \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}} = O(\log^2(t/\varepsilon) \cdot \log^{-1} \mathfrak{q}_{\mathcal{P}}) = o(n)$.
- (v) \mathcal{V}' 's running time $2^{o(n)}$: The verifier running time of the scheme is $O(2^{\mathfrak{q}_{\mathcal{V}'} \cdot \log t} \cdot v)$. Since $\mathfrak{q}_{\mathcal{V}} \cdot \log t = o(n)$ and $v = 2^{o(n)}$, its total running time is $2^{o(n)}$.

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

5.2 Short ROM-SNARGs to Low Query ROM-SNARGs, Proving Lemma 6

In this section, we prove Lemma 6 (see Section 2 for a high-level overview of the proof). Let $\text{ARG} = (\mathcal{P}, \mathcal{V})$ be ROM-SNARG with (t, ε) -salted soundness, random oracle of length λ , a non-adaptive deterministic verifier, prover query complexity $\mathfrak{q}_{\mathcal{P}}$, and verifier query complexity $\mathfrak{q}_{\mathcal{V}}$. The low query verifier \mathcal{V}' is defined as follows:

Algorithm 14 (Low-query verifier \mathcal{V}').

Oracle: $\zeta: \{0, 1\}^\lambda \mapsto \{0, 1\}^\lambda$.
Parameter: $\gamma \leq \lambda$. Let $k = 20 \lceil \mathfrak{s} / \gamma \rceil$.
Input: Instance \mathfrak{x} and proof π .
Operation:

1. Emulate V on (x, π) to get a list of queries $w = (w_1, \dots, w_{q_V})$.
2. Sample $k' \in [k]$, uniformly st random and uniformly sample a k' -size subset $\mathcal{J} \subseteq [q_V]$.
3. For each $i \in [q_V]$:
 - If $i \in \mathcal{J}$, set $S_i = \{\zeta(w_i)\}$.
 - Otherwise, let S_i be a 2^γ -size *random* subset of $\{0, 1\}^\lambda$.
4. Accept if there exists $(y_1, \dots, y_{q_V}) \in S_1 \times \dots \times S_{q_V}$ that make V accepts given (y_1, \dots, y_{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 $\text{ARG}' = (P, V')$ 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. ARG' has completeness $\geq (\lambda \cdot q_P \cdot q_V^{20 \cdot \lceil s/\gamma \rceil})^{-1}$.

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 $(P^\zeta(w), V'^\zeta)(x)$: denote P 's queries by $X = (X_1, \dots, X_{q_P})$, define $Z = (Z_1, \dots, Z_{q_P})$ 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, \dots, X_{q_V} .

The length of Π is s , thus a standard argument yields that $H(\Pi) \leq H_0(\Pi) \leq s$. Since each Z_i is a bit string of length λ (recall that λ is the output length of ζ), it holds that $H(Z | \Pi) \geq H(Z) - H(\Pi) \geq \lambda \cdot q_P - s$.

Since (by definition) $H(Z | \Pi) = \mathbb{E}_{\pi \leftarrow \Pi} [H(Z | \Pi = \pi)]$, with probability at least $1/2$ over $\pi \leftarrow \Pi$, it holds that $H(Z | \Pi = \pi) \geq \lambda \cdot q_P - 2 \cdot s$. Fix any such proof π , and let $Y = (Y_1, \dots, Y_{q_P}) = Z |_{\Pi=\pi}$. For $\ell = 2 \cdot s$, it holds that $H(Y) \geq \lambda \cdot q_P - \ell$. Applying Theorem 11 on Y yields that with probability $1/2$ over $y \leftarrow Y$ there exists a subset $\mathcal{B} \subseteq [q_P]$ with $|\mathcal{B}| \leq \lceil 8\ell/\gamma \rceil + 4$ such that:

$$\Pr_{S \leftarrow (\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda))^{q_P - |\mathcal{B}|}} [S \cap \text{Supp}(Y |_{Y_{\mathcal{B}}=y_{\mathcal{B}}}) \neq \emptyset] \geq \frac{1}{32 \cdot \lambda \cdot q_P}. \quad (25)$$

An immediate corollary of Equation (25) is that with probability at least $1/2$ over the choice of $y \leftarrow Y$, the following process outputs 1 with probability $\frac{1}{32 \cdot \lambda \cdot q_P}$:

1. For each $i \in [q_V]$:
 - 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 $(S_1 \times \dots \times S_{q_V}) \cap \text{Supp}((Y |_{Y_{\mathcal{B}}=y_{\mathcal{B}}})_{[q_V]}) \neq \emptyset$.

The perfect completeness of the argument scheme ARG yields that for any $\pi \in \text{Supp}(\Pi)$, it holds that $V(\mathbf{x}, \pi)$ accepts on any value of $z \in \text{Supp}((Y = Z |_{\Pi=\pi})_{[q_V]})$ given as oracle answers. Thus, it accepts any value of $z \in \text{Supp}((Y |_{Y_{\mathcal{B}}=y_{\mathcal{B}}})_{[q_V]})$ for any $y \in \text{Supp}(Y)$.

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

$$\begin{aligned} \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{32 \cdot \lambda \cdot q_P} \cdot \frac{1}{k} \cdot \frac{1}{\binom{q_V}{k}} &\geq \frac{1}{128 \cdot \lambda \cdot q_P} \cdot \frac{1}{k} \cdot \frac{(k/e)^k}{q_V^k} \\ &\geq \frac{1}{e \cdot 128 \cdot \lambda \cdot q_P} \frac{(k/e)^{k-1}}{q_V^k} \geq \frac{1}{e \cdot 128 \cdot \lambda \cdot q_P} \frac{(20/e)^{19}}{q_V^k} \geq \frac{1}{\lambda \cdot q_P \cdot q_V^k} . \end{aligned}$$

□

5.2.2 Soundness

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

Claim. ARG' has $(t - q_V \cdot 2^\gamma, \varepsilon)$ -soundness.

Proof. Let \tilde{P}' be a $t' := t - q_V \cdot 2^\gamma$ -query cheating prover such that $\Pr \left[\langle \tilde{P}', V'(\mathbf{x}) \rangle = 1 \right] > \varepsilon$, for some $\mathbf{x} \notin \mathcal{L}$. We show how to use \tilde{P}' to construct the following t -query cheating prover \tilde{P} such that $\Pr \left[\text{SaltedSoundness}_{V, \lambda, t'}(\tilde{P}, \mathbf{x}) = 1 \right] > \varepsilon$, violating the assumed salted-soundness of (P, V) .

We assume without loss of generality that \tilde{P}' is deterministic. Indeed, since \tilde{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 15 (\tilde{P}).

Oracle: $\zeta: \{0, 1\}^\lambda \mapsto \{0, 1\}^\lambda$. Input: Instance \mathbf{x} .

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

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 \tilde{S}_i sampled by \tilde{P} with $\tilde{S}_i \supseteq S_i$ for every i . It follows that the probability that \tilde{P} makes V accept x is at least as high as the probability that \tilde{P}' makes P' accept x , which by assumption is at least ε . This concludes the proof since by construction, \tilde{P}' makes t' queries. \square

5.2.3 Putting it Together

Proof of Lemma 6. Immediately follows by Sections 5.2.1 and 5.2.2. \square

References

- [AHIV17] S. Ames, C. Hazay, Y. Ishai, and M. Venkatasubramanian. “Ligero: Lightweight Sublinear Arguments Without a Trusted Setup”. In: CCS '17.
- [BBBPWM18] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. “Bulletproofs: Short Proofs for Confidential Transactions and More”. In: S&P '18.
- [BBCPGL18] C. Baum, J. Bootle, A. Cerulli, R. d. Pino, J. Groth, and V. Lyubashevsky. “Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits”. In: CRYPTO '18.
- [BBHR19] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. “Scalable Zero Knowledge with No Trusted Setup”. In: CRYPTO '19.
- [BCCGP16] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. “Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting”. In: EUROCRYPT '16.
- [BCGGMTV14] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. “Zerocash: Decentralized Anonymous Payments from Bitcoin”. In: SP '14.
- [BCIOP13] N. Bitansky, A. Chiesa, Y. Ishai, R. Ostrovsky, and O. Paneth. “Succinct Non-Interactive Arguments via Linear Interactive Proofs”. In: TCC '13.
- [BCRSVW19] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward. “Aurora: Transparent Succinct Arguments for R1CS”. In: EUROCRYPT '19.
- [BCS16] E. Ben-Sasson, A. Chiesa, and N. Spooner. “Interactive Oracle Proofs”. In: TCC '16-B.
- [BCS21] J. Bootle, A. Chiesa, and K. Sotiraki. “Sumcheck Arguments and Their Applications”. In:
- [BFS20] B. Bünz, B. Fisch, and A. Szepieniec. “Transparent SNARKs from DARK Compilers”. In: EUROCRYPT '20.
- [BGLR93] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. “Efficient Probabilistically Checkable Proofs and Applications to Approximations”. In: STOC '93.
- [BISW17] D. Boneh, Y. Ishai, A. Sahai, and D. J. Wu. “Lattice-Based SNARGs and Their Application to More Efficient Obfuscation”. In: EUROCRYPT '17.

- [BISW18] D. Boneh, Y. Ishai, A. Sahai, and D. J. Wu. “Quasi-Optimal SNARGs via Linear Multi-Prover Interactive Proofs”. In: EUROCRYPT ’18.
- [BLNS20] J. Bootle, V. Lyubashevsky, N. K. Nguyen, and G. Seiler. “A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge”. In:
- [BM17] B. Barak and M. Mahmoody-Ghidary. “Merkle’s Key Agreement Protocol is Optimal: An $O(n^2)$ Attack on Any Key Agreement from Random Oracles”. In: *J. Cryptol.* (2017).
- [BMG07] B. Barak and M. Mahmoody-Ghidary. “Lower bounds on signatures from symmetric primitives”. In:
- [CCHLRR18] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, and R. D. Rothblum. “Fiat–Shamir From Simpler Assumptions”. Cryptology ePrint Archive, Report 2018/1004.
- [CDGS18] S. Coretti, Y. Dodis, S. Guo, and J. Steinberger. “Random oracles and non-uniformity”. In:
- [CF13] D. Catalano and D. Fiore. “Vector Commitments and Their Applications”. In:
- [CHMMVW20] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, and N. Ward. “Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS”. In: EUROCRYPT ’20.
- [CMSZ21] A. Chiesa, F. Ma, N. Spooner, and M. Zhandry. “Post-Quantum Succinct Arguments”. In: *IACR Cryptol. ePrint Arch.* (2021).
- [COS20] A. Chiesa, D. Ojha, and N. Spooner. “Fractal: Post-Quantum and Transparent Recursive Proofs from Holography”. In: EUROCRYPT ’20.
- [CY20] A. Chiesa and E. Yogev. “Barriers for Succinct Arguments in the Random Oracle Model”. In:
- [CY21a] A. Chiesa and E. Yogev. “Subquadratic SNARGs in the Random Oracle Model”. In: CRYPTO ’21.
- [CY21b] A. Chiesa and E. Yogev. “Tight Security Bounds for Micali’s SNARGs”. In:
- [DGK17] Y. Dodis, S. Guo, and J. Katz. “Fixing cracks in the concrete: Random oracles with auxiliary input, revisited”. In:
- [DHMTW14] H. Dell, T. Husfeldt, D. Marx, N. Taslaman, and M. Wahlén. “Exponential time complexity of the permanent and the Tutte polynomial”. In: *ACM Transactions on Algorithms* (2014).
- [EIRS01] J. Edmonds, R. Impagliazzo, S. Rudich, and J. Sgall. “Communication complexity towards lower bounds on circuit depth”. In: *Computational Complexity* (2001).
- [FS86] A. Fiat and A. Shamir. “How to prove yourself: practical solutions to identification and signature problems”. In: CRYPTO ’86.
- [GGKT05] R. Gennaro, Y. Gertner, J. Katz, and L. Trevisan. “Bounds on the Efficiency of Generic Cryptographic Constructions”. In: *SICOMP* (2005).
- [GGPR13] R. Gennaro, C. Gentry, B. Parno, and M. Raykova. “Quadratic Span Programs and Succinct NIZKs without PCPs”. In: EUROCRYPT ’13.
- [GLLZ20] S. Guo, Q. Li, Q. Liu, and J. Zhang. “Unifying Presampling via Concentration Bounds.” In: *IACR Cryptol. ePrint Arch.* (2020).
- [GMNO18] R. Gennaro, M. Minelli, A. Nitulescu, and M. Orrù. “Lattice-Based zk-SNARKs from Square Span Programs”. In: CCS ’18.

- [GNS21] C. Ganesh, A. Nitulescu, and E. Soria-Vazquez. “Rinocchio: SNARKs for Ring Arithmetic”. In: *IACR Cryptol. ePrint Arch.* (2021).
- [Gro10] J. Groth. “Short Pairing-Based Non-interactive Zero-Knowledge Arguments”. In: ASIACRYPT ’10.
- [Gro16] J. Groth. “On the Size of Pairing-Based Non-interactive Arguments”. In: EUROCRYPT ’16.
- [GSV18] A. Grinberg, R. Shaltiel, and E. Viola. “Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs”. In:
- [GW11] C. Gentry and D. Wichs. “Separating Succinct Non-Interactive Arguments From All Falsifiable Assumptions”. In: STOC ’11.
- [HMORY19] I. Haitner, N. Mazon, R. Oshman, O. Reingold, and A. Yehudayoff. “On the Communication Complexity of Key-Agreement Protocols”. In:
- [IR89] R. Impagliazzo and S. Rudich. “Limits on the provable consequences of one-way permutations”. In:
- [ISW21] Y. Ishai, H. Su, and D. J. Wu. “Shorter and Faster Post-Quantum Designated-Verifier zkSNARKs from Lattices”. In:
- [Kil92] J. Kilian. “A note on efficient zero-knowledge proofs and arguments”. In: STOC ’92.
- [LM19] R. W. F. Lai and G. Malavolta. “Subvector Commitments with Application to Succinct Arguments”. In:
- [LSTW21] J. Lee, S. T. V. Setty, J. Thaler, and R. S. Wahby. “Linear-time zero-knowledge SNARKs for R1CS”. In: *IACR Cryptol. ePrint Arch.* (2021).
- [MBKM19] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn. “Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings”. In:
- [Mer82] R. C. Merkle. “Secure Communications over Insecure Channels”. In:
- [Mic00] S. Micali. “Computationally Sound Proofs”. In: *SIAM Journal on Computing* (2000). Preliminary version appeared in FOCS ’94.
- [Nit19] A. Nitulescu. “Lattice-Based Zero-Knowledge SNARKs for Arithmetic Circuits”. In:
- [PGHR13] B. Parno, C. Gentry, J. Howell, and M. Raykova. “Pinocchio: Nearly Practical Verifiable Computation”. In: Oakland ’13.
- [Raz98] R. Raz. “A parallel repetition theorem”. In: *SIAM Journal on Computing* (1998).
- [Set19] S. Setty. “Spartan: Efficient and general-purpose zkSNARKs without trusted setup”. Cryptology ePrint Archive, Report 2019/550.
- [Sta18] libstark. *libstark: a C++ library for zkSTARK systems*. 2018. URL: <https://github.com/elibensasson/libSTARK>.
- [SV10] R. Shaltiel and E. Viola. “Hardness amplification proofs require majority”. In: *SIAM Journal on Computing* (2010).
- [Unr07] D. Unruh. “Random oracles and auxiliary input”. In:
- [WTSTW18] R. S. Wahby, I. Tzialla, A. Shelat, J. Thaler, and M. Walfish. “Doubly-efficient zkSNARKs without trusted setup”. In:
- [Zc14] Electric Coin Company. “Zcash Cryptocurrency”. <https://z.cash/>.
- [ZGKPP17] Y. Zhang, D. Genkin, J. Katz, D. Papadopoulos, and C. Papamanthou. “A Zero-Knowledge Version of vSQL”. Cryptology ePrint Archive, Report 2017/1146.