Approximate Lower Bound Arguments

Pyrros Chaidos[1,4], Aggelos Kiayias[2,4], Leonid Reyzin[3⋆], and Anatoliy Zinovyev[3]

[1] National & Kapodistrian University of Athens
[2] University of Edinburgh
[3] Boston University
[4] IOG

Abstract. Suppose a prover, in possession of a large body of valuable evidence, wants to quickly convince a verifier by presenting only a small portion of the evidence.

We define an Approximate Lower Bound Argument, or ALBA, which allows the prover to do just that: to succinctly prove knowledge of a large number of elements satisfying a predicate (or, more generally, elements of a sufficient total weight when a predicate is generalized to a weight function). The argument is approximate because there is a small gap between what the prover actually knows and what the verifier is convinced the prover knows. This gap enables very efficient schemes.

We present noninteractive constructions of ALBA in the random oracle and uniform reference string models and show that our proof sizes are nearly optimal. We also show how our constructions can be made particularly communication-efficient when the evidence is distributed among multiple provers, which is of practical importance when ALBA is applied to a decentralized setting.

We demonstrate two very different applications of ALBAs: for large-scale decentralized signatures and for proving universal composability of succinct proofs.

1 Introduction

Suppose a prover is in possession of a large body of valuable evidence that is individually verifiable. The evidence is so voluminous that presenting and verifying all of it is very expensive. Instead, the prover wants to convince a verifier by presenting only a small portion of the evidence.

More formally, let R be a predicate. We explore succinct arguments of knowledge where a prover knows a set S_{p} of values that satisfy R such that $|S_{\mathrm{p}}| \geq n_{\mathrm{p}}$ and wants to convince the verifier that $|S_{\mathrm{p}}| > n_{\mathrm{f}}$ where n_{f} is a somewhat smaller value than n_{p}. Given $n_{\mathrm{f}} < n_{\mathrm{p}}$, the verifier obtains a lower bound approximation to the actual cardinality of S_{p} and hence we call this primitive an *Approximate Lower Bound Argument* or ALBA.

⋆ Work done while visiting the Blockchain Technology Lab at the University of Edinburgh.

This problem has a long history. In 1983, in order to prove that that BPP $\subseteq$ $\mathrm{RP}^{\mathrm{NP}}$, Sipser and Gács [Sip83, Section V, Corollary to Theorem 6] showed a simple two-round interactive protocol for proving a lower bound on the size of the set S of accepting random strings. Their construction is based on hash collisions: the verifier chooses some number of universal hash functions $h_1, \ldots, h_m$ [CW79] and the prover shows s, s' such that $s \neq s'$ and $h_i(s) = h_i(s')$ for some $i \in \{1, \ldots, m\}$. If S is small (of size less than n_{p}), then such hash collisions are very unlikely to exist, and if S is big (of size greater than n_{f}), then they must exist by the pigeonhole principle. In 1986, Goldwasser and Sipser [GS86, Section 4.1] used a slightly different approach, based on the existence of inverses rather than collisions, for proving that public coins suffice for interactive proofs (cf. Appendix A). To the best of our knowledge, the term "approximate lower bound" in the context of proof systems appears first in Babai's work [Bab85, Section 5.2].

The pertinent efficiency consideration in designing ALBAs is primarily the length of the interaction between prover and verifier, and also the provers' and verifier's computational complexity. As it may be expected, the complexity metrics depend on the "gap" $n_{\mathrm{p}}/n_{\mathrm{f}}$, something that gives rise to the natural question of what is the smallest possible dependency between proof size and this gap. Putting this aspect at the forefront, it turns out that the classical techniques for constructing ALBAs mentioned above are quite suboptimal performance wise. While this does not affect the classical applications of ALBAs (such as in proving that any IP language can be decided by an Arthur-Merlin protocol, where the gap can be a large constant and the prover has exponential time), as we will see it does become a pressing concern in modern applications of ALBAs.

1.1 Our Setting

The prover and verifier have access to a binary string predicate R and the prover needs to show some elements of S_{p} to the verifier so he is convinced that the prover possesses more than n_{f} elements that pass R. The goal is to find some property that is unlikely to hold for small sets S_{f}, likely to hold for large sets S_{p}, and can be shown with just a few elements.

Generalization to Weighted Sets. We generalize a predicate R that determines validity of set elements, and consider instead a function *weight* that assigns value to data: it takes a set element and outputs a nonnegative integer indicating the value of the element. In that context we wish to explore succinct arguments of knowledge that convince a verifier that the prover knows a set S that satisfies a lower bound $\sum_{s \in S} \mathit{weight}(s) > n_{\mathrm{f}}$. When *weight* is $\{0, 1\}$-valued, we are in the setting of a predicate, and we call this case "unweighted."

We emphasize that R or *weight* are used in a black-box way in our protocols. Thus, our protocols can be used in settings when these functions do not have a known specification — for example, they may be evaluated by human judges who weigh evidence or via some complex MPC protocol that uses secret inputs.

Setup and Interaction Models. Our main focus is on building ALBA protocols that are succinct Non-Interactive Random-Oracle Proofs of Knowledge or

NIROPK (see Section 2 for the definition). If the prover is successful in convincing the verifier, then the knowledge extractor can obtain a set of size n_{f} by simply observing the random oracle queries; in other words, the protocol is straight-line extractable in the nonprogrammable random oracle model.

We also show simple modifications of our protocols that replace random oracles with pseudorandom functions (PRFs). By simply publishing the PRF seed as a shared random string, we obtain a non-interactive proof of knowledge in the Uniform Random String (URS) model, in which extractor works by reprogramming the URS. Alternatively, we can obtain a two-round public coin proof of knowledge by having the verifier send the PRF seed (we would then use rewinding for extraction). Protocols in these two models, however, require that the predicate R is fixed in advance; i.e., before the URS is published or the verifier sends the first message. Our NIROPK protocols, as presented, also possess this requirement, but could be modified to gain adaptive security, albeit downgrading soundness from information-theoretical to computational.

Decentralized Setting. The set S_{p} may be distributed among many parties. For instance, in a blockchain setting it could be the case that multiple contributing peers hold signatures on a block of transactions and they wish to collectively advance a protocol which approves that block. To capture such settings, we introduce decentralized ALBAs: in such a scheme, the provers diffuse messages via a peer to peer network, and an aggregator (who may be one of the provers themselves) collects the messages and produces the proof. Note that not all provers may decide to transmit a message. In addition to the complexity considerations of regular ALBAs, in the decentralized setting we also wish to minimize the total communication complexity in the prover interaction phase as well as the computational complexity of the aggregator.

1.2 Our Results

Our goal is to design protocols that give the prover a short, carefully chosen, sequence of elements from S_{p}. We show how to do this with near optimal efficiency.

Let λ be the parameter that controls soundness and completeness: the honest prover (who possesses a set of weight n_{p}) will fail with probability $2^{-\lambda}$ and the dishonest prover (who possesses a set of weight at most n_{f}) will succeed with, say, also probability $2^{-\lambda}$. Let u be the length of the sequence the prover sends.

The unweighted case. We first show an unweighted ALBA in which the prover sends only

$$u = \frac{\lambda + \log \lambda}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}} \tag{1}$$

elements of S_{p}. Moreover, we show that this number is essentially tight, by proving that at least

$$u = \frac{\lambda}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}}$$

elements of S_p are necessary. (Note that all formulas in this section omit small additive constants for simplicity; the exact formulas are given in subsequent sections.)

Such a protocol is relatively easy to build in the random oracle model if one disregards the running time of the prover: just ask the prover to brute force a sequence of u elements of S_p on which the random oracle gives a sufficiently rare output. Calibrate the probability ϵ of this output so that $\epsilon \cdot n_\mathrm{f}^u < 2^{-\lambda}$ for soundness, but $(1-\epsilon)^{n_\mathrm{p}^u} < 2^{-\lambda}$ for completeness. A bit of algebra shows that $u = \frac{\lambda + \log \lambda}{\log \frac{n_\mathrm{p}}{n_\mathrm{f}}}$ suffices to satisfy both soundness and completeness constraints, so the proof is short.[5] However, in this scheme, the prover has to do an exhaustive search of $1/\epsilon$ sequences of length u, and thus the running time is exponential.

It follows that the main technical challenge is in finding a scheme that maintains the short proof while allowing the prover to find one quickly. In other words, the prover needs to be able to find a sequence of u elements with some special rare property (that is likely to occur among n_p elements but not among n_f elements), without looking through all sequences. We do so in Section 3 by demonstrating the *Telescope* construction.

The core idea in the Telescope construction is to find a sequence of values that itself and also all its prefixes satisfy a suitable condition determined by a hash function (and modeled as a random oracle). This prefix invariant property enables the prover to sieve through the possible sequences efficiently expanding gradually the candidate sequence as in an extending Telescope. We augment this basic technique further via parallel self composition to match the proof length of the exhaustive search scheme. The resulting prover time (as measured in the number of random oracle queries) is dropped from exponential to $O(n_\mathrm{p} \cdot \lambda^2)$. We then show how to drop further the prover complexity to $O(n_\mathrm{p} + \lambda^2)$ by prehashing the head of the Telescope and expressing the prefix invariant property as a hash collision. We also provide a lower bound argument establishing that the constructions we present are essentially optimal in terms of proof size.

Weights and Decentralized Provers. In the case of weighted sets where each weight is an integer, the straightforward way to design a weighted scheme is to give each set element a multiplicity equal to its weight and apply the algorithms we described above. However, this approach is inefficient since this multiplicity is exponential in the weight function's outputs. A way to solve this problem is to select (with the help of the random oracle) a reasonably-sized subset of the resulting multiset by sampling, for each weighted element, a binomial distribution in accordance with its weight. Given this precomputation, we can proceed then with the Telescope construction as above and with only a logarithmic penalty due to the weights.

Turning our attention to the decentralized setting we present two constructions. In the first one, each party performs a private random-oracle-based coin

[5] Let $\epsilon = 2^{-\lambda} n_\mathrm{f}^{-u}$ to satisfy soundness. Then $(1-\epsilon)^{n_\mathrm{p}^u} < \exp(-2^{-\lambda} n_\mathrm{f}^{-u})^{n_\mathrm{p}^u} < \exp(2^{-\lambda}(n_\mathrm{p}/n_\mathrm{f})^u)$ is needed for completeness, so it suffices to have $\exp(2^{-\lambda}(n_\mathrm{p}/n_\mathrm{f})^u) < 2^{-\lambda}$, i.e., $2^{-\lambda}(n_\mathrm{p}/n_\mathrm{f})^u > \lambda/\log 2$, i.e. $(n_\mathrm{p}/n_\mathrm{f})^u > 2^{\lambda + \log\lambda - \log\log e}$. Taking logarithm to the base $n_\mathrm{p}/n_\mathrm{f}$ gives the desired result.

flip to decide whether to share her value. The aggregator produces a proof by concatenating a number of the resulting values equal to a set threshold. In the second construction, we combine the above idea with the Telescope construction letting the aggregator do a bit more work; this results in essentially optimal proof size with total communication complexity $O(\lambda^3)$ or proof size an additive term $\sqrt{\lambda}$ larger than optimal and total communication complexity $O(\lambda^2)$.

1.3 Applications

Beyond the classical applications of ALBAs in complexity theory described earlier [CW79,Sip83,Bab85,GS86], there are further applications of the primitive in cryptography.

Weighted Multisignatures and Compact Certificates. In a multisignature scheme, a signature is accepted if sufficiently many parties have signed the message (depending on the flavor, the signature may reveal with certainty, fully hide, or reveal partially who the signers are). In consensus protocols and blockchain applications, schemes that accommodate large numbers of parties have been put to use in the context of certifying the state of the ledger. In a "proof-of-stake" setting, each party is assigned a weight (corresponding to its stake), and the verifier needs to be assured that parties with sufficient stake have signed a message.

Most existing approaches to building large-scale multisignatures exploit properties of particular signatures or algebraic structures. For example, the recent results of Das et al. and Garg et al. [GJM+23,DCX+23] are based on bilinear pairings and require a structured setup.

In contrast, our work relies *only* on random oracles, making it compatible with any complexity assumption used for the underlying signature scheme, including ones that are post-quantum secure. Expectedly, the black box nature of our construction with respect to the underlying signature results in longer proofs (they can be shortened using succinct proof systems, as we discuss in Section 1.4).

In more detail, in order to apply an ALBA scheme to the problem of multisignatures, we treat individual signatures as set elements. The underlying signature scheme needs to be *unique*: it should be impossible (or computationally infeasible) to come up with two different signatures for the same message and public key. Otherwise, it is easy to come up with a set of sufficient total weight by producing multiple signatures for just a few keys[6]. Using an ALBA with decentralized provers is particularly suited to the blockchain setting as signatures will be collected from all participants.

A closely related approach is compact certificates by Micali et al. [MRV+21] who also treat the underlying signature scheme as a black box. In more detail, their construction collects all individual signatures in a Merkle tree, and selects a subset of signatures to reveal via lottery (that can be instantiated via the Fiat-Shamir transform [BR93]). Compared to compact certificates, our Telescope

[6] The verifier could check that all public keys are distinct, but since the proof contains just a small subset of the signatures, a malicious prover could try many signatures, or "grind," until it finds a proof that satisfies this check.

scheme obviates the need for the Merkle tree and hence shaves off a multiplicative logarithmic factor in the certificate length. It is also not susceptible to grinding while in compact certificates the adversary can try different signatures to include in the Merkle tree, and unlike compact certificates that rely inherently on the random oracle, our scheme can be instantiated in the CRS/URS model. Finally, our decentralized prover constructions drastically reduce communication. On the other hand, in compact certificates the lottery can be tied to public keys rather than signatures and hence can work with an arbitrary signature scheme (not necessarily unique).

Reducing communication complexity was also the focus of Chaidos and Kiayias in Mithril, a weighted threshold multisignature, [CK21], that also uses unique signatures and random-oracle based selection. In our terminology, Mithril applies an decentralized ALBA scheme to unique signatures (possibly followed by compactification via succinct proof systems, as discussed in Section 1.4). In comparison to Mithril, our decentralized prover construction achieves significantly smaller proof sizes (when comparing with the simple concatenation version of [CK21]) at the cost of higher communication. In Section 4.1 we present a simple lottery that is asymptotically similar to Mithril with concatenation proofs, and offer a comparison in Section 7.

Straight-Line Witness Extraction for SNARKS. Ganesh et al. [GKO$^+$23] addressed the problem of universal composability [Can00] for witness-succinct non-interactive arguments of knowledge. Universal composability requires the ability to extract the witness without rewinding the prover. However, since the proof is witness-succinct (i.e., shorter than the witness), the extractor must look elsewhere to obtain the witness. Building on the ideas of Pass [Pas03] and Fischlin [Fis05] Ganesh et al. proposed the following approach: the prover represents the witness as a polynomial of some degree d, uses a polynomial commitment to commit to it, and then makes multiple random oracle queries on evaluations of this polynomial (together with proofs that the evaluations are correct with respect to the commitment) until it obtains some rare output of the random oracle (much like a Bitcoin proof of work). The prover repeats this process many times, and includes in the proof only the queries that result in the rare outputs. The verifier can be assured that the prover made more than d queries with high probability, because otherwise it would not be able to obtain sufficiently many rare outputs. Thus, the knowledge extractor can reconstruct the witness via polynomial interpolation by simply observing the prover's random oracle queries.

We observe that this approach really involves the prover trying to convince the verifier that the size of the set of random oracle queries is greater than d. This approach is just an ALBA protocol, but not a particularly efficient one. Applying our scheme instead of the one custom-built by Ganesh et al. results in less work for the prover and shorter proofs. To get a proof of size $u \leq \lambda$, the protocol of Ganesh et al. requires the prover to compute $d \cdot u \cdot 2^{\lambda/u}$ polynomial

evaluations and decommitment proofs,[7] whereas our ALBA scheme requires only $d \cdot \lambda^{1/u} \cdot 2^{\lambda/u}$ of those,[8] a speed-up by a factor of about u.

1.4 Relation to General-Purpose Witness-Succinct Proofs

In cases where the *weight* function can be realized by a program, one can use general-purpose witness-succinct proofs to tackle the construction of ALBA schemes via utilizing SNARKs [Gro16,GWC19].

These general purpose tools, however, are quite expensive, especially for the prover. Because they require encoding the *weight* function as a circuit, their complexity depends heavily on the complexity of *weight*. Moreover, they are inapplicable when *weight* cannot be specified as a function ahead of time, but is evaluated by a more complex process — for example, via a secure multi-party computation protocol or a human judge weighing the strength of the evidence.

On the other hand, these tools can give very short, even constant-size, proofs. To get the best of both worlds — prover efficiency and constant-size proofs — one can combine an ALBA proof with a witness-succinct proof of knowledge of the ALBA proof. This is indeed the approach proposed by Chaidos and Kiayias [CK21]: it first reduces witness size n_f to u by using very fast random-oracle-based techniques, and then has the prover prove u (instead of n_f) *weight* computations. We can also apply this technique to our constructions, something that can result in a constant size proof with a computationally efficient prover. And given that our constructions can work in the CRS/URS model, one can avoid heuristically instantiating the random oracle inside a circuit.

2 Definitions

Below we present a definition of ALBA inspired by the non-interactive random oracle proof of knowledge (NIROPK) [BCS16]. To introduce arbitrary weights, we use a weight oracle $W : \{0,1\}^* \to \mathbb{N} \cup \{0\}$ and denote for a set S, $W(S) = \sum_{s \in S} W(s)$.

Definition 1. $(\mathsf{Prove}, \mathsf{Verify}, \mathsf{Extract})$ *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-NIROPK ALBA scheme if and only if*

- $\mathsf{Prove}^{H,W}$ *is a probabilistic expected polynomial time random oracle access program;*
- $\mathsf{Verify}^{H,W}$ *is a polynomial time program that has access to the random oracle* H *and a weight oracle* W*;*

[7] This value follows from the formula $\lambda = r(b - \log d)$ in the "Succinctness" paragraph of [GKO+23, Section 3.1]. Note that r is u in our notation, and the expected number of random oracle queries by the prover is $r \cdot 2^b$. Solving the formula for b, we get $2^b = d2^{\lambda/r}$.

[8] This value is obtained by setting $n_\mathrm{f} = d$ and solving (1) for n_p.

– $\mathsf{Extract}^{H,W,\mathcal{A}}$ *is a p.p.t. program that has access to a weight oracle* W*;* [9]
– *completeness: for all weight oracles* W *and all* S_p *such that* $W(S_p) \geq n_p$, $\Pr[\mathsf{Verify}^{H,W}(\mathsf{Prove}^{H,W}(S_p)) = 1] \geq 1 - 2^{-\lambda_{rel}}$*;*
– *proof of knowledge: consider the following experiment* $ExtractExp(\mathcal{A}^{H,W}, W)$*:*

$S_f \leftarrow \mathsf{Extract}^{H,W,\mathcal{A}}()$*;*
output 1 *iff* $W(S_f) > n_f$*;*

we require that for all weight oracles W *and all probabilistic oracle access programs* $\mathcal{A}^{H,W}$*,*

$$\Pr[ExtractExp(\mathcal{A}, W) = 1] \geq \Pr\left[\mathsf{Verify}^{H,W}\left(\mathcal{A}^{H,W}()\right) = 1\right] - 2^{-\lambda_{sec}}.$$

Additionally, we say the extractor $\mathsf{Extract}^{H,W,\mathcal{A}}$ *is straight-line if it is only allowed to run* $\mathcal{A}^{H,W}$ *once with the real* H *and* W *and only observes the transcript with its oracles.*

The above formulation of ALBAs captures the setting where a prover has the entire set S_p in its possession. We will also be interested in ALBAs where the prover is *decentralized* — by this we refer to a setting where a number of prover entities, each one possessing an element $s \in S_\mathrm{p}$ wish to act in coordination towards convincing the verifier. We now define a decentralized ALBA.

Definition 2. $(\mathsf{Prove}, \mathsf{Aggregate}, \mathsf{Verify}, \mathsf{Extract})$ *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-decentralized NIROPK ALBA scheme if and only if*

– $\mathsf{Prove}^{H,W}$ *is a p.p.t. random oracle access program;*
– $\mathsf{Aggregate}^{H,W}$ *is a probabilistic expected polynomial time random oracle access program;*
– $\mathsf{Verify}^{H,W}$ *is a polynomial time program that has access to the random oracle* H *and a weight oracle* W*;*
– $\mathsf{Extract}^{H,W,\mathcal{A}}$ *is a p.p.t. program that has access to a weight oracle* W*;* [10]
– *completeness: consider the following experiment* $CompExp(S_p, W)$*:*

$S \coloneqq \emptyset$*;*
for $s \in S_p$ ***do***
    $m \leftarrow \mathsf{Prove}^{H,W}$*;*
    if $m \neq \epsilon$ ***then***
        $S \coloneqq S \cup \{m\}$*;*
$\pi \leftarrow \mathsf{Aggregate}^{H,W}(S)$*;*
$r \leftarrow \mathsf{Verify}^{H,W}(\pi)$*;*
return r*;*

we require that for all weight oracles W *and all* S_p *such that* $W(S_p) \geq n_p$, $\Pr[CompExp(S_p, W) = 1] \geq 1 - 2^{-\lambda_{rel}}$*;*
– *proof of knowledge: consider the following experiment* $ExtractExp(\mathcal{A}^{H,W}, W)$*:*

[9] Here and above we assume W's output size is $\mathrm{poly}(\lambda_\mathrm{sec} + \lambda_\mathrm{rel})$. We assume $\mathsf{Extract}$ is also polynomial in the number of oracle queries that $\mathcal{A}^{H,W}$ makes.
[10] Here and above we assume W's output size is $\mathrm{poly}(\lambda_\mathrm{sec} + \lambda_\mathrm{rel})$. We assume $\mathsf{Extract}$ is also polynomial in the number of oracle queries that $\mathcal{A}^{H,W}$ makes.

$S_f \leftarrow \mathsf{Extract}^{H,W,\mathcal{A}}()$;
output 1 *iff* $W(S_f) > n_f$;

we require that for all weight oracles W *and all probabilistic oracle access programs* $\mathcal{A}^{H,W}$,

$$\Pr[\mathit{ExtractExp}(\mathcal{A}, W) = 1] \geq \Pr\left[\mathsf{Verify}^{H,W}\left(\mathcal{A}^{H,W}()\right) = 1\right] - 2^{-\lambda_{sec}}.$$

Additionally, we say the extractor $\mathsf{Extract}^{H,W,\mathcal{A}}$ *is straight-line if it is only allowed to run* $\mathcal{A}^{H,W}$ *once with the real* H *and* W *and only observes the transcript with its oracles.*

In this model, we would like to minimize not only the proof size, but also the amount of communication characterized by the size of S in CompExp. Note that the above definition can be extended to multiple rounds of communication, but this is not something we explore in this work - all our decentralized constructions are "1-round."

3 Telescope ALBA

In this section we present a sequence of two ALBA schemes. We start with a less efficient but simpler construction to illustrate the main idea. We then proceed to optimize the scheme's efficiency.

For both constructions, we will assume we have three random oracles H_0, H_1, and H_2 having particular output distributions. We explain how to implement these out of a single random oracle which outputs binary strings in Section B. Further, we assume the unweighted case and add weights in Section 5.

3.1 Basic Construction

The main idea is as follows. Let d, u and q be parameters. The prover first considers all pairs consisting of an integer in $[d]$ and one of the elements of S_p and selects each of the $n_\mathrm{p}d$ pairs with probability $1/n_\mathrm{p}$. In expectation he will have d pairs selected. Now these pairs are treated as single units and they are paired with each element of S_p, resulting in triples that are selected again with probability $1/n_\mathrm{p}$. This process is repeated u times ending with, in expectation, d tuples consisting of one integer in $[d]$ and u set elements. Now, each of the tuples is selected with probability q and any selected tuple will be a valid proof.

More formally, let W be a weight oracle, let H_1 and H_2 be random functions returning 1 with probability $1/n_\mathrm{p}$ and q respectively, and returning 0 otherwise. Any sequence $t, s_1, ..., s_u$ such that

- $1 \leq t \leq d$;
- for all $1 \leq i \leq u$, $H_1(t, s_1, ..., s_i) = 1$;
- $H_2(t, s_1, ..., s_u) = 1$;
- for all $1 \leq i \leq u$, $W(s_i) = 1$

is a valid proof (see Section 3.3 how to implement H_1 efficiently). Define the program Verify accordingly.

Intuitively, this works because the honest prover maintains d tuples in expectation at each stage, while the malicious prover's tuples decrease $n_{\mathrm{p}}/n_{\mathrm{f}}$ times with each stage. To implement Prove, simply run depth first search that tries to extend the current tuple by one more element.

We now state the main result of this section; it will follow from the proofs below. Taking Extract, defined as Algorithm 1 later in the section, we have

Theorem 1. *Using parameters from corollary 1,* (Prove, Verify, Extract) *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-NIROPK with* Extract *being a straight-line extractor.*

We first demonstrate a simple soundness property that ignores the complexities of the NIROPK definition. We define soundness to be the probability that a valid proof can be constructed using elements S_{f} with $|S_{\mathrm{f}}| = n_{\mathrm{f}}$. We then show how to deal with the adversary's adaptivity and build a straight-line NIROPK extractor at the end of the section.

Theorem 2. *Let*

$$u \geq \frac{\lambda_{sec} + \log(qd)}{\log \frac{n_p}{n_f}}.$$

Then soundness is $\leq 2^{-\lambda_{sec}}$.

Proof. We analyze soundness, denoted by S, using simple union bound.

$$S \leq \left(\frac{1}{n_{\mathrm{p}}}\right)^u \cdot q \cdot d \cdot n_{\mathrm{f}}^u = \left(\frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^u \cdot qd.$$

Then

$$-\log S \geq -\left(u \log \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}} + \log(qd)\right) = u \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} - \log(qd) \geq \lambda_{\mathrm{sec}}.$$

We now analyze completeness.

Theorem 3. *Let*

$$d \geq \frac{2u\lambda_{rel}}{\log e}; q = \frac{2\lambda_{rel}}{d \log e}.$$

Then completeness is $\geq 1 - 2^{-\lambda_{rel}}$ *and the probability that there exists a valid proof with a particular integer* t *is at least* $q - (u+1) \cdot \frac{q^2}{2}$.

Proof. Completeness can be described using the following recursive formula. For $0 \leq k \leq u$, let $f(k)$ be the probability that when fixing a prefix of an integer in $[d]$ and $u - k$ elements $t, s_1, ..., s_{u-k}$, there is no suffix of honest player's elements that works, meaning there is no $s_{u-k+1}, ..., s_u \in S_{\mathrm{p}}$ such that for all $u - k + 1 \leq i \leq u$, $H_1(t, s_1, ..., s_i) = 1$, and $H_2(t, s_1, ..., s_u) = 1$. Then one can see that

– $f(0) = 1 - q$;
– for $0 \leq k < u$, $f(k+1) = \big((1 - \frac{1}{n_\mathrm{p}}) + \frac{1}{n_\mathrm{p}} \cdot f(k)\big)^{n_\mathrm{p}}$;
– the probability that there does not exist a valid proof with a particular integer t is $f(u)$;
– the probability that the algorithm fails in the honest case is $\big(f(u)\big)^d$.

This recursive formula can be approximated:

$$f(k+1) = \left(1 + \frac{1}{n_\mathrm{p}}\big(f(k) - 1\big)\right)^{n_\mathrm{p}} \leq \left(e^{\frac{1}{n_\mathrm{p}}(f(k)-1)}\right)^{n_\mathrm{p}} = e^{f(k)-1}. \qquad (2)$$

It is convenient to look at the negative logarithm of this expression; we will prove by induction that $-\ln f(k) \geq q - k \cdot \frac{q^2}{2}$.
Basic case: $-\ln f(0) = -\ln(1-q) \geq -\ln(e^{-q}) = q$.
Inductive step: by equation 2,

$$-\ln f(k+1) \geq 1 - f(k) = 1 - e^{-\left(q - k \cdot \frac{q^2}{2}\right)} \geq$$
$$1 - \left(1 - \left(q - k \cdot \frac{q^2}{2}\right) + \frac{\left(q - k \cdot \frac{q^2}{2}\right)^2}{2}\right) \geq$$
$$\left(q - k \cdot \frac{q^2}{2}\right) - \frac{q^2}{2} = q - (k+1) \cdot \frac{q^2}{2}.$$

Hence, $-\ln f(u) \geq q - u \cdot \frac{q^2}{2}$ and the probability that the honest prover fails is $\big(f(u)\big)^d \leq \exp\Big(-\big(q - u \cdot \frac{q^2}{2}\big)d\Big)$. Using the values for d and q, one can see that this is at most $2^{-\lambda_\mathrm{rel}}$. Additionally, the probability that there exists a valid proof with a particular integer t is

$$1 - f(u) \geq e^{-\left(q - u \cdot \frac{q^2}{2}\right)} \geq$$
$$1 - \left(1 - \left(q - u \cdot \frac{q^2}{2}\right) + \frac{(q - u \cdot \frac{q^2}{2})^2}{2}\right) \geq q - (u+1) \cdot \frac{q^2}{2}.$$

Corollary 1. *Let*

$$u \geq \frac{\lambda_{sec} + \log \lambda_{rel} + 1 - \log\log e}{\log \frac{n_p}{n_f}}; d \geq \frac{2u\lambda_{rel}}{\log e}; q = \frac{2\lambda_{rel}}{d \log e}.$$

Then soundness is $\leq 2^{-\lambda_{sec}}$ *and completeness is* $\geq 1 - 2^{-\lambda_{rel}}$.

It is worth noting that the constant in d, and thus algorithm's running time, can be reduced. We show how to do it in Section C.1. Although the scheme still remains less efficient than the improved construction in Section 3.2, the optimizations can potentially be transferred over; we leave that for future work.

We now return to the issue of building a straight-line extractor that satisfies definition 1. For $H = (H_1, H_2)$, define

Algorithm 1 $\mathsf{Extract}^{H,W,\mathcal{A}}$

function $A_1^{H,W}$
    $\pi \leftarrow \mathcal{A}^{H,W}()$;
    $v \leftarrow \mathsf{Verify}^{H,W}(\pi)$;
    return π;
run $\mathcal{A}_1^{H,W}()$ and observe its oracles transcript τ;
$S_{\mathrm{f}} := \emptyset$;
for x queried to H_1 or H_2 in τ **do**
    if $W(x) = 1$ **then**
        add x to S_{f};
return S_{f}.

Theorem 4. *Define parameters as in theorem 2. Then* $\mathsf{Verify}^{H,W}$ *with* $\mathsf{Extract}^{H,W,\mathcal{A}}$ *satisfy the proof of knowledge property of definition 1. Additionally,* $\mathsf{Extract}^{H,W,\mathcal{A}}$ *is straight-line.*

Proof. The extractor succeeds whenever $\mathcal{A}$ succeeds, unless $\mathcal{A}$ succeeds after querying fewer than n_{f} elements of S, which happens with probability is at most $2^{-\lambda_{\mathrm{sec}}}$ by the following lemma. Thus, the proof of knowledge property follows by the union bound.

Proof. Let E_1 be the event that a valid proof can be made from the first n_{f} (or fewer) weight-1 elements that $\mathcal{A}_1^{H,W}$ queries to H and let E_2 be the event that $\mathcal{A}_1^{H,W}$ queries strictly more than n_{f} weight-1 elements to H. Then

$$\Pr\left[\mathsf{Verify}^{H,W}\left(\mathcal{A}^{H,W}()\right) = 1\right] = \Pr\left[v = 1\right][\leq]$$

$v = 1$ implies that τ contains weight-1 elements that can create a valid proof; then

$$\begin{gathered}
[\leq] \Pr\left[\mathcal{A}_1^{H,W}\text{terminates} \wedge \left(E_1 \vee E_2\right)\right] = \\
\Pr\left[\left(\mathcal{A}_1^{H,W}\text{terminates} \wedge E_1\right) \vee \left(\mathcal{A}_1^{H,W}\text{terminates} \wedge E_2\right)\right] \leq \\
\Pr\left[E_1 \vee \left(\mathcal{A}_1^{H,W}\text{terminates} \wedge E_2\right)\right] \leq \\
\Pr[E_1] + \Pr\left[\mathcal{A}_1^{H,W}\text{terminates} \wedge E_2\right] \leq \\
\Pr[E_1] + \Pr\left[\mathrm{ExtractExp}(\mathcal{A}, W) = 1\right] \leq \\
2^{-\lambda_{\mathrm{sec}}} + \Pr\left[\mathrm{ExtractExp}(\mathcal{A}, W) = 1\right]
\end{gathered}$$

where the last step follows from the following lemma.

Lemma 1. *Define parameters as in theorem 2 and let E be the event that a valid proof can be made from the first n_f (or less) weight-1 elements that $\mathcal{A}_1^{H,W}$ queries to H. Then* $\Pr[E] \leq 2^{-\lambda_{sec}}$.

Proof. Theorem 2 assumes a static set of random oracle queries, while an adaptive adversary may change the queries in response to random oracle answers. In order to be able to apply Theorem 2, we simply need to switch from thinking about X values as input to H to thinking about indices as inputs. We will define a new function Q to do so.

Let $X_1, \ldots, X_N$ be the first n_{f} distinct weight-1 elements that are present in random oracle queries of $\mathcal{A}$. If $N < n_{\mathrm{f}}$, pad the sequence $X_1, \ldots, X_N$ with dummy elements that are distinct from all queries of $\mathcal{A}$ up to n_{f}; the weights of those dummy elements do not matter. Define $Q(1, t, \cdots)$, and $Q(2, t, \cdots)$ to be the same as H_1 and H_2, respectively, but operating on indices rather than values of the Xs: That is, $Q(i, t, v_1, ..., v_j) = H_i\big(t, X_{v_1}, ..., X_{v_j}\big)$. Note that Q depends on $\mathcal{A}$, because the mapping from i to X_i is determined by $\mathcal{A}$. Partition the domain of Q into n_{f} parts, inductively, as follows: part k consists of all index sequences that contain the index k at least once and do not contain indices above k.

Let Q_k denote Q restricted to the kth part, and observe that Q_k is independent of $Q_1, \ldots, Q_{k-1}$ and is distributed identically to H_i, because it contains a new random oracle input X_k that is not contained in $Q_1, \ldots, Q_{k-1}$.

Let cert be true if and only if there are indices that "form a valid proof", i.e., and only if there exist $1 \le t \le d$ and $v_1, ..., v_u \in [n_{\mathrm{f}}]$ such that for all $1 \le i \le u$, $Q(1, t, v_1, ..., v_i) = 1$, and $Q(2, t, v_1, ..., v_u) = 1$. $\Pr[E] \le \Pr[\mathrm{cert}]$, because cert happens whenever E happens (and may also happen using some of the dummy values $X_{N+1} \ldots, X_{n_{\mathrm{f}}}$). And $\Pr[\mathrm{cert}] \le 2^{-\lambda_{\mathrm{sec}}}$ by the same exact argument as in Theorem 2.

Running time In this section we analyze the algorithm's running time.

Assume S_{p} is a set with cardinality n_{p}. All tuples $(j, s_1, ..., s_i)$ where $1 \le j \le d$, $1 \le i \le u$ and $s_1, ..., s_i \in S_{\mathrm{p}}$ can be represented as d trees of height u vertices. We would like to analyze the number of "accessible" vertices in these trees. Let the indicator random variable

$$A_{j,s_1,\ldots,s_i} = \begin{cases} 1 & \text{if for all } 1 \le r \le i,\ H_1(j, s_1, ..., s_r) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

If $A_{j,s_1,\ldots,s_i} = 1$ we say the vertex $(j, s_1, ..., s_i)$ is accessible.

Let us first prove that the expected number of accessible vertices in a single tree at a particular height is 1.

Theorem 5. *For any j and $1 \le i \le u$,*

$$\mathbb{E}\left[\sum_{s_1,\ldots,s_i \in S_p} A_{j,s_1,\ldots,s_i}\right] = 1.$$

Proof.

$$\mathbb{E}\left[\sum_{s_1,\ldots,s_i \in S_\mathrm{p}} A_{j,s_1,\ldots,s_i}\right] =$$
$$\sum_{s_1,\ldots,s_i \in S_\mathrm{p}} \mathbb{E}[A_{j,s_1,\ldots,s_i}] =$$
$$\sum_{s_1,\ldots,s_i \in S_\mathrm{p}} \mathbb{E}\left[\prod_{j=1}^{i} H_1(j, s_1, ..., s_j)\right] =$$
$$\sum_{s_1,\ldots,s_i \in S_\mathrm{p}} \prod_{j=1}^{i} \mathbb{E}\left[H_1(j, s_1, ..., s_j)\right] =$$
$$\sum_{s_1,\ldots,s_i \in S_\mathrm{p}} \prod_{j=1}^{i} \frac{1}{n_\mathrm{p}} =$$
$$\sum_{s_1,\ldots,s_i \in S_\mathrm{p}} \left(\frac{1}{n_\mathrm{p}}\right)^i =$$
$$n_\mathrm{p}^i \cdot \left(\frac{1}{n_\mathrm{p}}\right)^i =$$
$$1.$$

Assuming the algorithm implements DFS, theorem 3 gives a bound on the expected number of evaluated trees. And by the above theorem, the algorithm invokes H_1 $n_\mathrm{p}u$ times and H_2 once in expectation per tree. Thus, the expected total number of hash evaluations shall be the product of the expected number of evaluated trees and $(n_\mathrm{p}u + 1)$. This, however, needs a more careful proof.

Theorem 6. *The expected number of hash evaluations is at most*

$$\left(q - (u+1) \cdot \frac{q^2}{2}\right)^{-1} (n_p u + 1)$$

Proof. Modify the algorithm so that it keeps evaluating trees until a valid certificate is found without limit on j. Clearly, such an algorithm will have a larger expected number of hash invocations.

Define N to be the number of evaluated trees, X_j to be the number of hash invocations in tree j, and let t be the expected number of hash invocations by

the modified algorithm. Then

$$t = \mathbb{E}\left[\sum_{j=1}^{N} X_j\right] =$$

$$\mathbb{E}\left[\sum_{j=1}^{N} X_j \middle| N = 1\right] \cdot \Pr[N = 1] + \mathbb{E}\left[\sum_{j=1}^{N} X_j \middle| N \neq 1\right] \cdot \Pr[N \neq 1] =$$

$$\mathbb{E}[X_1 | N = 1] \cdot \Pr[N = 1] + \mathbb{E}\left[X_1 + \sum_{j=2}^{N} X_j \middle| N \neq 1\right] \cdot \Pr[N \neq 1] =$$

$$\mathbb{E}[X_1 | N = 1] \cdot \Pr[N = 1] + \mathbb{E}[X_1 | N \neq 1] \cdot \Pr[N \neq 1] +$$

$$\mathbb{E}\left[\sum_{j=2}^{N} X_j \middle| N \geq 2\right] \cdot \Pr[N \neq 1] =$$

$$\mathbb{E}[X_1] + t \cdot \Pr[N \neq 1] \leq$$

$$n_{\mathrm{p}} u + 1 + t \cdot \Pr[N \neq 1].$$

Therefore, $t \cdot \Pr[N = 1] \leq n_{\mathrm{p}} u + 1$ and $t \leq \frac{n_{\mathrm{p}} u + 1}{\Pr[N=1]}$.

By theorem 3,

$$\Pr[N = 1] \geq q - (u + 1) \cdot \frac{q^2}{2}$$

from which the statement of the theorem follows.

Taking parameter values from corollary 1 and letting $\lambda = \lambda_{\mathrm{sec}} = \lambda_{\mathrm{rel}}$, we thus obtain an expected number of hash evaluations of $O(\lambda^2 \cdot n_{\mathrm{p}})$.

We might also wish to have a tighter bound on the running time or on the number of accessible vertices to argue that an adversary cannot exploit an imperfect hash function or a PRF by making too many queries. Below we present a Chernoff style bound on the number of accessible vertices in all d trees

$$Z = \sum_{\substack{1 \leq j \leq d, \\ 1 \leq i \leq u, \\ s_1, \ldots, s_i \in S_{\mathrm{p}}}} A_{j, s_1, \ldots, s_i}.$$

Note that $\mathbb{E}[Z] = du$.

Theorem 7.

$$\Pr[Z \geq (1 + \delta) du] \leq \exp\left(- \frac{\delta^2}{4(1 + \delta)} \cdot \frac{d}{u} \right).$$

Proof. Let $t > 0$ and define the sequence $\{x_k\}$ as follows: let $x_0 = 1$ and for $k \geq 0$, let

$$x_{k+1} = \left(\frac{1}{n} x_k e^t + 1 - \frac{1}{n} \right)^{n_{\mathrm{p}}}.$$

By lemma 9, $\mathbb{E}[e^{tZ}] = x_u^d$.

Define the following sequence $\{y_k\}$: let $y_0 = 0$ and $y_{k+1} = y_k + t + (y_k + t)^2$. We will prove by induction that if $y_u \leq 1$ then for all $0 \leq k \leq u$, $x_k \leq e^{y_k}$.
Basis case: $x_0 = 1 \leq 1 = e^{y_0}$. Inductive step: $x_{k+1} = \left(\frac{1}{n}x_k e^t + 1 - \frac{1}{n}\right) = \left(1 + \frac{1}{n}\left(x_k e^t - 1\right)\right) \leq \left(e^{\frac{1}{n}(x_k e^t - 1)}\right)^n = \exp\left(x_k e^t - 1\right) \leq \exp\left(e^{y_k + t} - 1\right)$. Since $y_k + t \leq y_k + t + (y_k + t)^2 = y_{k+1} \leq y_u \leq 1$, $x_{k+1} \leq \exp\left(1 + y_k + t + (y_k + t)^2 - 1\right) = \exp(y_{k+1})$.
Hence, $\mathbb{E}[e^{tZ}] \leq e^{y_u d}$.

By Markov's inequality,

$$\begin{gathered}\Pr\left[Z \geq (1+\delta)du\right] = \Pr\left[e^{tZ} \geq e^{(1+\delta)tdu}\right] \leq \\ \frac{e^{y_u d}}{e^{(1+\delta)tdu}} = \exp\Big(-d\big((1+\delta)tu - y_u\big)\Big).\end{gathered} \tag{3}$$

We now need to find some t and y_u that maximize $(1+\delta)tu - y_u$. However, instead of picking a suitable t and finding a bound for y_u in terms of it, we do the opposite. We first choose an upper bound α for y_u and then calculate a suitable t. We use the observation that $y_k \geq y_{k+1} - y_{k+1}^2 - t \geq y_{k+1} - y_u^2 - t$. Details follow.

Let $\alpha < \frac{1}{u}$ and t be such that $\alpha - u\alpha^2 - ut = 0$ (i.e., $t = \frac{\alpha}{u} - \alpha^2$); it can be seen that $t > 0$. We will prove by induction that $y_k \leq \frac{\alpha k}{u}$.
Basis step: $y_0 = 0 \leq 0 = \frac{\alpha \cdot 0}{u}$.
Inductive step: $y_{k+1} = y_k + t + (y_k + t)^2 \leq \frac{\alpha k}{u} + \frac{\alpha}{u} - \alpha^2 + \left(\frac{\alpha k}{u} + \frac{\alpha}{u} - \alpha^2\right)^2 = \frac{\alpha(k+1)}{u} - \alpha^2 + \left(\frac{\alpha(k+1)}{u} - \alpha^2\right)^2 \leq \frac{\alpha(k+1)}{u} - \alpha^2 + \left(\alpha - \alpha^2\right)^2 \leq \frac{\alpha(k+1)}{u} - \alpha^2 + \alpha^2 = \frac{\alpha(k+1)}{u}$.
Hence, $y_u \leq \alpha$.

Then $(1+\delta)tu - y_u \geq (1+\delta)tu - \alpha = (1+\delta)\left(\frac{\alpha}{u} - \alpha^2\right)u - \alpha$. Differentiating with respect to α, we find that this expression is maximized when

$$\alpha = \frac{\delta}{2(1+\delta)u}.$$

It is easily verified that $\alpha < \frac{1}{u}$.

Therefore,

$$\begin{gathered}(1+\delta)tu - y_u \geq \\ (1+\delta)\left(\frac{\delta}{2(1+\delta)u^2} - \frac{\delta^2}{4(1+\delta)^2u^2}\right)u - \frac{\delta}{2(1+\delta)u} = \\ \frac{\delta}{2u} - \frac{\delta^2}{4(1+\delta)u} - \frac{\delta}{2(1+\delta)u} = \\ \frac{2\delta(1+\delta) - \delta^2 - 2\delta}{4(1+\delta)u} = \\ \frac{\delta^2}{4(1+\delta)u}.\end{gathered}$$

Hence by equation 3,

$$\Pr\left[Z \geq (1+\delta)du\right] \leq \exp\left(-\frac{\delta^2}{4(1+\delta)} \cdot \frac{d}{u}\right).$$

Taking parameter values from corollary 1 and letting $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$, we thus conclude that the algorithm does $O(\lambda^3 \cdot n_{\text{p}})$ hash evaluations with overwhelming probability.

3.2 Construction with Prehashing

The basic scheme described above has proving expected time $O(\lambda^2 \cdot n_{\text{p}})$ and verification time $O(\lambda)$ if we let $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$. The modification described in this section has proving expected time $O(\lambda^2 + n_{\text{p}})$ and verification time is unchanged.

The improvement is inspired by bins-and-balls collisions. Whereas in the previous scheme for every tuple we tried each of n_{p} possible extensions, here we hash tuples to a uniform value in $[n_{\text{p}}]$ and hash individual set elements to a uniform value in $[n_{\text{p}}]$, and consider a valid extension to be such that the tuple and the extension hash to the same value. More formally, we have a weight oracle W, random functions H_0 and H_1 producing a uniformly random value in $[n_{\text{p}}]$ and hash function H_2 returning 1 with probability q and 0 otherwise, and consider a tuple $(t, s_1, ..., s_u)$ a valid proof if and only if

- $1 \leq t \leq d$;
- for all $1 \leq i \leq u$, $H_1(t, s_1, ..., s_{i-1}) = H_0(s_i)$;
- $H_2(t, s_1, ..., s_u) = 1$;
- for all $1 \leq i \leq u$, $W(s_i) = 1$.

(see Section 3.3 how to implement H_1 efficiently). Define the Verify program accordingly.

As before, we have d valid tuples in expectation at each stage but by precomputing $H_0(\cdot)$ (balls to bins) we avoid trying all n_{p} extensions for a tuple. The analysis of completeness, however, is more complicated. Before, we assumed in the recursive formula that failure events for each element extension are all independent. Here, it is not true: the fact that one extension eventually succeeds can tell that the balls-to-bins are well distributed. Indeed, if each bin gets exactly one ball, then there will always be a tuple that succeeds except maybe for the requirement that $H_2(\cdot) = 1$. However, if all balls land in one bin, then the success probability is smaller. To get rid of this dependency, we can however fix the balls-to-bins arrangement. Then such events become independent again.

The proof has two parts: the first one says that if the arrangement of the balls is "nice", then with high probability the honest player succeeds. The second part proves that we get a "nice" distribution of balls with high probability. The "nice" property itself is artificial, but one can notice that if the number of bins of size s is exactly the expected number of bins of size s if the size of each bin is a Poisson random variable with mean 1, then the analysis of completeness becomes

very similar to that of the previous scheme. By using Poisson approximation, we can show that the property we care about does hold with high probability.

We need, however, assume that the number of set elements n_{p} is large enough (on the order of λ^3). Alternatively, we can generate multiple balls per set element.

We now state the main result of this section; it will follow from the proofs below. Taking Extract, defined as Algorithm 1 earlier in the section, we have

Theorem 8. *Using parameters from corollary 2,* (Prove, Verify, Extract) *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-NIROPK with* Extract *being a straight-line extractor.*

We first demonstrate a simple soundness property that ignores the complexities of the NIROPK definition. We define soundness to be the probability that a valid proof can be constructed using elements S_{f} with $|S_{\mathrm{f}}| = n_{\mathrm{f}}$. We then show how to deal with the adversary's adaptivity and build a straight-line NIROPK extractor at the end of the section.

Theorem 9. *Let*

$$u \geq \frac{\lambda_{sec} + \log(qd)}{\log \frac{n_p}{n_f}}.$$

Then soundness is $\leq 2^{-\lambda_{sec}}$.

Proof. By union bound, the probability S that a valid proof can be constructed out of n_{f} elements is at most

$$\left(\frac{1}{n_{\mathrm{p}}}\right)^u \cdot q \cdot d \cdot n_{\mathrm{f}}^u = \left(\frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^u \cdot qd.$$

Then

$$-\log S \geq -\left(u \log \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}} + \log(qd)\right) = u \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} - \log(qd) \geq \lambda_{\mathrm{sec}}.$$

Theorem 10. *Assume*

$$d \geq \frac{16u(\lambda_{rel} + \log 3)}{\log e}; q = \frac{2(\lambda_{rel} + \log 3)}{d \log e}; n_p \geq \frac{d^2 \log e}{9(\lambda_{rel} + \log 3)}.$$

Then completeness is $\geq 1 - 2^{-\lambda_{rel}}$.

Proof. Let $X_i = |\{s \in S_{\mathrm{p}} : H_0(s) = i\}|$ be the number of balls in bin i, let E be the event that $\frac{1}{n_{\mathrm{p}}} \sum_{i=1}^{n} e^{-qX_i} \leq e^{-q+4q^2}$ and let F be the event that the honest prover fails. By lemma 10 with $\lambda := \frac{\lambda_{\mathrm{rel}} + \log 3}{\log e}$, $\Pr[F|E] \leq \frac{1}{3} \cdot 2^{-\lambda_{\mathrm{rel}}}$. Also

by lemma 13,

$$\Pr\left[\bar{E}\right] =$$
$$\Pr\left[\frac{1}{n_{\mathrm{p}}}\sum_{i=1}^{n} e^{-qX_i} > e^{-q+4q^2}\right] \leq$$
$$\Pr\left[\frac{1}{n_{\mathrm{p}}}\sum_{i=1}^{n} e^{-qX_i} \geq 1-q+4q^2\right] \leq$$
$$2e^{-\frac{9}{4}n_{\mathrm{p}}q^2}.$$

This is at most $\frac{2}{3}\cdot 2^{-\lambda_{\mathrm{rel}}}$ if and only if

$$3\cdot 2^{\lambda_{\mathrm{rel}}} \leq e^{\frac{9}{4}n_{\mathrm{p}}q^2} \iff$$
$$\frac{9}{4}\log e\cdot n_{\mathrm{p}}q^2 \geq \lambda_{\mathrm{rel}} + \log 3 \iff$$
$$n_{\mathrm{p}} \geq \frac{4(\lambda_{\mathrm{rel}}+\log 3)}{9\log e\cdot q^2} \iff$$
$$n_{\mathrm{p}} \geq \frac{4(\lambda_{\mathrm{rel}}+\log 3)}{9\log e\cdot \left(\frac{2(\lambda_{\mathrm{rel}}+\log 3)}{d\log e}\right)^2} \iff$$
$$n_{\mathrm{p}} \geq \frac{4(\lambda_{\mathrm{rel}}+\log 3)}{9\log e\cdot \frac{4(\lambda_{\mathrm{rel}}+\log 3)^2}{d^2\log^2 e}} \iff$$
$$n_{\mathrm{p}} \geq \frac{d^2\log e}{9(\lambda_{\mathrm{rel}}+\log e)}$$

which is true by our assumption about n_{p}.

Hence, $\Pr[F] \leq 2^{-\lambda_{\mathrm{rel}}}$.

Corollary 2. *Assume*

$$u \geq \frac{\lambda_{sec}+\log(\lambda_{rel}+\log 3)+1-\log\log e}{\log\frac{n_p}{n_f}}; d \geq \frac{16u(\lambda_{rel}+\log 3)}{\log e};$$
$$q = \frac{2(\lambda_{rel}+\log 3)}{d\log e}; n_p \geq \frac{d^2\log e}{8(\lambda_{rel}+\log 3)}.$$

Then soundness is $\leq 2^{-\lambda_{sec}}$ *and completeness is* $\geq 1-2^{-\lambda_{rel}}$.

The last step is showing a straight-line extractor for the improved scheme. This is done as in Section 3.1. For $H=(H_0,H_1,H_2)$, define $\mathsf{Extract}^{H,W,\mathcal{A}}$ similarly to that section. We have the following theorem whose proof is the same as the proof of theorem 4.

Theorem 11. *Define parameters as in theorem 9. Then* $\mathsf{Verify}^{H,W}$ *with* $\mathsf{Extract}^{H,W,\mathcal{A}}$ *satisfy the proof of knowledge property of definition 1. Additionally,* $\mathsf{Extract}^{H,W,\mathcal{A}}$ *is straight-line.*

It is worth noting that the $n_{\mathrm{p}} \geq \Omega(\lambda^3)$ requirement can be removed as follows. One can use Markov's inequality to show that the event E in theorem 10 happens with probability e.g. $\frac{3}{4}$ to achieve a scheme with completeness $\frac{1}{2}$. Such a scheme can then be amplified to achieve arbitrary λ_{rel} by setting $\lambda_{\mathrm{sec}} \coloneqq \lambda_{\mathrm{sec}} + \log \lambda_{\mathrm{rel}}$ and having the verifier accept any one of λ_{rel} independent proofs. As a result, the expected running time is no longer $\Omega(\lambda^3)$ but $O(\lambda^2 + n_{\mathrm{p}})$, but we need to apply H_0 to all elements twice in expectation as opposed to exactly ones. Hence, for large n_{p} it still makes sense to use the algorithm as described in the beginning of the section.

Running time In this section we analyze the algorithm's running time. Assume S_{p} is a set with cardinality n_{p}. As described in section 3.1, all tuples $(j, s_1, ..., s_i)$ where $1 \leq j \leq d$, $1 \leq i \leq u$ and $s_1, ..., s_i \in S_{\mathrm{p}}$ can be represented as d trees of height u vertices. We would like to analyze the number of "accessible" vertices in these trees. Let the indicator random variable

$$A_{j,s_1,...,s_i} = \begin{cases} 1 & \text{if for all } 1 \leq r \leq i,\ H_1(j, s_1, ..., s_{r-1}) = H_0(s_i) \\ 0 & \text{otherwise.} \end{cases}$$

If $A_{j,s_1,...,s_i} = 1$ we say the vertex $(j, s_1, ..., s_i)$ is accessible.

Similarly to section 3.1, one can prove that the expected number of accessible vertices in a single tree at a particular height is 1. This holds independent of the value of H_0!

Theorem 12. *For any j and $1 \leq i \leq u$,*

$$\mathbb{E}\left[\sum_{s_1,...,s_i \in S_p} A_{j,s_1,...,s_i} \middle| H_0\right] = 1.$$

We will now analyze the expected running time of the algorithm. The hash function H_0 is invoked exactly n_{p} times, so we will only upper bound the expected number of invocations of H_1 and H_2.

Theorem 13. *The expected number of invocations of H_1 and H_2 is at most*

$$\frac{u+1}{1 - e^{-q+4uq^2}} + 2e^{-\frac{9}{4}n_p q^2} \cdot d(u+1).$$

Proof. Let R be the number of invocations of H_1 and H_2, let $X_i = |\{s \in S_{\mathrm{p}} : H_0(s) = i\}|$ be the number of balls in bin i, define random function $f(x) = \frac{1}{n_{\mathrm{p}}} \sum_{i=1}^{n_{\mathrm{p}}} x^{X_i}$ and let E be the event that $f(e^{-q}) \leq e^{-q+4q^2}$. Then

$$\mathbb{E}[R] = \mathbb{E}[R|E] \cdot \Pr[E] + \mathbb{E}[R|\neg E] \cdot \Pr[\neg E][\leq]$$

By the above theorem, evaluating a single tree takes in expectation at most $u+1$ invocations of H_1 and H_2. Thus,

$$[\leq] \mathbb{E}[R|E] + d(u+1) \cdot \Pr[\neg E][\leq]$$

And by lemma 13,

$$[\leq] \mathbb{E}[R|E] + 2e^{-\frac{9}{4}n_{\mathrm{p}}q^2} \cdot d(u+1).$$

Let $F(t)$ be the event that there is no valid certificate with integer t. Then by lemma 11 for all t, $\Pr[F(t)|H_0] = f^{(u)}(1-q) \leq f^{(u)}(e^{-q})$. By lemma 12, this is at most $e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}}\right)^u$. Then $\Pr\left[F(t)\middle|E\right] \leq e^{-q+4uq^2}$. Then similarly to theorem 6,

$$\mathbb{E}[R|E] \leq \frac{1}{1-e^{-q+4uq^2}} \cdot (u+1)$$

and hence

$$\mathbb{E}[R] \leq \frac{u+1}{1-e^{-q+4uq^2}} + 2e^{-\frac{9}{4}n_{\mathrm{p}}q^2} \cdot d(u+1).$$

Taking parameter values from corollary 2 and letting $\lambda = \lambda_{\mathrm{sec}} = \lambda_{\mathrm{rel}}$, we thus get expected number of evaluations of H_1 and H_2 $O(\lambda^2)$. This is dominated by n_{p} invocations of H_0 since n_{p} is assumed to be $\Omega(\lambda^3)$.

Below we also present a tight bound on the number of accessible vertices in all d trees

$$Z = \sum_{\substack{1\leq j\leq d,\\ 1\leq i\leq u,\\ s_1,\ldots,s_i\in S_{\mathrm{p}}}} A_{j,s_1,\ldots,s_i}.$$

Note that $\mathbb{E}[Z] = du$.

Theorem 14. *Let*

$$\lambda > 0; \quad \lambda' = \frac{\lambda+2}{\log e}; \quad n_p \geq \frac{u^2\lambda'}{2};$$
$$u! \cdot \frac{u - e^{\frac{1}{3u}}}{u^3} \geq 72e^{-\frac{2}{3}} \cdot 2^{\lambda}; \quad \delta = e^{1+\sqrt{\frac{18u^2\lambda'}{n_p}}}\left(\frac{3u\lambda'}{d}+1\right).$$

Then

$$\Pr[Z \geq \delta du] \leq 2^{-\lambda}.$$

Proof. Let

$$0 < \alpha \leq \frac{1}{2u},$$
$$c = \frac{3}{\alpha} \cdot \sqrt{\frac{2\lambda'}{n_{\mathrm{p}}}} + 3,$$
$$\delta = e^{cu\alpha}\left(\frac{\lambda'}{d\alpha}+1\right),$$

$X_i = |\{s \in S_\mathrm{p} : H_0(s) = i\}|$ be the number of balls in bin i and let E be the event that $\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} e^{\alpha \cdot X_i} \leq e^{\alpha + c\alpha^2}$ with $\Pr[E] > 0$. By lemma 14,

$$\Pr[Z \geq \delta du | E] \leq e^{-\lambda'} = \frac{2^{-\lambda}}{4}. \tag{4}$$

Also define Y_i to be Poisson random variables with expectation 1,

$$A_i = \begin{cases} e^{\alpha Y_i} & \text{if } Y_i \leq u \\ 0 & \text{otherwise} \end{cases}$$

and

$$B_i = \begin{cases} 0 & \text{if } Y_i \leq u \\ e^{\frac{Y_i}{3u}} & \text{otherwise.} \end{cases}$$

Since $1 + \alpha + c\alpha^2 \leq e^{\alpha + c\alpha^2}$,

$$\Pr\left[\bar{E}\right] \leq \Pr\left[\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} e^{\alpha \cdot X_i} \geq 1 + \alpha + c\alpha^2\right] [\leq]$$

By Poisson approximation [MU05, theorem 5.10],

$$[\leq] 2 \cdot \Pr\left[\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} e^{\alpha \cdot Y_i} \geq 1 + \alpha + c\alpha^2\right] =$$
$$2 \cdot \Pr\left[\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} (A_i + B_i) \geq 1 + \alpha + c\alpha^2\right] =$$
$$2 \cdot \Pr\left[\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} A_i + \frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} B_i \geq 1 + \alpha + c\alpha^2\right] \leq$$
$$2 \cdot \left(\Pr\left[\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} A_i \geq 1 + \alpha + (c-1)\alpha^2\right] + \Pr\left[\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} B_i \geq \alpha^2\right]\right)$$

By lemma 16,

$$\Pr\left[\frac{1}{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{p}} A_i \geq 1 + \alpha + (c-1)\alpha^2\right] \leq e^{-\lambda'} = \frac{2^{-\lambda}}{4}.$$

We would like to choose the value of α that optimizes

$$\delta = e^{cu\alpha}\left(\frac{\lambda'}{d\alpha} + 1\right).$$

Ignoring the "+1" term, we differentiate $e^{cu\alpha} \cdot \frac{\lambda'}{d\alpha}$ with respect to α and find that it is minimized when $\alpha = \frac{1}{3u}$. Then δ becomes

$$e^{1+\sqrt{\frac{18u^2\lambda'}{n_\mathrm{p}}}}\left(\frac{3u\lambda'}{d}+1\right).$$

Finally, by lemma 17,

$$\Pr\left[\frac{1}{n_\mathrm{p}}\sum_{i=1}^{n_\mathrm{p}} B_i \geq \alpha^2\right] \leq 2^{-\lambda-3}$$

provided

$$u! \cdot \frac{u - e^{\frac{1}{3u}}}{u^3} \geq 9e^{-\frac{2}{3}} \cdot 2^{\lambda+3}$$

which is true by the statement of the theorem.

Hence $\Pr[\bar{E}] \leq 2\left(\frac{2^{-\lambda}}{4} + \frac{2^{-\lambda}}{8}\right) = \frac{3}{4} \cdot 2^{-\lambda}$. Combined with equation 4, we conclude

$$\Pr[Z \geq \delta du] \leq 2^{-\lambda}.$$

Taking parameter values from corollary 2 and letting $\lambda = \lambda_{\mathrm{sec}} = \lambda_{\mathrm{rel}}$, we thus conclude that the algorithm evaluates H_1 and H_2 $O(\lambda^3)$ times with overwhelming probability.

3.3 Implementing Random Oracles with Long Inputs

We describe our protocols assuming a random oracle H_1 that can accommodate inputs of any length, which, in particular, implies independence of outputs for inputs of different lengths. However, to have an accurate accounting for running times, one has to charge for the cost of running a random oracle in proportion to the input length. Because the Telescope construction runs $H_1(j)$, $H_1(j, s_1)$, $H_1(j, s_1, s_2)$, $H_1(j, s_1, s_2 \ldots, s_u)$, the cost of just one u-tuple is quadratic in u. To reduce this cost to linear (thus saving a factor of u in running time), we will implement $H_1(j, s_1, \ldots, s_{i+1})$ to reuse most of the computation of $H_1(j, s_1, \ldots, s_i)$. The most natural way to do so is to slightly modify the Merkle-Damgård construction: use a two-input random oracle f ("compression function") with a sufficiently long output and a function g that maps the range of f to the distribution needed by H_1 (see Section B for how we implement g). Inductively define $H'_1(j, s_1, \ldots, s_{i+1}) = f(H'_1(j, s_1, \ldots, s_i), s_{i+1})$ and let $H_1(x) = g(H'_1(x))$.

While not indifferentiable from a random oracle (see Coron et al. [CDMP05] for similar constructions that are), this construction suffices for our soundness and extractability arguments, because those arguments need independence only for a single chain (they handle multiple different chains by the union bound). Neither length extension attacks nor collisions are important. Completeness suffers very slightly by the probability of f-collisions, which can be made negligible by making the output of f large enough and using the bound on the number of queries made by the honest prover (theorems 7 and 14).

3.4 Optimality of the certificate size

In this section, we show that the number of set elements u included in a proof is essentially optimal for our constructions. Because our construction works for a black-box weight function that formally is implemented via an oracle (and in reality may be implemented by MPC, a human judge, etc.), the verifier must query the weight function on some values; else the verifier has no knowledge of whether any values in the prover's possession have any weight.

Thus, for the sake of proving optimality, we consider only protocols that make this part of verification explicit. We define an algorithm Read that takes a proof and returns set elements; these set elements must have been in the prover's possession. We bound the proof size in terms of the number of set elements returned by Read, showing that if it is too small, the protocol cannot be secure.

We emphasize that some nonstandard definition is necessary for the lower bound, because if the weight function can be specified by a polynomial-size circuit, then witness-succinct SNARKs can be used to give a proof whose size is independent of n_{p} and n_{f} (though there are likely barriers to extractability [CGKS22] and the known way to achieve extractability is through an ALBA-like construction of lower efficiency than ours [GKO+23]). We also note that the following definition can be used for upper bound results too, as demonstrated in Section 6 for the CRS model.

Definition 3. $(\mathsf{Prove}, \mathsf{Read}, \mathsf{Verify})$ *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-ALBA scheme if and only if*

- Prove^H *is a probabilistic expected polynomial time random oracle access program;*
- Verify^H *is polynomial time random oracle access program;*
- Read *is a polynomial time program;*
- *completeness: consider the following experiment* $CompExp(S_p)$*:*

  $\pi \leftarrow \mathsf{Prove}^H(S_p)$*;*

  output 1 *iff* $\mathsf{Read}(\pi) \subseteq S_p$ *and* $\mathsf{Verify}^H(\pi) = 1$*;*

  we require that for all sets S_p *with size* $\geq n_p$*,* $\Pr[CompExp(S_p) = 1] \geq 1 - 2^{-\lambda_{rel}}$*.*
- *soundness: consider the following experiment* $SoundExp(S_f)$*:*

  output 1 *iff* $\exists \pi, \mathsf{Read}(\pi) \subseteq S_f \wedge \mathsf{Verify}^H(\pi) = 1$*;*

  we require that for all sets S_f *with size* $\leq n_f$*,* $\Pr[SoundExp(S_f) = 1] \leq 2^{-\lambda_{sec}}$*;*

We now prove a lower bound for a scheme satisfying this definition.

Theorem 15. *Assume* $\lambda_{rel} \geq 1$*, define* $\alpha = \frac{\lambda_{sec}-3}{\log \frac{n_p}{n_f}}$*, assume* $n_f \geq 3\alpha^2$*, let* S_p *be an arbitrary set of size* n_p*, and let* $(\mathsf{Prove}, \mathsf{Read}, \mathsf{Verify})$ *be a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-proof of cardinality scheme. Then*

$$\Pr\left[\left|\mathsf{Read}(\mathsf{Prove}^H(S_p))\right| > \alpha\right] \geq \frac{1}{4}.$$

Proof. Suppose not and define $u = \lfloor \alpha \rfloor$. Then $\Pr\left[\left|\mathsf{Read}(\mathsf{Prove}^H(S_\mathrm{p}))\right| \leq u\right] \geq \frac{3}{4}$.

Let $\pi \leftarrow \mathsf{Prove}^H(S_\mathrm{p})$, S_f be a uniformly random subset of S_p of size n_f, A be the event that $|\mathsf{Read}(\pi)| \leq u$ and B the event that $\mathsf{Read}(\pi) \subseteq S_\mathrm{p} \wedge \mathsf{Verify}^H(\pi) = 1$. By the above, $\Pr[A] \geq \frac{3}{4}$, and by completeness, $\Pr[B] \geq \frac{1}{2}$. Then

$$\begin{gathered}
\Pr\left[\mathsf{Read}(\pi) \subseteq S_\mathrm{f} \wedge \mathsf{Verify}^H(\pi) = 1\right] \geq \\
\Pr\left[\mathsf{Read}(\pi) \subseteq S_\mathrm{f} \wedge \mathsf{Verify}^H(\pi) = 1 \middle| A \wedge B\right] \cdot \Pr\left[A \wedge B\right] = \\
\Pr\left[\mathsf{Read}(\pi) \subseteq S_\mathrm{f} \middle| A \wedge B\right] \cdot \Pr[A \wedge B] \geq \\
\Pr\left[\mathsf{Read}(\pi) \subseteq S_\mathrm{f} \middle| A \wedge B\right] \cdot \left(\Pr[A] + \Pr[B] - 1\right) \geq \\
\Pr\left[\mathsf{Read}(\pi) \subseteq S_\mathrm{f} \middle| A \wedge B\right] \cdot \left(\frac{3}{4} + \frac{1}{2} - 1\right) = \\
\frac{1}{4} \cdot \Pr\left[\mathsf{Read}(\pi) \subseteq S_\mathrm{f} \middle| A \wedge B\right] \geq \\
\frac{1}{4} \cdot \frac{n_\mathrm{f}}{n_\mathrm{p}} \cdot \frac{n_\mathrm{f}-1}{n_\mathrm{p}-1} \times ... \times \frac{n_\mathrm{f}-(u-1)}{n_\mathrm{p}-(u-1)} \geq \\
\frac{1}{4} \cdot \left(\frac{n_\mathrm{f}-u}{n_\mathrm{p}}\right)^u = \\
\frac{1}{4} \cdot \left(\frac{n_\mathrm{f}}{n_\mathrm{p}}\right)^u \cdot \left(\frac{n_\mathrm{f}-u}{n_\mathrm{f}}\right)^u = \\
\frac{1}{4} \cdot \left(\frac{n_\mathrm{f}}{n_\mathrm{p}}\right)^u \cdot \left(1 - \frac{u}{n_\mathrm{f}}\right)^u [\geq]
\end{gathered}$$

Since $\frac{u}{n_\mathrm{f}} \leq \frac{u}{3\alpha^2} \leq \frac{u}{3u^2} \leq \frac{1}{2}$ and $1 - x \geq e^{-x-x^2} \geq e^{-\frac{3}{2}x}$ for $0 \leq x \leq \frac{1}{2}$,

$$\begin{gathered}
[\geq] \frac{1}{4} \cdot \left(\frac{n_\mathrm{f}}{n_\mathrm{p}}\right)^u \cdot \left(e^{-\frac{3u}{2n_\mathrm{f}}}\right)^u \geq \\
\frac{1}{4} \cdot \left(\frac{n_\mathrm{f}}{n_\mathrm{p}}\right)^u \cdot e^{-\frac{3u^2}{6u^2}} > \\
\frac{1}{8} \cdot \left(\frac{n_\mathrm{f}}{n_\mathrm{p}}\right)^u.
\end{gathered}$$

Therefore, by the averaging argument, there exists a subset S'_f of S_p of size n_f such that

$$\Pr[\mathsf{Read}(\pi) \subseteq S'_\mathrm{f} \wedge \mathsf{Verify}^H(\pi) = 1] > \frac{1}{8} \cdot \left(\frac{n_\mathrm{f}}{n_\mathrm{p}}\right)^u.$$

On the other hand, by soundness from Definition 3)

$$\Pr[\mathsf{Read}(\pi) \subseteq S'_\mathrm{f} \wedge \mathsf{Verify}^H(\pi) = 1] \leq \Pr[\mathrm{SoundExp}(S'_\mathrm{f}) = 1] \leq 2^{-\lambda_\mathrm{sec}}.$$

Thus,

$$\frac{1}{8} \cdot \left(\frac{n_\mathrm{f}}{n_\mathrm{p}}\right)^u < 2^{-\lambda_\mathrm{sec}} \iff$$
$$\left(\frac{n_\mathrm{p}}{n_\mathrm{f}}\right)^u > 2^{\lambda_\mathrm{sec}-3} \iff$$
$$u \log \frac{n_\mathrm{p}}{n_\mathrm{f}} > \lambda_\mathrm{sec} - 3 \iff$$
$$u > \alpha \,,$$

which is a contradiction.

4 ALBAs with Decentralized Prover

In the previous section we assume the ALBA prover has all the set elements at hand. In many applications however, such as threshold signatures, this is not the case. The set elements may be spread across numerous parties who will then jointly compute a proof. A trivial solution is to use a centralized protocol, by designating one of the parties as the lead prover and have all other parties communicate their set elements to that party. However, this incurs a communication cost equal to the size of the set, which we would rather avoid.

In this section we present protocols where the various parties holding set elements start out by performing computations locally and only conditionally communicate their elements to a designated prover or aggregator. Whilst our constructions we present in this section are still unweighted, they can be expanded to integer weights which we present in Section 5.

4.1 Simple Lottery Construction

The simple lottery scheme is parametrized by the expected number of network participants μ. Let H be a random oracle that outputs 1 with probability $p = \frac{\mu}{n_\mathrm{p}}$ and 0 otherwise. Each set element s is sent out to the network if and only if $H(s) = 1$. Now let $r_\mathrm{s}, r_\mathrm{c} > 1$ such that $r_\mathrm{s} r_\mathrm{c} = \frac{n_\mathrm{p}}{n_\mathrm{f}}$ and set $u = r_\mathrm{s} \cdot pn_\mathrm{f}$ (or equivalently $u = \frac{pn_\mathrm{p}}{r_\mathrm{c}}$). The aggregator needs to collect and concatenate u set elements and the verifier accepts if it receives u values that each hash to 1.

Lemma 2. *Assuming*

$$u \geq \frac{\lambda_{sec} \cdot \ln 2}{\ln r_s - 1 + \frac{1}{r_s}} \,,$$

soundness of the scheme is $\leq 2^{-\lambda_{sec}}$.

Proof. Let $S_\mathrm{f} = \{s_1, ..., s_{n_\mathrm{f}}\}$ be malicious prover's set and define $X_i = H(s_i)$. To violate soundness, the malicious prover needs $\sum X_i \geq u = r_\mathrm{s} \cdot pn_\mathrm{f}$, while the expectation $\mathbb{E} \sum X_i = pn_\mathrm{f}$. By Chernoff bound (lemma 5) (with $\delta = r_\mathrm{s} - 1$),

$$\Pr\Big[\sum X_i \geq u\Big] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{pn_\mathrm{f}} = \left(\frac{e^{r_\mathrm{s}-1}}{r_\mathrm{s}^{r_\mathrm{s}}}\right).$$

This is at most $2^{-\lambda_{\text{sec}}}$ if and only if

$$\begin{gathered} pn_{\text{f}}(r_{\text{s}} - 1 - r_{\text{s}} \ln r_{\text{s}}) \leq -\lambda_{\text{sec}} \cdot \ln 2 \iff \\ pn_{\text{f}}(r_{\text{s}} \ln r_{\text{s}} - r_{\text{s}} + 1) \geq \lambda_{\text{sec}} \cdot \ln 2 \iff \\ u\left(\ln r_{\text{s}} - 1 + \frac{1}{r_{\text{s}}} \right) \geq \lambda_{\text{sec}} \cdot \ln 2 \iff \\ u \geq \frac{\lambda_{\text{sec}} \cdot \ln 2}{\ln r_{\text{s}} - 1 + \frac{1}{r_{\text{s}}}} \end{gathered}$$

which is true by our assumption about u.

Lemma 3. *Assuming*

$$u \geq \frac{\lambda_{rel} \cdot \ln 2}{r_c - 1 - \ln r_c},$$

completeness of the above decentralized scheme is $\geq 1 - 2^{-\lambda_{rel}}$.

Proof. Let $S_{\text{p}} = \{s_1, ..., s_{n_{\text{p}}}\}$ be honest prover's set and define $X_i = H(s_i)$. The honest prover fails whenever $\sum X_i < u = \frac{pn_{\text{p}}}{r_{\text{c}}}$, while the expectation $\mathbb{E} \sum X_i = pn_{\text{p}}$. By Chernoff bound (lemma 7) (with $\delta = 1 - \frac{1}{r_{\text{c}}}$),

$$\Pr\left[\sum X_i < u \right] \leq \Pr\left[\sum X_i \leq u \right] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right)^{pn_{\text{p}}} = \left(\frac{e^{\frac{1}{r_{\text{c}}} - 1}}{(\frac{1}{r_{\text{c}}})^{\frac{1}{r_{\text{c}}}}} \right)^{pn_{\text{p}}}.$$

This is at most $2^{-\lambda_{\text{rel}}}$ if and only if

$$\begin{gathered} pn_{\text{p}}\left(\frac{1}{r_{\text{c}}} - 1 - \frac{1}{r_{\text{c}}} \cdot \ln \frac{1}{r_{\text{c}}} \right) \leq -\lambda_{\text{rel}} \cdot \ln 2 \iff \\ pn_{\text{p}}\left(1 - \frac{1}{r_{\text{c}}} - \frac{1}{r_{\text{c}}} \cdot \ln r_{\text{c}} \right) \geq \lambda_{\text{rel}} \cdot \ln 2 \iff \\ u(r_{\text{c}} - 1 - \ln r_{\text{c}}) \geq \lambda_{\text{rel}} \cdot \ln 2 \iff \\ u \geq \frac{\lambda_{\text{rel}} \cdot \ln 2}{r_{\text{c}} - 1 - \ln r_{\text{c}}} \end{gathered}$$

which is true by our assumption about u.

Thus, to minimize u, we need to minimize

$$\max\left\{ \frac{\lambda_{\text{sec}} \cdot \ln 2}{\ln r_{\text{s}} - 1 + \frac{1}{r_{\text{s}}}}, \frac{\lambda_{\text{rel}} \cdot \ln 2}{r_{\text{c}} - 1 - \ln r_{\text{c}}} \right\}.$$

Noting that that the first term is decreasing with respect to r_{s} and the second term is decreasing with respect to r_{c}, the minimum is achieved when the two terms are equal. If $\lambda_{\text{sec}} = \lambda_{\text{rel}} = \lambda$, then setting $r_{\text{c}} = \frac{n_{\text{p}}}{n_{\text{p}} - n_{\text{f}}} \cdot \ln \frac{n_{\text{p}}}{n_{\text{f}}}$ and $r_{\text{s}} = \frac{n_{\text{p}} - n_{\text{f}}}{n_{\text{f}}} \cdot \frac{1}{\ln \frac{n_{\text{p}}}{n_{\text{f}}}}$ gives the smallest u.

We note the interesting fact that choosing r_{s} and r_{c} that minimize u also minimizes μ. Since $\mu = pn_{\mathrm{p}} = ur_{\mathrm{c}}$, we have

$$\mu \geq \max\left\{\frac{\lambda_{\mathrm{sec}} \cdot \ln 2}{\ln r_{\mathrm{s}} - 1 + \frac{1}{r_{\mathrm{s}}}} \cdot r_{\mathrm{c}}, \frac{\lambda_{\mathrm{rel}} \cdot \ln 2}{r_{\mathrm{c}} - 1 - \ln r_{\mathrm{c}}} \cdot r_{\mathrm{c}}\right\}.$$

The first term is decreasing with respect to r_{s} since r_{c} is, and it can be seen that the second term is decreasing with respect to r_{c}. Hence, μ is minimized when the two terms are equal which is the same as the condition for minimizing u.

4.2 Decentralized Telescope

The next logical step to minimize the size of the proof is to run a smarter aggregator, Telescope. As previously, we have parameter μ and select each element to be sent to the network with probability $\frac{\mu}{n_{\mathrm{p}}}$. After receiving enough elements selected by the simple lottery, we run the algorithm from section 3.2. It assumes that the honest number of set elements is large enough, so each element will produce k sub-elements, for an appropriate k, if necessary.

We employ threshold analysis here: calculate the number of set elements selected by the simple lottery such that 1) this number is achievable with probability $1-\frac{1}{4}\cdot 2^{-\lambda_{\mathrm{rel}}}$ and 2) the centralized algorithm will produce a valid certificate with probability $1 - \frac{3}{4} \cdot 2^{-\lambda_{\mathrm{rel}}}$.

For all $1 \leq i \leq n_{\mathrm{p}}$, let X_i be 1 if and only if element s_i is selected and 0 otherwise. Let $X = \sum_{i=1}^{n_{\mathrm{p}}} X_i$; then $\mathbb{E}[X] = \mu$. Assume $\rho \in \mathbb{N}$ satisfies $\Pr[X \geq \rho] \geq 1 - 2^{-\lambda_{\mathrm{rel}}-2}$. Reducing the honest-malicious gap from $\frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}$ to $\frac{\rho}{\frac{\mu}{n_{\mathrm{p}}} \cdot n_{\mathrm{f}}} = \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} \cdot \frac{\rho}{\mu}$ results in increasing the certificate size to

$$\frac{\lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}} + 2) + 1 + \log e - \log\log e}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}}$$

(we have $\lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}} + 2) + 1 + \log e - \log\log e$ instead of $\lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}} + \log 3) + 1 - \log\log e$ in theorem 2 because we instantiate it with $\lambda_{\mathrm{sec}} := \lambda_{\mathrm{sec}} + \log e$ and $\lambda_{\mathrm{rel}} := \lambda_{\mathrm{rel}} + \log \frac{4}{3}$ for technical reasons).

One can think of the gap $\frac{\rho n_{\mathrm{p}}}{\mu n_{\mathrm{f}}}$ as $\frac{(1-\delta)n_{\mathrm{p}}}{n_{\mathrm{f}}}$ if we set $\rho = (1-\delta)\mu$. Note that we only decrease n_{p} in the $\frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}$ gap. n_{f} remains the same since the union-bound argument for soundness still works, but with some modifications. Particularly, it requires μ to be on the order of u^2. If we wanted to decrease μ even further, we could improve the proof below or employ a two-sided threshold analysis as well.

Let $\mathrm{Lottery} : \{0,1\}^* \to \{0,1\}$ be an oracle returning 1 with probability $\frac{\mu}{n_{\mathrm{p}}}$ and assume $H = (H_0, H_1, H_2, \mathrm{Lottery})$ where H_0, H_1, H_2 are as defined in Section 3.2. Also let $A.\mathsf{Prove}^{H,W}$, $A.\mathsf{Verify}^{H,W}$ be as in Section 3.2 and define the following.

```
procedure B.Prove^{H,W}(s)
    if W(s) = 1 ∧ Lottery(s) = 1
    then
        return s;
    else
        return empty string;
procedure B.Aggregate^{H,W}(S)
    return A.Prove^{H,W}(S);
procedure B.Verify^{H,W}(π)
    parse π as (t, s_1, ..., s_u);
    return 1 iff A.Verify(π) = 1∧
    ∀i ∈ [u] : Lottery(s_i) = 1;
```

We now state the main result of this section which follows from the proofs below. Taking $B.\mathsf{Extract}$ as defined later in the section, we have

Theorem 16. *Using parameters from corollary 3,* ($B.\mathsf{Prove}$, $B.\mathsf{Aggregate}$, $B.\mathsf{Verify}$, $B.\mathsf{Extract}$) *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-decentralized NIROPK ALBA scheme with* $B.\mathsf{Extract}$ *being a straight-line extractor.*

Theorem 17. *Assume*

$$k \geq \frac{d^2 \log e}{9\rho(\lambda_{rel}+2)}$$

and instantiate the algorithm in section 3.2 with $d \geq \frac{16u(\lambda_{rel}+2)}{\log e}$, $q := \frac{2(\lambda_{rel}+2)}{d \log e}$, *and* $n_p := k\rho$. *Then completeness is* $\geq 1 - 2^{-\lambda_{rel}}$.

Proof. As assumed above, the simple lottery chooses at least ρ set elements with probability at least $1 - 2^{-\lambda_{\mathrm{rel}}-2}$. Given this event, by theorem 10, the algorithm outputs a valid certificate with probability at least $1 - 2^{-\lambda_{\mathrm{rel}}-\log\frac{4}{3}}$. Therefore, completeness is $\geq 1 - 2^{-\lambda_{\mathrm{rel}}}$.

We now calculate soundness defined as the probability that a valid proof can be constructed using elements S_{f} with $|S_{\mathrm{f}}| = n_{\mathrm{f}}$.

Theorem 18. *Assume*

$$\mu \geq \frac{n_p u^2}{n_f}; \qquad \frac{\rho n_p}{\mu n_f} > 1;$$
$$u \geq \frac{\lambda_{sec} + \log(\lambda_{rel}+2) + 1 + \log e - \log\log e}{\log\frac{n_p}{n_f} + \log\frac{\rho}{\mu}}.$$

Then soundness is $\leq 2^{-\lambda_{sec}}$.

Proof. Denote soundness by S. First we upper bound the number of malicious certificate tuples with exactly l distinct set elements, for all $1 \leq l \leq u$. To do that, we first choose l out of u positions for the distinct set elements, then choose l distinct elements with permutation for the l positions; finally, each of the elements in the l positions have k possible sub-elements, there are kl choices for the other $(u-l)$ positions, and there are d choices for the tuple's integer. Overall, the number of tuples with exactly l distinct elements is at most

$$d \cdot C(u,l) \cdot P(n_{\mathrm{f}}, l) \cdot k^l \cdot (kl)^{u-l}.$$

Then by union bound,

$$S \leq \sum_{l=1}^{u} \left(\left(\frac{\mu}{n_{\mathrm{p}}}\right)^{l} \cdot \left(\frac{1}{k\rho}\right)^{u} \cdot q \cdot d \cdot C(u,l) \cdot P(n_{\mathrm{f}},l) \cdot k^{l} \cdot (kl)^{u-l} \right) =$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \sum_{l=1}^{u} \left(\left(\frac{\mu}{n_{\mathrm{p}}}\right)^{l} \cdot C(u,l) \cdot P(n_{\mathrm{f}},l) \cdot l^{u-l} \right) =$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \sum_{l=1}^{u} \left(\left(\frac{\mu}{n_{\mathrm{p}}}\right)^{l} \cdot \frac{u!}{l!(u-l)!} \cdot \frac{n_{\mathrm{f}}!}{(n_{\mathrm{f}}-l)!} \cdot l^{u-l} \right) \leq$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \sum_{l=1}^{u} \left(\left(\frac{\mu}{n_{\mathrm{p}}}\right)^{l} \cdot \frac{u^{u-l}}{(u-l)!} \cdot n_{\mathrm{f}}^{l} \cdot u^{u-l} \right) =$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \sum_{l=1}^{u} \left(\left(\frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^{l} \cdot \frac{u^{2(u-l)}}{(u-l)!} \right) =$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \left(\frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^{u} \cdot \sum_{l=1}^{u} \left(\left(\frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^{l-u} \cdot \frac{u^{2(u-l)}}{(u-l)!} \right) =$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \left(\frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^{u} \cdot \sum_{l=1}^{u} \left(\left(\frac{u^{2} n_{\mathrm{p}}}{\mu n_{\mathrm{f}}}\right)^{u-l} \cdot \frac{1}{(u-l)!} \right) \leq$$

By our assumption about μ,

$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \left(\frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^{u} \cdot \sum_{l=1}^{u} \frac{1}{(u-l)!} \leq$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \left(\frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^{u} \cdot \sum_{i=0}^{\infty} \frac{1}{i!} =$$
$$\left(\frac{1}{\rho}\right)^{u} \cdot qd \cdot \left(\frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^{u} \cdot e =$$
$$\left(\frac{\mu n_{\mathrm{f}}}{\rho n_{\mathrm{p}}}\right)^{u} \cdot \frac{2(\lambda_{\mathrm{rel}}+2)}{\log e} \cdot e.$$

Then

$$-\log S \geq -\left(u \log \frac{\mu n_{\mathrm{f}}}{\rho n_{\mathrm{p}}} + 1 + \log(\lambda_{\mathrm{rel}}+2) - \log\log e + \log e \right) =$$
$$u \log \frac{\rho n_{\mathrm{p}}}{\mu n_{\mathrm{f}}} - \log(\lambda_{\mathrm{rel}}+2) - 1 - \log e + \log\log e \geq$$
$$\lambda_{\mathrm{sec}}.$$

Now to show the proof of knowledge property, define $\mathsf{Extract}^{H,W,\mathcal{A}}$ similarly to Section 3.1. We have the following theorem whose proof is similar to that of theorem 4.

Theorem 19. *Define parameters as in theorem 18. Then* $\mathsf{Verify}^{H,W}$ *with* $\mathsf{Extract}^{H,W,\mathcal{A}}$ *satisfy the proof of knowledge property of definition 2. Additionally,* $\mathsf{Extract}^{H,W,\mathcal{A}}$ *is straight-line.*

For simplicity, below we continue talking about soundness which ultimately implies the proof of knowledge. Theorem 17 and theorem 18 gives

Corollary 3. *Assume*

$$\mu \geq \frac{n_p u^2}{n_f}; \qquad \frac{\rho n_p}{\mu n_f} > 1; \qquad k \geq \frac{d^2 \log e}{9\rho(\lambda_{rel}+2)};$$
$$u \geq \frac{\lambda_{sec} + \log(\lambda_{rel}+2) + 1 + \log e - \log\log e}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}}$$

and instantiate the algorithm in section 3.2 with $u := u$, $d \geq \frac{16u(\lambda_{rel}+2)}{\log e}$, $q := \frac{2(\lambda_{rel}+2)}{d \log e}$, *and* $n_p := k\rho$. *Then soundness is* $\leq 2^{-\lambda_{sec}}$ *and completeness is* $\geq 1 - 2^{-\lambda_{rel}}$.

Using this, we can see how big μ needs to be if we increase $u \log \frac{n_\mathrm{p}}{n_\mathrm{f}}$ only by some amount C. To calculate a suitable ρ, we just use Chernoff bound. $\Pr[X \leq (1-\delta)\mu] \leq e^{-\frac{\mu\delta^2}{2}}$. Setting this to $2^{-\lambda_\mathrm{rel}-2}$, we get $\delta = \sqrt{\frac{2(\lambda_\mathrm{rel}+2)}{\mu \log e}}$. We now set $\rho = \lceil (1-\delta)\mu \rceil$.

Corollary 4. *Assume*

$$C > 0; \quad u \geq \frac{\lambda_{sec} + \log(\lambda_{rel}+2) + 1 + \log e - \log\log e + C}{\log \frac{n_p}{n_f}}; \quad k \geq \frac{d^2 \log e}{9\rho(\lambda_{rel}+2)}$$
$$\mu \geq \max\left\{\frac{8(\lambda_{rel}+2)}{\log e}, \frac{n_p u^2}{n_f}, \frac{9u^2(\lambda_{rel}+2)\log e}{2C^2}\right\}; \mu > \frac{2(\lambda_{rel}+2)}{\left(1-\frac{n_f}{n_p}\right)^2 \log e};$$

and instantiate the algorithm in section 3.2 just like in corollary 3. Then soundness is $\leq 2^{-\lambda_{sec}}$ *and completeness is* $\geq 1 - 2^{-\lambda_{rel}}$.

Proof. We only need to show that $\frac{\rho n_\mathrm{p}}{\mu n_\mathrm{f}} > 1$ and

$$u \geq \frac{\lambda_\mathrm{sec} + \log(\lambda_\mathrm{rel}+2) + 1 + \log e - \log\log e}{\log \frac{n_\mathrm{p}}{n_\mathrm{f}} + \log \frac{\rho}{\mu}}$$

to apply corollary 3.

The first inequality follows from

$$\begin{gathered}
\frac{(1-\delta)\mu n_{\mathrm{p}}}{\mu n_{\mathrm{f}}} > 1 \Longleftrightarrow \\
\frac{(1-\delta) n_{\mathrm{p}}}{n_{\mathrm{f}}} > 1 \Longleftrightarrow \\
1-\delta > \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}} \Longleftrightarrow \\
\delta < 1 - \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}} \Longleftrightarrow \\
\delta^2 < \left(1 - \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^2 \Longleftrightarrow \\
\frac{2(\lambda_{\mathrm{rel}}+2)}{\mu \log e} < \left(1 - \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^2 \Longleftrightarrow \\
\mu > \frac{2(\lambda_{\mathrm{rel}}+2)}{\left(1-\frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^2 \log e}
\end{gathered}$$

which is true by our assumption about μ.

Now we will show that

$$u \geq \frac{\lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}}+2) + 1 + \log e - \log\log e}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}}.$$

Define $\lambda' = \lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}}+2) + 1 + \log e - \log\log e$. Then this inequality is equivalent to

$$\begin{gathered}
u \geq \frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}} \Longleftarrow \\
\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}} \leq \frac{\lambda'+C}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}} \Longleftarrow \\
\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log(1-\delta)} \leq \frac{\lambda'+C}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}} [\Longleftarrow]
\end{gathered}$$

Since $\delta = \sqrt{\frac{2(\lambda_{\text{rel}}+2)}{\mu \log e}} \leq \sqrt{\frac{2(\lambda_{\text{rel}}+2)}{\frac{8(\lambda_{\text{rel}}+2)}{\log e} \log e}} = \frac{1}{2}$, we know that $1-\delta \geq e^{-\delta-\delta^2} \geq e^{-\frac{3}{2}\delta}$. Therefore,

$$
\begin{aligned}
[\Longleftarrow] \frac{\lambda'}{\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log e^{-\frac{3}{2}\delta}} \leq \frac{\lambda'+C}{\log \frac{n_{\text{p}}}{n_{\text{f}}}} &\iff \\
\frac{\lambda'}{\log \frac{n_{\text{p}}}{n_{\text{f}}} - \frac{3}{2}\delta \log e} \leq \frac{\lambda'+C}{\log \frac{n_{\text{p}}}{n_{\text{f}}}} &\iff \\
\lambda' \log \frac{n_{\text{p}}}{n_{\text{f}}} \leq (\lambda'+C)\left(\log \frac{n_{\text{p}}}{n_{\text{f}}} - \frac{3}{2}\delta \log e \right) &\iff \\
\frac{3}{2}\delta \log e(\lambda'+C) \leq C \log \frac{n_{\text{p}}}{n_{\text{f}}} &\iff \\
\delta \leq \frac{2C \log \frac{n_{\text{p}}}{n_{\text{f}}}}{3(\lambda'+C)\log e} &\iff \\
\delta^2 \leq \left(\frac{2C \log \frac{n_{\text{p}}}{n_{\text{f}}}}{3(\lambda'+C)\log e} \right)^2 &\iff \\
\frac{2(\lambda_{\text{rel}}+2)}{\mu \log e} \leq \left(\frac{2C \log \frac{n_{\text{p}}}{n_{\text{f}}}}{3(\lambda'+C)\log e} \right)^2 &\iff \\
\mu \geq \frac{2(\lambda_{\text{rel}}+2)}{\log e} \left(\frac{3(\lambda'+C)\log e}{2C \log \frac{n_{\text{p}}}{n_{\text{f}}}} \right)^2 &\iff \\
\mu \geq \frac{9(\lambda_{\text{rel}}+2)\log e}{2C^2} \left(\frac{\lambda'+C}{\log \frac{n_{\text{p}}}{n_{\text{f}}}} \right)^2 &\Longleftarrow \\
\mu \geq \frac{9u^2(\lambda_{\text{rel}}+2)\log e}{2C^2} &
\end{aligned}
$$

which is true by our assumption about μ.

Thus, if we let $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$ and let u only be a constant larger than optimal, we have $\mu = O(\lambda^3)$ as well as the time complexity of the centralized algorithm also $O(\lambda^3)$. Moreover, μ is proportional to $\frac{1}{C^2}$. We note, however, that setting $\lambda_{\text{rel}} \coloneqq 1$ and $\lambda_{\text{sec}} \coloneqq \lambda_{\text{sec}} + \log \lambda_{\text{rel}}$ and amplifying the completeness as mentioned in Section 3.2 lets us reduce the expected communication complexity to $O(\lambda^2)$, but it requires some network engineering to avoid redundant communication.

We also present a differnt corollary showing what u needs to be when expressed in terms of μ.

Corollary 5. *Assume*

$$C > 0;$$
$$u \geq \left(1 + \frac{3\sqrt{2\log e} \cdot \sqrt{\lambda_{rel} + 2}}{\sqrt{\mu} \cdot \log \frac{n_p}{n_f}}\right) \cdot \frac{\lambda_{sec} + \log(\lambda_{rel} + 2) + 1 + \log e - \log\log e}{\log \frac{n_p}{n_f}};$$
$$\mu \geq \max\left\{\frac{8(\lambda_{rel} + 2)}{\log e}, \frac{18(\lambda_{rel} + 2)\log e}{\log^2 \frac{n_p}{n_f}}, \frac{n_p u^2}{n_f}\right\}; \mu > \frac{2(\lambda_{rel} + 2)}{\left(1 - \frac{n_f}{n_p}\right)^2 \log e};$$
$$k \geq \frac{d^2 \log e}{9\rho(\lambda_{rel} + 2)}$$

and instantiate the algorithm in section 3.2 just like in corollary 3. Then soundness is $\leq 2^{-\lambda_{sec}}$ *and completeness is* $\geq 1 - 2^{-\lambda_{rel}}$.

Proof. The proof of corollary 4 shows that the assumption $\mu > \frac{2(\lambda_{\mathrm{rel}}+2)}{\left(1-\frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^2 \log e}$ implies $\frac{\rho n_{\mathrm{p}}}{\mu n_{\mathrm{f}}} > 1$. Thus, we only need to prove

$$u \geq \frac{\lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}} + 2) + 1 + \log e - \log\log e}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}}$$

to apply corollary 3.

Define $\lambda' = \lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}} + 2) + 1 + \log e - \log\log e$. This follows from

$$\frac{\lambda_{\mathrm{sec}} + \log(\lambda_{\mathrm{rel}} + 2) + 1 + \log e - \log\log e}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}} =$$
$$\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}} \leq$$
$$\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log(1 - \delta)} \leq$$
$$\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} - \frac{3}{2}\delta \log e} =$$
$$\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}\left(1 - \frac{3\delta \log e}{2 \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}}\right)} [\leq]$$

It is easy to verify that $\frac{1}{1-\epsilon} \leq 1 + 2\epsilon$ for $0 \leq \epsilon \leq \frac{1}{2}$, and since $\frac{3\delta \log e}{2 \log \frac{n_\mathrm{p}}{n_\mathrm{f}}} \leq \frac{1}{2}$,

$$[\leq]\left(1 + \frac{3\delta \log e}{\log \frac{n_\mathrm{p}}{n_\mathrm{f}}}\right) \cdot \frac{\lambda'}{\log \frac{n_\mathrm{p}}{n_\mathrm{f}}} =$$
$$\left(1 + \frac{3 \log e}{\log \frac{n_\mathrm{p}}{n_\mathrm{f}}} \cdot \sqrt{\frac{2(\lambda_\mathrm{rel} + 2)}{\mu \log e}}\right) \cdot \frac{\lambda'}{\log \frac{n_\mathrm{p}}{n_\mathrm{f}}} =$$
$$\left(1 + \frac{3\sqrt{2 \log e} \cdot \sqrt{\lambda_\mathrm{rel} + 2}}{\sqrt{\mu} \cdot \log \frac{n_\mathrm{p}}{n_\mathrm{f}}}\right) \cdot \frac{\lambda'}{\log \frac{n_\mathrm{p}}{n_\mathrm{f}}} \leq$$
$$u.$$

4.3 Optimality of the certificate size - communication tradeoff

We can attempt to find a lower bound for the tradeoff between the certificate size u and μ. For this purpose, we use the following definition.

Definition 4. $(\mathsf{Prove}, \mathsf{Read}, \mathsf{Verify})$ *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f, \mu)$*-lottery based ALBA scheme if and only if*

- Prove^H *is a probabilistic expected polynomial time random oracle access program;*
- Verify^H *is a p.p.t. random oracle access program;*
- Read *is a polynomial time program;*
- *if* L *is a random binary function such that for all* x, $\Pr[L(x) = 1] = \frac{\mu}{n_p}$ *and we define* $Lottery(S) = \{x \in S : L(x) = 1\}$*, then*
  - *completeness: consider the following experiment* $CompExp(S_p)$*:*

    $\pi \leftarrow \mathsf{Prove}^H(Lottery(S_p))$*;*

    output 1 *iff* $\mathsf{Read}(\pi) \subseteq Lottery(S_p)$ *and* $\mathsf{Verify}^H(\pi) = 1$*;*

    we require that for all sets S_p *with size* $\geq n_p$, $\Pr[CompExp(S_p) = 1] \geq 1 - 2^{-\lambda_{rel}}$.
  - *soundness: consider the following experiment* $SoundExp(S_f)$*:*

    output 1 *iff* $\exists \pi, \mathsf{Read}(\pi) \subseteq Lottery(S_f) \wedge \mathsf{Verify}^H(\pi) = 1$*;*

    we require that for all sets S_f *with size* $\leq n_f$, $\Pr[SoundExp(S_f) = 1] \leq 2^{-\lambda_{sec}}$*;*

The following theorem presents our lower bound.

Theorem 20. *Assume* ρ *satisfies* $\Pr\left[B(n_p, \frac{\mu}{n_p}) \leq \rho\right] \geq 2^{-\lambda_{rel}+1}$ *where* $B(n, p)$ *is a binomial random variable with* n *experiments each with probability of success* p*. Also assume*

$$\frac{\rho n_p}{\mu n_f} > 1; \quad \mu \geq \frac{3u^2 n_p \log e}{2n_f}; \quad n_f \geq \rho,$$

let S_p be an arbitrary set of size n_p and let $(\mathsf{Prove}, \mathsf{Read}, \mathsf{Verify})$ *be a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f, \mu)$*-lottery based ALBA scheme such that*

$$\Pr\left[\left|\mathsf{Read}(\mathsf{Prove}^H(\mathit{Lottery}(S_p)))\right| \leq u\right] = 1.$$

Then

$$u > \frac{\lambda_{sec} - 4}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}}.$$

Proof. By completeness, if $\pi \leftarrow \mathsf{Prove}^H(\mathrm{Lottery}(S_\mathrm{p}))$, then

$$\begin{gathered}
2^{-\lambda_\mathrm{rel}} \geq \\
\Pr\left[\neg\big(\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_\mathrm{p}) \wedge \mathsf{Verify}^H(\pi) = 1\big)\right] \geq \\
\Pr\left[\neg\big(\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_\mathrm{p}) \wedge \mathsf{Verify}^H(\pi) = 1\big)\middle|\left|\mathrm{Lottery}(S_\mathrm{p})\right| \leq \rho\right] \times \\
\Pr\left[\left|\mathrm{Lottery}(S_\mathrm{p})\right| \leq \rho\right] \geq \\
\Pr\left[\neg\big(\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_\mathrm{p}) \wedge \mathsf{Verify}^H(\pi) = 1\big)\middle|\left|\mathrm{Lottery}(S_\mathrm{p})\right| \leq \rho\right] \cdot 2^{-\lambda_\mathrm{rel}+1}.
\end{gathered}$$

Therefore,

$$\Pr\left[\neg\big(\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_\mathrm{p}) \wedge \mathsf{Verify}^H(\pi) = 1\big)\middle|\left|\mathrm{Lottery}(S_\mathrm{p})\right| \leq \rho\right] \leq \frac{1}{2}$$

and

$$\Pr\left[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_\mathrm{p}) \wedge \mathsf{Verify}^H(\pi) = 1\middle|\left|\mathrm{Lottery}(S_\mathrm{p})\right| \leq \rho\right] \geq \frac{1}{2}.$$

By the averaging argument, there exists $0 \leq m \leq \rho$ such that

$$\Pr\left[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_\mathrm{p}) \wedge \mathsf{Verify}^H(\pi) = 1\middle|\left|\mathrm{Lottery}(S_\mathrm{p})\right| = m\right] \geq \frac{1}{2}. \quad (5)$$

Now for all $S_\mathrm{f} \subseteq S_\mathrm{p}$ of size n_f, define

procedure $\mathcal{A}^{L,H}_{S_\mathrm{f}}$
- $S \leftarrow \mathrm{Lottery}(S_\mathrm{f})$;
- **if** $m < |S|$ **then**
  - remove $(|S| - m)$ random elements from S;
- **else**
  - add $(m - |S|)$ random elements from $S_\mathrm{f} \setminus S$ to S;
- $\pi \leftarrow \mathsf{Prove}^H(S)$;
- **output** π.

Let S_{f} be a uniformly random subset of S_{p} of size n_{f} and let $\pi \leftarrow \mathcal{A}_{S_{\mathrm{f}}}^{L,H}()$. We now lower bound the following:

$$\begin{gathered}\Pr[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_{\mathrm{f}}) \wedge \mathsf{Verify}^{H}(\pi) = 1] \geq \\ \Pr\left[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_{\mathrm{f}}) \wedge \mathsf{Verify}^{H}(\pi) = 1 \middle| \left|\mathrm{Lottery}(S_{\mathrm{f}})\right| \geq \frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right] \times \\ \Pr\left[\left|\mathrm{Lottery}(S_{\mathrm{f}})\right| \geq \frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right] [>]\end{gathered}$$

It is proven in [GM14] that for all $m \geq 1$ and $p > \frac{1}{m}$, $\Pr[B(m,p) \geq mp] > \frac{1}{4}$. Thus,

$$\begin{gathered}[>]\frac{1}{4} \cdot \Pr\left[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_{\mathrm{f}}) \wedge \mathsf{Verify}^{H}(\pi) = 1 \middle| \left|\mathrm{Lottery}(S_{\mathrm{f}})\right| \geq \frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right] \geq \\ \frac{1}{4} \cdot \Pr\left[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_{\mathrm{f}}) \wedge \mathsf{Verify}^{H}(\pi) = 1 \middle| \right. \\ \left. \mathsf{Read}(\pi) \subseteq S \wedge \mathsf{Verify}^{H}(\pi) = 1 \wedge \left|\mathrm{Lottery}(S_{\mathrm{f}})\right| \geq \frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right] \times \\ \times \Pr\left[\mathsf{Read}(\pi) \subseteq S \wedge \mathsf{Verify}^{H}(\pi) = 1 \middle| \left|\mathrm{Lottery}(S_{\mathrm{f}})\right| \geq \frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right] [\geq]\end{gathered}$$

One can see that in $\mathcal{A}_{S_{\mathrm{f}}}$, independent of $\left|\mathrm{Lottery}(S_{\mathrm{f}})\right|$, S is a uniformly random subset of S_{p} of size m, and using equation 5,

$$\begin{gathered}[\geq]\frac{1}{8} \cdot \Pr\left[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_{\mathrm{f}}) \wedge \mathsf{Verify}^{H}(\pi) = 1 \middle| \right. \\ \left. \mathsf{Read}(\pi) \subseteq S \wedge \mathsf{Verify}^{H}(\pi) = 1 \wedge \left|\mathrm{Lottery}(S_{\mathrm{f}})\right| \geq \frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right] = \\ \frac{1}{8} \cdot \Pr\left[\mathsf{Read}(\pi) \subseteq \mathrm{Lottery}(S_{\mathrm{f}}) \middle| \right. \\ \left. \mathsf{Read}(\pi) \subseteq S \wedge \mathsf{Verify}^{H}(\pi) = 1 \wedge \left|\mathrm{Lottery}(S_{\mathrm{f}})\right| \geq \frac{\mu n_{\mathrm{f}}}{n_{\mathrm{p}}}\right] [\geq]\end{gathered}$$

One can also see that $\text{Lottery}(S_{\text{f}})$ is a uniformly random subset of S of size $|\text{Lottery}(S_{\text{f}})|$. Then,

$$[\geq]\frac{1}{8}\cdot\prod_{i=0}^{u-1}\frac{\left\lceil\frac{\mu n_{\text{f}}}{n_{\text{p}}}\right\rceil-i}{\rho-i}\geq$$
$$\frac{1}{8}\cdot\left(\frac{\frac{\mu n_{\text{f}}}{n_{\text{p}}}-u}{\rho}\right)^{u}=$$
$$\frac{1}{8}\cdot\left(\frac{\frac{\mu n_{\text{f}}}{n_{\text{p}}}}{\rho}\right)^{u}\cdot\left(\frac{\frac{\mu n_{\text{f}}}{n_{\text{p}}}-u}{\frac{\mu n_{\text{f}}}{n_{\text{p}}}}\right)^{u}=$$
$$\frac{1}{8}\cdot\left(\frac{\mu n_{\text{f}}}{\rho n_{\text{p}}}\right)^{u}\cdot\left(1-\frac{un_{\text{p}}}{\mu n_{\text{f}}}\right)^{u}[\geq]$$

Since $\frac{un_{\text{p}}}{\mu n_{\text{f}}}\leq\frac{un_{\text{p}}}{\frac{3u^2n_{\text{p}}\log e}{2n_{\text{f}}}\cdot n_{\text{f}}}=\frac{2}{3u\log e}\leq\frac{2}{3\log e}\leq\frac{1}{2}$ and $1-x\geq e^{-x-x^2}\geq e^{-\frac{3}{2}x}$ for $0\leq x\leq\frac{1}{2}$,

$$[\geq]\frac{1}{8}\cdot\left(\frac{\mu n_{\text{f}}}{\rho n_{\text{p}}}\right)^{u}\cdot\left(e^{-\frac{3un_{\text{p}}}{2\mu n_{\text{f}}}}\right)^{u}\geq$$
$$\frac{1}{8}\cdot\left(\frac{\mu n_{\text{f}}}{\rho n_{\text{p}}}\right)^{u}\cdot\exp\left(-\frac{3u^2n_{\text{p}}}{2\cdot\frac{3u^2n_{\text{p}}\log e}{2n_{\text{f}}}\cdot n_{\text{f}}}\right)=$$
$$\frac{1}{8}\cdot\left(\frac{\mu n_{\text{f}}}{\rho n_{\text{p}}}\right)^{u}\cdot\exp\left(-\frac{1}{\log e}\right)=$$
$$\frac{1}{16}\cdot\left(\frac{\mu n_{\text{f}}}{\rho n_{\text{p}}}\right)^{u}.$$

Hence, by the averaging argument there exists a subset S'_{f} of S_{p} of size n_{f} such that if $\pi'\leftarrow\mathcal{A}^{L,H}_{S'_{\text{f}}}()$ then

$$\Pr[\mathsf{Read}(\pi')\subseteq\text{Lottery}(S'_{\text{f}})\wedge\mathsf{Verify}^{H}(\pi')=1]>\frac{1}{16}\cdot\left(\frac{\mu n_{\text{f}}}{\rho n_{\text{p}}}\right)^{u}.$$

On the other hand, by soundness from definition 4,

$$\Pr[\mathsf{Read}(\pi')\subseteq\text{Lottery}(S'_{\text{f}})\wedge\mathsf{Verify}^{H}(\pi')=1]\leq$$
$$\Pr[\text{SoundExp}(S'_{\text{f}})=1]\leq 2^{-\lambda_{\text{sec}}}.$$

Therefore,

$$\frac{1}{16}\cdot\left(\frac{\mu n_{\mathrm{f}}}{\rho n_{\mathrm{p}}}\right)^{u} < 2^{-\lambda_{\mathrm{sec}}} \iff$$
$$\left(\frac{\rho n_{\mathrm{p}}}{\mu n_{\mathrm{f}}}\right)^{u} > 2^{\lambda_{\mathrm{sec}}-4} \iff$$
$$u\left(\log\frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}+\log\frac{\rho}{\mu}\right) > \lambda_{\mathrm{sec}}-4 \iff$$
$$u > \frac{\lambda_{\mathrm{sec}}-4}{\log\frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}+\log\frac{\rho}{\mu}}.$$

Using this, we can establish a lower bound similar to the upper bound corollary 4.

Corollary 6. *Let $C > 0$, define*

$$\alpha = \frac{\lambda_{sec}-4+C}{\log\frac{n_p}{n_f}}; u = \lfloor\alpha\rfloor$$

and assume

$$\max\left\{\frac{4}{\lambda_{rel}}, \frac{\lambda_{rel}}{(1-\frac{n_f}{n_p})^2}, \frac{3u^2 n_p \log e}{2n_f}\right\} \le \mu \le \min\left\{\frac{\alpha^2\lambda_{rel}\log^2 e}{4C^2}, \frac{(\frac{4}{e})^{\lambda_{rel}}}{4e^{10}}\right\};$$
$$n_f \ge 2\mu.$$

Let S_p be an arbitrary set of size n_p and let $(\mathsf{Prove}, \mathsf{Read}, \mathsf{Verify})$ be a $(\lambda_{sec}, \lambda_{rel}, n_p, n_f, \mu)$-lottery based ALBA scheme. Then

$$\Pr\left[\left|\mathsf{Read}(\mathsf{Prove}^{H}(\mathit{Lottery}(S_p)))\right| > \alpha\right] > 0.$$

Proof. Suppose otherwise, then $\Pr\left[\left|\mathsf{Read}(\mathsf{Prove}^{H}(\mathrm{Lottery}(S_{\mathrm{p}})))\right| \le u\right] = 1$.

Let $\delta = \sqrt{\frac{\lambda_{\mathrm{rel}}}{4\mu}}$ and $\rho = \lfloor(1-\delta)\mu\rfloor$. By lemma 18,

$$\Pr\left[B\left(n_{\mathrm{p}}, \frac{\mu}{n_{\mathrm{p}}}\right) \le \rho\right] = \Pr\left[B\left(n_{\mathrm{p}}, \frac{\mu}{n_{\mathrm{p}}}\right) \le (1-\delta)\mu\right] \ge 2^{-\lambda_{\mathrm{rel}}+1}.$$

In order to use theorem 20, we need to show that $\frac{\rho n_{\mathrm{p}}}{\mu n_{\mathrm{f}}} > 1$. First we show that $(1-\delta)\mu - 1 \ge (1-2\delta)\mu$. This is equivalent to

$$\delta\mu \ge 1 \iff \sqrt{\frac{\lambda_{\mathrm{rel}}}{4\mu}}\cdot\mu \ge 1 \iff \sqrt{\frac{\lambda_{\mathrm{rel}}\cdot\mu}{4}} \ge 1 \iff \mu \ge \frac{4}{\lambda_{\mathrm{rel}}}$$

which is true by our assumption. Then

$$\frac{\rho n_{\mathrm{p}}}{\mu n_{\mathrm{f}}} > \frac{\big((1-\delta)\mu - 1\big)n_{\mathrm{p}}}{\mu n_{\mathrm{f}}} \ge \frac{(1-2\delta)\mu n_{\mathrm{p}}}{\mu n_{\mathrm{f}}} = \frac{(1-2\delta)n_{\mathrm{p}}}{n_{\mathrm{f}}}.$$

This is at least 1 if and only if

$$
\begin{aligned}
1 - 2\delta \geq \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}} &\iff \\
2\delta \leq 1 - \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}} &\iff \\
4\delta^2 \leq \left(1 - \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^2 &\iff \\
\frac{\lambda_{\mathrm{rel}}}{\mu} \leq \left(1 - \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^2 &\iff \\
\mu \geq \frac{\lambda_{\mathrm{rel}}}{\left(1 - \frac{n_{\mathrm{f}}}{n_{\mathrm{p}}}\right)^2}&
\end{aligned}
$$

which is true by our assumption.

By theorem 20, $u > \frac{\lambda_{\mathrm{sec}} - 4}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}}$. We need to prove that $\frac{\lambda_{\mathrm{sec}} - 4}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}} \geq \alpha$. Define $\lambda' = \lambda_{\mathrm{sec}} - 4$. This is equivalent to

$$
\begin{aligned}
\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log \frac{\rho}{\mu}} \geq \frac{\lambda' + C}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}} &\impliedby \\
\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log(1 - \delta)} \geq \frac{\lambda' + C}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}} &\impliedby \\
\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} + \log e^{-\delta}} \geq \frac{\lambda' + C}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}} &\iff \\
\frac{\lambda'}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} - \delta \log e} \geq \frac{\lambda' + C}{\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}} &\iff \\
\lambda' \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} \geq (\lambda' + C)\left(\log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} - \delta \log e\right) &\iff \\
(\lambda' + C)\delta \log e \geq C \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}} &\iff \\
\delta \geq \frac{C \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}}{(\lambda' + C) \log e} &\iff \\
\frac{\lambda_{\mathrm{rel}}}{4\mu} \geq \left(\frac{C \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}}{(\lambda' + C) \log e}\right)^2 &\iff \\
\mu \leq \frac{\lambda_{\mathrm{rel}}}{4} \cdot \left(\frac{(\lambda' + C) \log e}{C \log \frac{n_{\mathrm{p}}}{n_{\mathrm{f}}}}\right)^2 &\iff \\
\mu \leq \frac{\alpha^2 \lambda_{\mathrm{rel}} \log^2 e}{4C^2}&
\end{aligned}
$$

which is true by our assumption.

Hence $u > \alpha$ and we reach a contradiction.

Alternatively, we present a corollary showing a lower bound on the certificate size as of function of μ. It is comparable to corollary 5.

Corollary 7. *Define*

$$\alpha = \left(1 + \frac{\sqrt{\lambda_{rel}} \cdot \log e}{2\sqrt{\mu} \log \frac{n_p}{n_f}}\right) \cdot \frac{\lambda_{sec} - 4}{\log \frac{n_p}{n_f}}; u = \lfloor \alpha \rfloor$$

and assume

$$\max\left\{\frac{4}{\lambda_{rel}}, \frac{\lambda_{rel}}{(1 - \frac{n_f}{n_p})^2}, \frac{3u^2 n_p \log e}{2n_f}\right\} \leq \mu \leq \frac{(\frac{4}{e})^{\lambda_{rel}}}{4e^{10}};$$
$$n_f \geq 2\mu.$$

Let S_p be an arbitrary set of size n_p and let $(\mathsf{Prove}, \mathsf{Read}, \mathsf{Verify})$ *be a $(\lambda_{sec}, \lambda_{rel}, n_p, n_f, \mu)$-lottery based ALBA scheme. Then*

$$\Pr\left[\left|\mathsf{Read}(\mathsf{Prove}^H(\mathit{Lottery}(S_p)))\right| > \alpha\right] > 0.$$

Proof. Suppose otherwise, then $\Pr\left[\left|\mathsf{Read}(\mathsf{Prove}^H(\mathrm{Lottery}(S_\mathrm{p})))\right| \leq u\right] = 1$.

Let $\delta = \sqrt{\frac{\lambda_\mathrm{rel}}{4\mu}}$ and $\rho = \lfloor (1-\delta)\mu \rfloor$. By lemma 18,

$$\Pr\left[B\left(n_\mathrm{p}, \frac{\mu}{n_\mathrm{p}}\right) \leq \rho\right] = \Pr\left[B\left(n_\mathrm{p}, \frac{\mu}{n_\mathrm{p}}\right) \leq (1-\delta)\mu\right] \geq 2^{-\lambda_\mathrm{rel}+1}.$$

In order to use theorem 20, we need to show that $\frac{\rho n_\mathrm{p}}{\mu n_\mathrm{f}} > 1$. The proof of corollary 6 shows how the assumption $\mu \geq \max\left\{\frac{4}{\lambda_\mathrm{rel}}, \frac{\lambda_\mathrm{rel}}{(1-\frac{n_\mathrm{f}}{n_\mathrm{p}})^2}\right\}$ implies it.

By theorem 20,

$$
\begin{gathered}
u > \\
\frac{\lambda_{\text{sec}} - 4}{\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log \frac{\rho}{\mu}} \geq \\
\frac{\lambda_{\text{sec}} - 4}{\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log(1 - \delta)} \geq \\
\frac{\lambda_{\text{sec}} - 4}{\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log e^{-\delta}} = \\
\frac{\lambda_{\text{sec}} - 4}{\log \frac{n_{\text{p}}}{n_{\text{f}}} - \delta \log e} = \\
\frac{\lambda_{\text{sec}} - 4}{\log \frac{n_{\text{p}}}{n_{\text{f}}} \left(1 - \frac{\delta \log e}{\log \frac{n_{\text{p}}}{n_{\text{f}}}}\right)} \geq \\
\left(1 + \frac{\delta \log e}{\log \frac{n_{\text{p}}}{n_{\text{f}}}}\right) \cdot \frac{\lambda_{\text{sec}} - 4}{\log \frac{n_{\text{p}}}{n_{\text{f}}}} = \\
\left(1 + \frac{\sqrt{\lambda_{\text{rel}}} \log e}{2\sqrt{\mu} \log \frac{n_{\text{p}}}{n_{\text{f}}}}\right) \cdot \frac{\lambda_{\text{sec}} - 4}{\log \frac{n_{\text{p}}}{n_{\text{f}}}} = \\
\alpha
\end{gathered}
$$

which is a contradiction.

5 Adding Weights

We will assume, without loss of generality, that the *weight* function outputs integers. A naive way to handle weights other than 0 and 1 is to interpret each set element s as $\mathit{weight}(s)$ elements $(s, 1), \ldots, (s, \mathit{weight}(s))$ and apply schemes designed for the unweighted case to (s, i) pairs. Unfortunately, this approach results in an exponential (in integer size) increase in prover runtime.

However, any lottery-based scheme in which the number of lottery winners is independent of n_{p} (or at most polylogarithmic in n_{p}) is amenable to a more efficient solution (and the Telescope scheme in Section 3 can be turned into a lottery-based scheme first using Section 4.2). We simply view $(s, 1), \ldots, (s, \mathit{weight}(s))$ pairs as $\mathit{weight}(s)$ different lottery participants. For efficiency, instead of having each of them play the lottery individually with probability p, we sample the number of winners from the binomial distribution $\text{Bin}(\mathit{weight}(s), p)$ (similar to the sortition algorithm used in Algorand [GHM+17]). We do so because it does not matter which i values win — what matters is only the number of winners. If the binomial sampling returns k, then $(s, 1), \ldots, (s, k)$ are considered winners. This does not increase the complexity compared to the unweighted lottery-based scheme, except for binomial sampling rather than lottery applied to each s.

6 Replacing the Random Oracle with PRF

In this section we show how to remove the need for the random oracle and instantiate our scheme in the Common Reference String model (or alternatively, the Uniform Random String model). We utilize a PRF for the hash function H with the CRS being a random PRF key (or alternatively, uniformly random bits sufficient to generate one). We note that although the PRF is only secure against computationally bounded distinguishers, our ALBA scheme retains information-theoretical security. Assume (GenKey, F) is a PRF such that for any oracle access program $\mathcal{A}^O$ with running time bounded by T,

$$\left| \Pr\left[\mathcal{A}^H() = 1\right] - \Pr\left[\mathcal{A}^{F(\mathsf{GenKey}(),\cdot)}() = 1\right] \right| \leq \varepsilon_{\mathrm{prf}}(T). \tag{6}$$

We will assume the unweighted case, but the following can extended to support weights as well. Combining the improved Telescope construction from Section 3.2 with the tight bound on the number of accessible vertices (theorem 14) and instantiating the scheme with the standard random oracle (Section B), one can build a Telescope scheme such that for some $B \in O(\lambda^3)$,

- the honest prover's DFS outputs a valid proof after visiting at most B vertices with probability $\geq 1 - 2^{-\lambda}$;
- there exists a valid proof containing elements from S_{f} or the number of accessible vertices exceeds B with probability $\leq 2^{-\lambda}$.

Implement $\mathsf{Prove}^H(S_{\mathrm{p}})$ as the standard DFS that visits at most B vertices and define $\mathsf{Verify}^H(\pi)$ in a natural way.

We show an ALBA scheme under definition 3 where the random oracle is replaced with CRS. To save space, the new definition (6) is omitted here, but can be found in Section C.2. Looking ahead, Section 6.1 shows how to build a knowledge extractor for it. Construct a Telescope scheme according to that definition as follows.

```
procedure R.Prove(crs, S_p)
    π ← Prove^{F(crs,·)}(S_p);
    return π;
procedure R.Verify((crs, π))
    r ← Verify^{F(crs,·)}(π);
    return r;
procedure R.Read(π)
    is defined in a natural way;
procedure R.GenCRS
    k ← GenKey();
    return k;
```

Theorem 21. *R is a $(\lambda'_{sec}, \lambda'_{rel}, n_p, n_f)$-CRS ALBA scheme where $\lambda'_{sec} = \lambda'_{rel} = -\log\left(2^{-\lambda} + \varepsilon_{prf}(c \cdot B)\right)$.*

Proof. Completeness follows from the fact that Prove's running time is bounded by $c \cdot B$ steps and that $\mathsf{Prove}^H(S_{\mathrm{p}})$, when instantiated with the random oracle H, finds a valid proof with probability $\geq 1 - 2^{-\lambda}$. Acting as a PRF distinguisher, we conclude that $\mathsf{Prove}^{F(\mathsf{GenKey}(),\cdot)}$ outputs a valid proof with probability $\geq 1 - 2^{-\lambda} - \varepsilon_{\mathrm{prf}}(c \cdot B)$.

To prove soundness, we can observe whether a DFS on set S_f finds a valid proof or does not terminate after visiting B vertices. In the random oracle case, one or both happen with probability $\leq 2^{-\lambda}$, so in the PRF case it is $\leq 2^{-\lambda} + \varepsilon_{\mathrm{prf}}(c \cdot B)$. But the probability that there *exists* a valid proof in the PRF case cannot be larger.

We present the full version of the proof in Section C.2.

6.1 Knowledge Extraction for definition 6 / 3

In this section we show how to generically convert an ALBA scheme under definition 6 to a proof of knowledge scheme as defined below. We still assume the unweighted scenario: $W : \{0,1\}^* \to \{0,1\}$ but the following can be generalized to add weights. Sometimes it will be convenient to treat W as a set: $\{s : W(s) = 1\}$.

Definition 5. $(\mathsf{Prove}, \mathsf{Verify}, \mathsf{Extract}, \mathsf{GenCRS})$ *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-CRS proof of knowledge ALBA scheme if and only if*

- Prove *is a probabilistic expected polynomial time program;*
- Verify^W *is a polynomial time program that has access to a weight oracle* W*;*
- $\mathsf{Extract}^W$ *is a probabilistic program having access to a weight oracle* W*;*
- GenCRS *is a p.p.t. program;*
- *completeness: consider the following experiment* $CompExp(W, S_p)$*:*

  $crs \leftarrow \mathsf{GenCRS}()$*;*
  $\pi \leftarrow \mathsf{Prove}(crs, S_p)$*;*
  $r \leftarrow \mathsf{Verify}^W(crs, \pi)$*;*
  return r*;*

  we require that for all weight oracles W *and all sets* $S_p \subseteq W$ *with* $|S_p| \geq n_p$*,* $\Pr[CompExp(W, S_p) = 1] \geq 1 - 2^{-\lambda_{rel}}$*;*
- *proof of knowledge: consider the following experiment* $SoundExp(\mathcal{A}^W, W)$*:*

  $crs \leftarrow \mathsf{GenCRS}()$*;*
  $\pi \leftarrow \mathcal{A}^W(crs)$*;*
  $r \leftarrow \mathsf{Verify}^W(crs, \pi)$*;*
  return r*;*

  we require that for all weight oracles W *and all probabilistic oracle access programs* $\mathcal{A}^W$*, if* $\mathcal{A}$ *runs in time* T *and* $\epsilon = \Pr[SoundExp(\mathcal{A}^W, W) = 1] - 2^{-\lambda_{sec}} > 0$*, then* $S_f \leftarrow \mathsf{Extract}^W(\mathcal{A})$ *runs in expected time* $poly\left(\frac{n_f T}{\epsilon}\right)$ *and* $\Pr\left[S_f \subseteq W \wedge |S_f| > n_f\right] = 1$*.*

Now let $X = (X.\mathsf{Prove}, X.\mathsf{Read}, X.\mathsf{Verify}, X.\mathsf{GenCRS})$ be a $(\lambda_{\mathrm{sec}}, \lambda_{\mathrm{rel}}, n_{\mathrm{p}}, n_{\mathrm{f}})$-CRS ALBA scheme (as in definition 6) and define $Y = (Y.\mathsf{Prove}, Y.\mathsf{Verify}, Y.\mathsf{Extract}, Y.\mathsf{GenCRS})$ as follows.

```
procedure Y.GenCRS
    return X.GenCRS();
procedure Y.Prove(crs, S_p)
    return X.Prove(crs, S_p);
procedure Y.Verify^W(crs, π)
    S := X.Read(π);
    return 1 iff S ⊆ W ∧
    X.Verify(crs, π) = 1;
```

```
procedure Y.Extract^W(A)
    S_f := ∅;
    while |S_f| ≤ n_f do
        crs ← X.GenCRS();
        π ← A^W(crs);
        S := X.Read(π);
        S_f := S_f ∪ (S ∩ W);
    return S_f;
```

Theorem 22. *Y is $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$-proof of knowledge ALBA scheme.*

Proof. Clearly $Y.\mathsf{Prove}$ and $Y.\mathsf{Verify}$ are polynomial time. It is also easy to see that Y satisfies the completeness property. We are left to prove the proof of knowledge property.

First, notice that $Y.\mathsf{Extract}$ can only output a set S_f such that $S_\mathrm{f} \subseteq W$ and $|S_\mathrm{f}| > n_\mathrm{f}$. Now examine a single loop iteration in $Y.\mathsf{Extract}$. We know that $\epsilon = \Pr\left[Y.\mathsf{Verify}^W(\mathrm{crs}, \pi) = 1\right] - 2^{-\lambda_\mathrm{sec}} > 0$ and $Y.\mathsf{Verify}^W(\mathrm{crs}, \pi) = 1$ implies that $S \subseteq W$ and $X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1$. So,

$$2^{-\lambda_\mathrm{sec}} + \epsilon = \Pr[Y.\mathsf{Verify}^W(\mathrm{crs}, \pi) = 1] \leq \Pr[S \subseteq W \wedge X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1].$$

At the same time, since $|S_\mathrm{f}| \leq n_\mathrm{f}$, by the soundness of X (considering the experiment $\mathrm{SoundExp}(S_\mathrm{f})$ from Definition 6), $\Pr[S \subseteq S_\mathrm{f} \wedge X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1] \leq 2^{-\lambda_\mathrm{sec}}$. Therefore,

$$\begin{gathered}
\epsilon = (2^{-\lambda_\mathrm{sec}} + \epsilon) - 2^{-\lambda_\mathrm{sec}} \leq \\
\Pr[S \subseteq W \wedge X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1] - \Pr[S \subseteq S_\mathrm{f} \wedge X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1] \leq \\
\Pr[(S \subseteq W \wedge X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1) \wedge \neg(S \subseteq S_\mathrm{f} \wedge X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1)] = \\
\Pr[S \subseteq W \wedge S \not\subseteq S_\mathrm{f} \wedge X.\mathsf{Verify}(\mathrm{crs}, \pi) = 1] \leq \\
\Pr[S \subseteq W \wedge S \not\subseteq S_\mathrm{f}] \leq \\
\Pr[\exists x \in (S \cap W) \setminus S_\mathrm{f}].
\end{gathered}$$

So, a single iteration of the loop adds at least one new element of W to S_f with probability at least ϵ. Therefore, in expectation, the loop runs for at most $(n_\mathrm{f} + 1) \cdot \frac{1}{\epsilon}$ iterations. Then it is easy to see that $Y.\mathsf{Extract}$ runs in expected time $\mathrm{poly}\left(\frac{n_\mathrm{f} T}{\epsilon}\right)$.

7 Concrete Parameters

In Figure 1 we compare our constructions with existing ALBA protocols such as Compact Certificates [MRV+21] and the Goldwasser-Sipser [GS86]scheme. Our analysis of the simple lottery scheme of section 4.1 is also applicable to Mithril [CK21] as the combinatorics are very similar. For Compact Certificates we note that their soundness is computational as opposed to ours which is information-theoretic for non-adaptive adversaries. In the interest of comparison, we also

n_p/n_f	60/40		66/33		80/20	
ALBA Protocol	Size	Comms	Size	Comms	Size	Comms
GS [GS86]	82944σ		16384σ		3237σ	
C. Cert. [MRV+21] (2^{32})	$274\sigma + 274\eta$		$160\sigma + 160\eta$		$80\sigma + 80\eta$	
C. Cert. [MRV+21] (2^{64})	$330\sigma + 330\eta$		$192\sigma + 192\eta$		$96\sigma + 96\eta$	
Telescope, no weights (Sect. 3)	232σ		136σ		68σ	
Telescope, weights (Sect. 4.2,5)	237σ		139σ		70σ	
Simple Lottery (Sect. 4.1)	4328σ	5264σ	1488σ	2062σ	380σ	702σ
Simple Lottery ($\lambda_{rel} = 64$)	3226σ	3925σ	1128σ	1564σ	298σ	551σ
Dec. Telescope (Sect. 4.2)	262σ	114264σ	151σ	49929σ	74σ	23104σ

Fig. 1. Certificate sizes and expected communication cost, expressed in revealed/sent set elements (σ) and revealed committed set elements (η). In most applications we expect $\eta < \sigma$ but within the same order of magnitude. The parameters $\lambda_{sec}, \lambda_{rel}$ are set to 128 unless otherwise indicated.

describe the cases of adversaries issuing 2^{32} and 2^{64} queries. We consider communication costs only where they are meaningful, i.e. in decentralized schemes. We note that these costs may be significantly lower in the case of weighted sets where the same element may appear multiple times with different indexes. For compact certificates, we derive values using the formula from [MRV+21]. For the simple lottery we use the bounds of Section 4.1, for Goldwasser-Sipser we use the analysis of Theorem 24 in the appendix, and for Telescope and Decentralized Telescope we use the bounds from Corollaries 2 and 4. For the weighted Telescope scheme we apply the transformation of Section 5 to the decentralized lottery and choose to parametrize for the minimum proof size in Corollary 4.

Acknowledgements

This material is based upon work supported in part by a gift from Input Output - IOG and by DARPA under Agreements No. HR00112020021 and HR00112020023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

References

Ash90. Robert B Ash. *Information theory*. 1990.

Bab85. László Babai. Trading group theory for randomness. In *17th ACM STOC*, pages 421–429. ACM Press, May 1985.

BCS16. Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 31–60. Springer, Heidelberg, October / November 2016.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

Can00. Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, January 2000.

CDMP05. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, Heidelberg, August 2005.

CGKS22. Matteo Campanelli, Chaya Ganesh, Hamidreza Khoshakhlagh, and Janno Siim. Impossibilities in succinct arguments: Black-box extraction and more. Cryptology ePrint Archive, Report 2022/638, 2022. `https://eprint.iacr.org/2022/638`.

CK21. Pyrros Chaidos and Aggelos Kiayias. Mithril: Stake-based threshold multisignatures. Cryptology ePrint Archive, Report 2021/916, 2021. `https://eprint.iacr.org/2021/916`.

CW79. Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

DCX+23. Sourav Das, Philippe Camacho, Zhuolun Xiang, Javier Nieto, Benedikt Bunz, and Ling Ren. Threshold signatures from inner product argument: Succinct, weighted, and multi-threshold. Cryptology ePrint Archive, Paper 2023/598, 2023. `https://eprint.iacr.org/2023/598`.

Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, Heidelberg, August 2005.

GHM+17. Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*, pages 51–68, 2017.

GJM+23. Sanjam Garg, Abhishek Jain, Pratyay Mukherjee, Rohit Sinha, Mingyuan Wang, and Yinuo Zhang. hints: Threshold signatures with silent setup. Cryptology ePrint Archive, Paper 2023/567, 2023. `https://eprint.iacr.org/2023/567`.

GKO+23. Chaya Ganesh, Yashvanth Kondi, Claudio Orlandi, Mahak Pancholi, Akira Takahashi, and Daniel Tschudi. Witness-succinct universally-composable SNARKs. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 315–346. Springer, Heidelberg, April 2023.

GM14. Spencer Greenberg and Mehryar Mohri. Tight lower bound on the probability of a binomial exceeding its expectation. *Statistics and Probability Letters*, 86:91–98, 2014.

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016.

GS86. Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th ACM STOC*, pages 59–68. ACM Press, May 1986.

GWC19. Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. `https://eprint.iacr.org/2019/953`.

MRV+21. Silvio Micali, Leonid Reyzin, Georgios Vlachos, Riad S. Wahby, and Nickolai Zeldovich. Compact certificates of collective knowledge. In *2021 IEEE Symposium on Security and Privacy*, pages 626–641. IEEE Computer Society Press, May 2021.

MU05. Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

Pas03. Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 316–337. Springer, Heidelberg, August 2003.

Sip83. Michael Sipser. A complexity theoretic approach to randomness. In *15th ACM STOC*, pages 330–335. ACM Press, April 1983.

A Goldwasser-Sipser Protocol

Consider $\mathcal{H}$ a family of pairwise independent hash functions over $\{0,1\}^\ell$.

Let S be the subset of interest with $|S| = N$. Honest participants have at least n_p values. Adversary has at most n_f values.

The core step of the GS protocol works like that

- The verifier sends random $h \in \mathcal{H}, y \in \{0,1\}^\ell$ to the prover.
- The prover responds with x.
- The verifier accepts provided that $x \in S$ and $h(x) = y$.

Theorem 23. *Let $\gamma \in (0,1)$. For the honest participants, it holds that they can convince the verifier with probability $(1-\gamma)n_p 2^{-\ell}$, provided that $\ell \geq \log(n_p/2\gamma)$. The adversary can convince the verifier with probability at most $2^{-\ell} n_f$.*

Proof. Consider the probability that the prover is capable of finding a suitable x that convinces the verifier in the above interactive proof.

For an adversarial prover, we have that by the union bound the probability they convince the verifier is at most $n_\mathrm{f} 2^{-\ell}$.

For the honest participants, the probability they convince the verifier is at least

$$n_{\mathrm{p}}2^{-\ell} - \sum_{x,x'} \mathsf{Pr}[h(x) = y \wedge h(x') = y] =$$
$$n_{\mathrm{p}}2^{-\ell} - \binom{n_{\mathrm{p}}}{2}2^{-2\ell} \geq$$
$$n_{\mathrm{p}}2^{-\ell} - (n_{\mathrm{p}}2^{-\ell})^2/2$$

where in the penultimate inequality we use pairwise independence. The latter inequality is at least $n_{\mathrm{p}}2^{-\ell}(1-\gamma)$ due to $n_{\mathrm{p}}2^{-\ell} \leq 2\gamma$.

The GS protocol repeats the core step u times. The verifier in the end accepts provided that T core steps are valid.

Theorem 24. *Suppose we want to achieve error $\lambda_{rel}, \lambda_{sec}$ for completeness and soundness respectively with the GS protocol. Then it is sufficient to choose $u \geq 8\max\{\lambda_{sec}, \lambda_{rel}\}x^4(x-1)^{-4}$ for $x = n_p/n_f$.*

Proof. Let $\gamma \in (0, 1 - n_{\mathrm{f}}/n_{\mathrm{p}})$ and $\ell = \log(n_{\mathrm{p}}/2\gamma)$. The expected number of adversarial successes is $\mu_{\mathrm{f}} = 2^{-\ell}n_{\mathrm{f}} = 2\gamma(n_{\mathrm{f}}/n_{\mathrm{p}})u$. Similarly the expected number of honest party successes is $\mu_{\mathrm{p}} = 2^{-\ell}(1-\gamma)n_{\mathrm{p}} = 2\gamma(1-\gamma)u$. We set a threshold $T = \gamma u(1-\gamma-n_{\mathrm{f}}/n_{\mathrm{p}})$. Let $t = \gamma(1-\gamma-n_{\mathrm{f}}/n_{\mathrm{p}})u$. Observe that $\mu_{\mathrm{f}}+t = T = \mu_{\mathrm{p}}-t$. It follows by the Hoeffding bound that: (1) the probability that the adversarial parties reach $T = \mu_{\mathrm{f}} + t$ successes is at most $\exp(-2t^2/u)$, (2) the probability that the honest parties have T successes or less is $\exp(-2t^2/u)$.

We require that $\exp(-2t^2/u) \leq 2^{-\lambda_{\mathrm{sec}}}$ and $\exp(-2t^2/u) \leq 2^{-\lambda_{\mathrm{rel}}}$ Given that $2t^2/u = \gamma^2(1-\gamma-n_{\mathrm{f}}/n_{\mathrm{p}})^2u$ we obtain that it should hold

$$u \geq \gamma^{-2}(1-\gamma-n_{\mathrm{f}}/n_{\mathrm{p}})^{-2}\max\{\lambda_{\mathrm{sec}}, \lambda_{\mathrm{rel}}\}/2$$

We can set now $\gamma = \delta(1 - n_{\mathrm{f}}/n_{\mathrm{p}})$ for some $\delta \in (0,1)$ and we obtain that $u \geq \delta^{-2}(1-\delta)^{-2}x^4/(x-1)^4\min\{\lambda_{\mathrm{sec}}, \lambda_{\mathrm{rel}}\}/2$. The statement of the theorem follows for $\delta = 1/2$.

B Implementing H_0, H_1, and H_2 with a Binary Random Oracle

In this section we address how H_0, H_1, and H_2 used in the Telescope construction (Section 3) are implemented from a single random oracle H that outputs binary strings. We know how collect enough bits from H, using the standard techniques for domain separation of inputs to ensure that domains of H corresponding to inputs of H_0, H_1, and H_2 don't overlap, and using counters as necessary to collect more bits if the output of H is short.

H_0 and H_1 need to output a uniformly distributed integer in $[n_{\mathrm{p}}]$ (or 1 with probability $1/[n_{\mathrm{p}}]$, which can be handled by outputting a random integer and checking if it is 0). If n_{p} is a power of 2, we are done. Else, set a failure bound

$\varepsilon_{\text{fail}}$, set $k = \lceil \log_2(n_{\text{p}}/\varepsilon_{\text{fail}}) \rceil$, and set $d = \lfloor 2^k/n_{\text{p}} \rfloor$. Use H to produce a k-bit string, interpret it as an integer $i \in [0, 2^k - 1]$, fail if $i \geq dn_{\text{p}}$, and output $i \bmod n_{\text{p}}$ otherwise. (Naturally, only the honest prover and verifier will actually fail; dishonest parties can do whatever they want.)

H_2 needs to output 1 with probability q. We will implement H_2 by finding a rational approximation x/y to q where y is a power of 2 and $0 \leq q - (x/y) < \varepsilon_{\text{fail}}$; we will get $i \in [0, y-1]$ out of H and output 1 if $i < x$. This will increase the probability of output 0 by at most $\varepsilon_{\text{fail}}$.

The probability of failure for a single oracle query to H_0 or H_1 is less than $n_{\text{p}}/2^k \leq \varepsilon_{\text{fail}}$. Conditioned on not failing, the distributions of H_0 and H_1 are perfectly accurate, which is important for our soundness / extractability arguments, as we have no bound on the number of adversarial queries to its oracles. (An approximate distribution would not work here.) The value of q simply becomes slightly lower, by at most $\varepsilon_{\text{fail}}$. Extractability works the same way as before, because queries to H_0, H_1, or H_2 are now replaced with queries to H, but the extractor can read those equally well. The facts that queries can fail and that q is slightly lower reduce the probability of adversarial success, which marginally improves the bounds in Theorems 2, 9, and 18 without changing anything else in the extractability proof.

The only effect is on reliability, which gets reduced by $\varepsilon_{\text{fail}} \cdot q_{\text{ro}}$, where q_{ro} is the number of random oracles queries made by the honest prover. Given tight bounds on the prover running time in Section 3, which are guaranteed with overwhelming probability, we can bound this loss by setting $\varepsilon_{\text{fail}}$ high enough.

C Additional Material

C.1 Improved completeness for Section 3.1

Theorem 25. *Assume* $0 \leq q \leq 1$ *and*

$$d \geq \frac{\lambda_{rel}}{\log e} \left(\frac{1}{q} + \frac{u + \ln u}{2} \right)$$

Then completeness is $\geq 1 - 2^{-\lambda_{rel}}$*, and the probability that there exists a valid proof with a particular integer* t *is at least*

$$\left(\frac{1}{q} + \frac{u + 1 + \ln u}{2} \right)^{-1}.$$

Proof. Completeness can be described using the following recursive formula. For $0 \leq k \leq u$, let $f(k)$ be the probability that when fixing a prefix of an integer in $[d]$ and $u-k$ elements $t, s_1, ..., s_{u-k}$, there is no suffix of honest player's elements that works, meaning there is no $s_{u-k+1}, ..., s_u$ such that for all $u-k+1 \leq i \leq u$, $H_1(t, s_1, ..., s_i) = 1$, and $H_2(t, s_1, ..., s_u) = 1$. Then

- $f(0) = 1 - q$;
- for $0 \leq k < u$, $f(k+1) = \left((1 - \frac{1}{n_{\text{p}}}) + \frac{1}{n_{\text{p}}} \cdot f(k) \right)^{n_{\text{p}}}$;

– the probability that the algorithm fails in the honest case is $\big(f(u)\big)^d$.

This recursive formula can be approximated:

$$f(k+1) = \left(1 + \frac{1}{n_{\mathrm{p}}}(f(k)-1)\right)^{n_{\mathrm{p}}} \leq \left(e^{\frac{1}{n_{\mathrm{p}}}(f(k)-1)}\right)^{n_{\mathrm{p}}} = e^{f(k)-1}.$$

We are thus interested in the sequence $\{x_i\}_{i\geq 0}$, where $x_0 = f(0) = 1-q$ and $x_{k+1} = e^{x_k - 1}$. By induction $f(k) \leq x_k$, because $f(i+1) \leq e^{f(i)-1} \leq e^{x_i - 1} = x_{i+1}$.

Claim. For $k \geq 1$,

$$-\ln x_k = 1 - x_{k-1} \geq \left(\frac{1}{q} + \frac{k + \ln(k-1)}{2}\right)^{-1}.$$

Proof. Let $z_k = -\ln x_k = 1 - x_{k-1}$ and note that $z_1 = q$. Then

$$z_{k+1} = 1 - x_k = 1 - e^{-z_k} \geq 1 - \left(1 - z_k + \frac{z_k^2}{2}\right) = z_k - \frac{z_k^2}{2}.$$

Let $t_1 = q$ and $t_{k+1} = t_k - t_k^2/2$. By induction, $z_k \geq t_k$, because $z_{i+1} = z_i - z_i^2/2 \geq t_i - t_i^2/2 = t_{i+1}$.

Let $y_k = \frac{1}{t_k}$. Then

$$y_{k+1} = \left(\frac{1}{y_k} - \frac{1}{2y_k^2}\right)^{-1} = \frac{2y_k^2}{2y_k - 1} = y_k + \frac{1}{2} + \frac{1}{4y_k - 2},$$

and, by induction,

$$y_{k+1} = y_1 + \frac{k}{2} + \sum_{i=1}^{k} \frac{1}{4y_i - 2}.$$

Since $y_i \geq y_1 + (i-1)/2$, we have $4y_i - 2 \geq 4y_1 + 2(i-1) - 2 \geq 2i$ because $y_1 = 1/q \geq 1$. We thus have

$$y_{k+1} \leq \frac{1}{q} + \frac{k}{2} + \sum_{i=1}^{k} \frac{1}{2i} \leq \frac{1}{q} + \frac{k}{2} + \frac{1}{2}\left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{k}\right) \leq \frac{1}{q} + \frac{k + 1 + \ln k}{2}.$$

Recalling that $-\ln x_k = z_k \geq t_k = (y_k)^{-1}$ concludes the proof of the claim.

Therefore, the probability that the honest prover succeeds for a single choice of integer t is at least $1 - x_u$, which by the above claim is at least

$$\left(\frac{1}{q} + \frac{u + 1 + \ln u}{2}\right)^{-1}$$

which means the expected number of attempts for different integers t is at most $\frac{1}{q} + \frac{u+1+\ln u}{2}$.

The probability that the prover fails after d attempts is $f(u)^d \leq x_u^d = \exp(d \ln x_u) \leq \exp(-\lambda_{\mathrm{rel}} / \log e) = 2^{-\lambda_{\mathrm{rel}}}$, by the above claim and the definition of d.

For the smallest running time, choose $q = 1$. Choosing a smaller q increases the running time but slightly decreases u, because $\log(qd)$ shrinks. Using the above and theorem 2, we can make the following choice:

Corollary 8. *Let*

$$u \geq \frac{\lambda_{sec} + \log \lambda_{rel} + 1 - \log \log e}{\log \frac{n_p}{n_f}}; d \geq \frac{(u + \ln u)\lambda_{rel}}{\log e}; q = \frac{2\lambda_{rel}}{d \log e}.$$

Then soundness is $\leq 2^{-\lambda_{sec}}$ *and completeness is* $\geq 1 - 2^{-\lambda_{rel}}$.

C.2 Replacing the Random Oracle with PRF

Definition 6. $(\mathsf{Prove}, \mathsf{Read}, \mathsf{Verify}, \mathsf{GenCRS})$ *is a* $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$*-CRS ALBA scheme if and only if*

- Prove *is a probabilistic expected polynomial time program;*
- Verify *is polynomial time program;*
- Read *is a polynomial time program;*
- GenCRS *is p.p.t. program;*
- *completeness: consider the following experiment* $CompExp(S_p)$*:*

  $crs \leftarrow \mathsf{GenCRS}()$*;*
  $\pi \leftarrow \mathsf{Prove}(crs, S_p)$*;*
  output 1 *iff* $\mathsf{Read}(\pi) \subseteq S_p$ *and* $\mathsf{Verify}(crs, \pi) = 1$*;*

  we require that for all sets S_p *with size* $\geq n_p$, $\Pr[CompExp(S_p) = 1] \geq 1 - 2^{-\lambda_{rel}}$.
- *soundness: consider the following experiment* $SoundExp(S_f)$*:*

  $crs \leftarrow \mathsf{GenCRS}()$*;*
  output 1 *iff* $\exists \pi, \mathsf{Read}(\pi) \subseteq S_f \wedge \mathsf{Verify}(crs, \pi) = 1$*;*

  we require that for all sets S_f *with size* $\leq n_f$, $\Pr[SoundExp(S_f) = 1] \leq 2^{-\lambda_{sec}}$*;*

Theorem 26. *Take any set* S_p *with* $|S_p| \geq n_p$*. Then the construction* R *satisfies* $\Pr[CompExp(S_p) = 1] \geq 1 - 2^{-\lambda} - \varepsilon_{prf}(c \cdot B)$.

Proof. Define

procedure $\mathcal{A}^O$
  $\pi \leftarrow \mathsf{Prove}^O(S_{\mathrm{p}})$;
  $r \leftarrow \mathsf{Verify}^O(\pi)$;
  return r.

By the assumption about our Telescope construction, $\Pr[\mathcal{A}^H() = 1] \geq 1 - 2^{-\lambda}$, and by equation 6,

$$\Pr\left[\mathcal{A}^{F(\mathsf{GenKey}(),\cdot)}() = 1\right] \geq 1 - 2^{-\lambda} - \varepsilon_{\mathrm{prf}}(c \cdot B).$$

But

$$\Pr\left[\mathcal{A}^{F(\mathsf{GenKey}(),\cdot)}() = 1\right] = \Pr[\mathrm{CompExp}(S_{\mathrm{p}}) = 1].$$

Theorem 27. *Let S_f be any set with $|S_f| \leq n_f$. Then the construction R satisfies $\Pr[\mathit{SoundExp}(S_f) = 1] \leq 2^{-\lambda} + \varepsilon_{prf}(c \cdot B)$.*

Proof. Define $\mathcal{A}^O$ as follows: run the standard Telescope DFS on set S_{f}; if we find a proof π that passes $\mathsf{Verify}^O(\pi)$ or the DFS does not terminate after visiting B vertices, then output 1; otherwise output 0.

By the assumption about our Telescope construction, $\Pr[\mathcal{A}^H() = 1] \leq 2^{-\lambda}$, and by equation 6,

$$\Pr\left[\mathcal{A}^{F(\mathsf{GenKey}(),\cdot)}() = 1\right] \leq 2^{-\lambda} + \varepsilon_{\mathrm{prf}}(c \cdot B).$$

But since $\mathrm{SoundExp}(S_{\mathrm{f}}) = 1$ implies $\mathcal{A}^{F(\mathsf{GenKey}(),\cdot)}() = 1$,

$$\Pr[\mathrm{SoundExp}(S_{\mathrm{f}}) = 1] \leq \Pr\left[\mathcal{A}^{F(\mathsf{GenKey}(),\cdot)}() = 1\right].$$

D Additional Lemmas

Lemma 4.

$$\sum_{i=0}^{\infty} \frac{1}{i!} = e; \sum_{i=0}^{\infty} \frac{i}{i!} = e; \sum_{i=0}^{\infty} \frac{i^2}{i!} = 2e$$

Proof. It is known that for any $x \in$, $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$. From this, the first equality follows.

We prove the second equality:

$$\sum_{i=0}^{\infty} \frac{i}{i!} = \sum_{i=1}^{\infty} \frac{i}{i!} = \sum_{i=1}^{\infty} \frac{1}{(i-1)!} = \sum_{i=0}^{\infty} \frac{1}{i!} = e.$$

We prove the third equality:

$$\sum_{i=0}^{\infty} \frac{i^2}{i!} = \sum_{i=1}^{\infty} \frac{i^2}{i!} = \sum_{i=1}^{\infty} \frac{i}{(i-1)!} = \sum_{i=0}^{\infty} \frac{i+1}{i!} = \sum_{i=0}^{\infty} \frac{i}{i!} + \sum_{i=0}^{\infty} \frac{1}{i!} = 2e.$$

D.1 Chernoff Bounds

Below let $X_1, ..., X_n$ be independent Bernoulli random variables, define $X = X_1 + ... + X_n$ and $\mu = \mathbb{E}[X]$.

Lemma 5 (Upper tail). *For any $\delta > 0$,*

$$\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mu}.$$

Lemma 6 (Upper tail, simpler). *For any $\delta \in (0, 1]$,*

$$\Pr[X \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}.$$

Lemma 7 (Lower tail). *For any $\delta \in (0, 1)$,*

$$\Pr[X \leq (1-\delta)\mu] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right)^{\mu}.$$

Lemma 8 (Lower tail, simpler). *For any $\delta \in (0, 1)$,*

$$\Pr[X \leq (1-\delta)\mu] \leq e^{-\mu\delta^2/2}.$$

D.2 Lemmas for Section 3.1

Lemma 9. *Let $t > 0$ and define the sequence $\{x_k\}$ as follows: let $x_0 = 1$ and for $k \geq 0$, let*

$$x_{k+1} = \left(\frac{1}{n}x_k e^t + 1 - \frac{1}{n}\right)^{n_p}.$$

Then $\mathbb{E}[e^{tZ}] = x_u^d$.

Proof. For $1 \leq j \leq d$, $1 \leq i \leq u$, $s_1, ..., s_i \in S_{\mathrm{p}}$ and $1 \leq k \leq i$, let the indicator random variable

$$I_{j,s_1,...,s_i,k} = \begin{cases} 1 & \text{if for all } k \leq r \leq i,\ H_1(j, s_1, ..., s_r) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$Z = \sum_{\substack{1 \leq j \leq d, \\ 1 \leq i \leq u, \\ s_1,...,s_i \in S_{\mathrm{p}}}} I_{j,s_1,...,s_i,1}.$$

Also for $1 \leq j \leq d$, $0 \leq i \leq u$ and $s_1, ..., s_i \in S_{\mathrm{p}}$, let

$$F(j, s_1, ..., s_i) = \sum_{\substack{i+1 \leq k \leq u, \\ s_{i+1},...,s_k \in S_{\mathrm{p}}}} I_{j,s_1,...,s_k,i+1}.$$

Then $Z = \sum_{j=1}^{d} F(j)$ and

$$\begin{gathered} \mathbb{E}[e^{tZ}] = \\ \mathbb{E}\left[\exp\left(t \cdot \sum_{j=1}^{d} F(j)\right)\right] = \\ \mathbb{E}\left[\prod_{j=1}^{d} e^{tF(j)}\right] = \\ \prod_{j=1}^{d} \mathbb{E}\left[e^{tF(j)}\right]. \end{gathered} \tag{7}$$

We will prove by induction that for all $1 \leq j \leq d$, $0 \leq k \leq u$ and $s_1, ..., s_{u-k} \in S_{\mathrm{p}}$,

$$\mathbb{E}\left[\exp\left(t \cdot F(j, s_1, ..., s_{u-k})\right)\right] = x_k.$$

Basis case ($k = 0$): $\mathbb{E}\left[\exp\left(t \cdot F(j, s_1, ..., s_u)\right)\right] = \mathbb{E}\left[\exp\left(t \cdot 0\right)\right] = 1 = x_0$.
Inductive step:

$$\mathbb{E}\left[\exp\left(t \cdot F(j, s_1, ..., s_{u-k-1})\right)\right] =$$
$$\mathbb{E}\left[\exp\left(t \cdot \sum_{\substack{u-k \le r \le u, \\ s_{u-k}, ..., s_r \in S_\mathrm{p}}} I_{j, s_1, ..., s_r, u-k}\right)\right] =$$
$$\mathbb{E}\left[\exp\left(t \cdot \sum_{s_{u-k} \in S_\mathrm{p}} \sum_{\substack{u-k \le r \le u, \\ s_{u-k+1}, ..., s_r \in S_\mathrm{p}}} I_{j, s_1, ..., s_r, u-k}\right)\right] =$$
$$\mathbb{E}\left[\prod_{s_{u-k} \in S_\mathrm{p}} \exp\left(t \cdot \sum_{\substack{u-k \le r \le u, \\ s_{u-k+1}, ..., s_r \in S_\mathrm{p}}} I_{j, s_1, ..., s_r, u-k}\right)\right].$$

Define the random variables

$$X_{s_{u-k}} = \sum_{\substack{u-k \le r \le u, \\ s_{u-k+1}, ..., s_r \in S_\mathrm{p}}} I_{j, s_1, ..., s_r, u-k}.$$

Since $X_{s_{u-k}}$ are all independent,

$$\begin{aligned} &\mathbb{E}\left[\exp\left(t \cdot F(j, s_1, ..., s_{u-k-1})\right)\right] = \\ &\mathbb{E}\left[\prod_{s_{u-k} \in S_\mathrm{p}} \exp\left(t \cdot X_{s_{u-k}}\right)\right] = \\ &\prod_{s_{u-k} \in S_\mathrm{p}} \mathbb{E}\left[\exp\left(t \cdot X_{s_{u-k}}\right)\right]. \end{aligned} \tag{8}$$

Let $E_{s_{u-k}}$ be the event that $H_1(s_1, ..., s_{u-k}) = 1$. Then

$$\begin{aligned} &\mathbb{E}\left[\exp\left(t \cdot X_{s_{u-k}}\right)\right] = \\ &\mathbb{E}\left[\exp\left(t \cdot X_{s_{u-k}}\right) \middle| E_{s_{u-k}}\right] \cdot \Pr\left[E_{s_{u-k}}\right] + \\ &\mathbb{E}\left[\exp\left(t \cdot X_{s_{u-k}}\right) \middle| \neg E_{s_{u-k}}\right] \cdot \Pr\left[\neg E_{s_{u-k}}\right] = \\ &\mathbb{E}\left[\exp\left(t \cdot X_{s_{u-k}}\right) \middle| E_{s_{u-k}}\right] \cdot \frac{1}{n_\mathrm{p}} + \mathbb{E}\left[\exp\left(t \cdot 0\right) \middle| E_{s_{u-k}}\right] \cdot \left(1 - \frac{1}{n_\mathrm{p}}\right) = \\ &\frac{1}{n_\mathrm{p}} \cdot \mathbb{E}\left[\exp\left(t \cdot X_{s_{u-k}}\right) \middle| E_{s_{u-k}}\right] + 1 - \frac{1}{n_\mathrm{p}}. \end{aligned} \tag{9}$$

Given $E_{s_{u-k}}$,

$$X_{s_{u-k}} =$$
$$\sum_{\substack{u-k\le r\le u,\\ s_{u-k+1},\ldots,s_r\in S_{\mathrm{p}}}} I_{j,s_1,\ldots,s_r,u-k} =$$
$$\sum_{\substack{u-k+1\le r\le u,\\ s_{u-k+1},\ldots,s_r\in S_{\mathrm{p}}}} I_{j,s_1,\ldots,s_r,u-k} + I_{j,s_1,\ldots,s_{u-k},u-k} =$$
$$\sum_{\substack{u-k+1\le r\le u,\\ s_{u-k+1},\ldots,s_r\in S_{\mathrm{p}}}} I_{j,s_1,\ldots,s_r,u-k+1} + 1 =$$
$$F(j,s_1,\ldots,s_{u-k}) + 1$$

Using equation 9,

$$\mathbb{E}\left[\exp\left(t\cdot X_{s_{u-k}}\right)\right] =$$
$$\frac{1}{n_{\mathrm{p}}}\cdot\mathbb{E}\left[\exp\left(t\cdot\left(F(j,s_1,\ldots,s_{u-k})+1\right)\right)\middle|E_{s_{u-k}}\right] + 1 - \frac{1}{n_{\mathrm{p}}} =$$
$$\frac{1}{n_{\mathrm{p}}}\cdot\mathbb{E}\left[\exp\left(t\cdot F(j,s_1,\ldots,s_{u-k})\right)\middle|E_{s_{u-k}}\right]\cdot e^t + 1 - \frac{1}{n_{\mathrm{p}}} =$$
$$\frac{1}{n_{\mathrm{p}}}\cdot\mathbb{E}\left[\exp\left(t\cdot F(j,s_1,\ldots,s_{u-k})\right)\right]\cdot e^t + 1 - \frac{1}{n_{\mathrm{p}}} =$$
$$\frac{1}{n_{\mathrm{p}}}x_k e^t + 1 - \frac{1}{n_{\mathrm{p}}}.$$

Combining this with equation 8 we get

$$\mathbb{E}\left[\exp\left(t\cdot F(j,s_1,\ldots,s_{u-k-1})\right)\right] =$$
$$\prod_{s_{u-k}\in S_{\mathrm{p}}}\mathbb{E}\left[\exp\left(t\cdot X_{s_{u-k}}\right)\right] =$$
$$\prod_{s_{u-k}\in S_{\mathrm{p}}}\left(\frac{1}{n_{\mathrm{p}}}x_k e^t + 1 - \frac{1}{n_{\mathrm{p}}}\right) =$$
$$\left(x_k e^t + 1 - \frac{1}{n_{\mathrm{p}}}\right)^{n_{\mathrm{p}}} =$$
$$x_{k+1}$$

which concludes the inductive step.

Therefore by equation 7, $\mathbb{E}[e^{tZ}] = \prod_{j=1}^{d}\mathbb{E}\left[e^{tF(j)}\right] = \prod_{j=1}^{d} x_u = x_u^d$.

D.3 Lemmas for Section 3.2

Lemma 10. *Let $\lambda > 0$, $d \geq 16u\lambda$, $q = \frac{2\lambda}{d}$, let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i, let E be the event that $\frac{1}{n_p}\sum_{i=1}^{n} e^{-qX_i} \leq e^{-q+4q^2}$ and let F be the event that the honest prover fails. Then $\Pr[F|E] \leq e^{-\lambda}$.*

Proof. Define random function $f(x) = \frac{1}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}} x^{X_i}$. By lemma 11, $\Pr[F|H_0] = \left(f^{(u)}(1-q)\right)^d$. Since $f(x)$ is an increasing function, this is at most $\left(f^{(u)}(e^{-q})\right)^d$ and by lemma 12, it is at most

$$\left(e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}}\right)^u\right)^d.$$

Therefore,

$$\Pr[F|E] = \mathbb{E}\left[\Pr[F|H_0, E]\middle|E\right] = \mathbb{E}\left[\Pr[F|H_0]\middle|E\right] =$$
$$\mathbb{E}\left[\Pr[F|H_0]\middle|f(e^{-q}) \leq e^{-q+4q^2}\right] \leq$$
$$\mathbb{E}\left[\left(e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}}\right)^u\right)^d\middle|f(e^{-q}) \leq e^{-q+4q^2}\right] \leq$$
$$e^{-(q-4uq^2)d}.$$

This is at most $e^{-\lambda}$ if and only if

$$(q-4uq^2)d \geq \lambda \iff$$
$$\left(\frac{2\lambda}{d} - 4u\left(\frac{2\lambda}{d}\right)^2\right)d \geq \lambda \iff$$
$$2 - 4u \cdot \frac{4\lambda}{d} \geq 1 \iff$$
$$1 \geq \frac{16u\lambda}{d} \iff$$
$$d \geq 16u\lambda$$

which is true by our assumption about d.

Lemma 11. *Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i, define random function $f(x) = \frac{1}{n_p}\sum_{i=1}^{n_p} x^{X_i}$ and let $G(t)$ be the event that there is no valid certificate with integer t. Then for all t, $\Pr[G(t)|H_0] = f^{(u)}(1-q)$.*

Proof. Let $F(t, s_1, ..., s_k)$ be the event that there is no suffix of honest player's signatures that works, meaning there is no $s_{k+1}, ..., s_u$ such that for all $k+1 \leq i \leq u$, $H_1(t, s_1, ..., s_{i-1}) = H_0(s_i)$ and $H_2(t, s_1, ..., s_u) = 1$. Then

- $F(t, s_1, ..., s_u)$ is true iff $H_2(t, s_1, ..., s_u) = 0$;

– for $0 \le k < u$: $F(t, s_1, ..., s_k) = \bigwedge_{s_{k+1} \in S_\mathrm{p}} \big((H_1(t, s_1, ..., s_k) \neq H_0(s_{k+1})) \vee F(t, s_1, ..., s_{k+1})\big)$;
– $F = \bigwedge_{i=1}^{d} F(i)$.

We will prove by induction that for all $0 \le i \le u$, $\Pr[F(t, s_1, ..., s_{u-i})|H_0] = f^{(i)}(1-q)$. The basis case is trivial: $\Pr[F(t, s_1, ..., s_u)|H_0] = \Pr[H_2(t, s_1, ..., s_u) = 0|H_0] = 1 - q = f^{(0)}(1-q)$. Inductive step:

$$\Pr[F(t, s_1, ..., s_{u-i-1})|H_0] =$$
$$\sum_{j=1}^{n_\mathrm{p}} \Pr[F(t, s_1, ..., s_{u-i-1})|H_0, H_1(t, s_1, ..., s_{u-i-1}) = j] \times$$
$$\Pr[H_1(t, s_1, ..., s_{u-i-1}) = j|H_0] =$$
$$\frac{1}{n_\mathrm{p}} \sum_{j=1}^{n_\mathrm{p}} \Pr[F(t, s_1, ..., s_{u-i-1})|H_0, H_1(t, s_1, ..., s_{u-i-1}) = j] =$$
$$\frac{1}{n_\mathrm{p}} \sum_{j=1}^{n_\mathrm{p}} \Pr[\bigwedge_{s_{u-i} \in S_\mathrm{p}, H_0(s_{u-i})=j} F(t, s_1, ..., s_{u-i})|H_0, H_1(t, s_1, ..., s_{u-i-1}) = j][=]$$

By the definition of F, $F(t, s_1, ..., s_{u-i})$ is independent of $H_1(t, s_1, ..., s_{u-i-1})$ even conditioned on H_0. Thus,

$$[=]\frac{1}{n_\mathrm{p}} \sum_{j=1}^{n_\mathrm{p}} \Pr[\bigwedge_{s_{u-i} \in S_\mathrm{p}, H_0(s_{u-i})=j} F(t, s_1, ..., s_{u-i})|H_0][=]$$

When H_0 is fixed, events $\{F(t, s_1, ..., s_{u-i}) : s_{u-i} \in S_\mathrm{p}, H_0(s_{u-i}) = j\}$ are independent since they only depend on the values of H_1 and H_2 with s_{u-i} in their inputs' $(u-i)$-th position. Therefore,

$$[=]\frac{1}{n_\mathrm{p}} \sum_{j=1}^{n_\mathrm{p}} \prod_{s_{u-i} \in S_\mathrm{p}, H_0(s_{u-i})=j} \Pr[F(t, s_1, ..., s_{u-i})|H_0] =$$
$$\frac{1}{n_\mathrm{p}} \sum_{j=1}^{n_\mathrm{p}} \prod_{s_{u-i} \in S_\mathrm{p}, H_0(s_{u-i})=j} f^{(i)}(1-q) =$$
$$\frac{1}{n_\mathrm{p}} \sum_{j=1}^{n_\mathrm{p}} \big(f^{(i)}(1-q)\big)^{X_j} =$$
$$f^{(i+1)}(1-q).$$

Hence, $\Pr[G(t)|H_0] = \Pr[F(t)|H_0] = f^{(u)}(1-q)$.

Lemma 12. *$n, u \in \mathbb{N}$ and define function $f(x) = \frac{1}{n}\sum_{i=1}^{n} x^{X_i}$ for some coefficients $\{X_i\}$ with $\sum_{i=1}^{n} X_i = n$. Then for $0 < z < 1$, $f^{(u)}(z) \le z \cdot \left(\frac{f(z)}{z}\right)^u$.*

Proof. Let $0 < z < 1$. Since the function z^x is convex, by Jensen's inequality, $f(z) = \frac{1}{n}\sum_{i=1}^{n} z^{X_i} \geq z^{\frac{1}{n}\sum_{i=1}^{n} X_i} = z$. So, the sequence $z, f(z), f^{(2)}(z), ...$ is non-decreasing and is < 1. Also, the function $g(x) = \frac{f(x)}{x}$ is non-increasing since

$$\left(\frac{f(x)}{x}\right)' = \left(\frac{\frac{1}{n}\sum_{i=1}^{n} x^{X_i}}{x}\right)' = \left(\frac{1}{n}\sum_{i=1}^{n} x^{X_i-1}\right)' =$$
$$\frac{1}{n}\sum_{i=1}^{n}(X_i-1)x^{X_i-2} =$$
$$\frac{x^{-2}}{n}\sum_{i=1}^{n}(X_i-1)x^{X_i} =$$
$$\frac{x^{-2}}{n}\left(\sum_{i:X_i\geq 1}(X_i-1)x^{X_i} - \sum_{i:X_i=0} 1\right) \leq$$
$$\frac{x^{-2}}{n}\left(\sum_{i:X_i\geq 1}(X_i-1) - \sum_{i:X_i=0} 1\right) =$$
$$\frac{x^{-2}}{n}\left(\sum_{i:X_i\geq 1} X_i - \sum_{i:X_i\geq 1} 1 - \sum_{i:X_i=0} 1\right) =$$
$$\frac{x^{-2}}{n}(n-n) = 0.$$

Hence, for all $i \geq 0$, $\frac{f^{(i+1)}(z)}{f^{(i)}(z)} = g\big(f^{(i)}(z)\big) \leq g(z) = \frac{f(z)}{z}$, and thus,

$$f^{(u)}(z) = z \cdot \prod_{i=0}^{u-1} \frac{f^{(i+1)}(z)}{f^{(i)}(z)} \leq z \cdot \left(\frac{f(z)}{z}\right)^u.$$

Lemma 13. *Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i. Then*

$$\Pr\left[\frac{1}{n_p}\sum_{i=1}^{n} e^{-qX_i} \geq 1 - q + 4q^2\right] \leq 2e^{-\frac{9}{4}n_p q^2}.$$

Proof. We use Poisson approximation: for $1 \leq i \leq n_{\mathrm{p}}$, let Y_i be independent Poisson random variables with mean 1; i.e., for all integers $j \geq 0$, $\Pr[Y_i = j] = \frac{1}{ej!}$. Then by [MU05, theorem 5.10],

$$\Pr\left[\frac{1}{n_{\mathrm{p}}}\sum_{i=1}^{n_{\mathrm{p}}} e^{-qX_i} \geq 1 - q + 4q^2\right] \leq 2 \cdot \Pr\left[\frac{1}{n_{\mathrm{p}}}\sum_{i=1}^{n_{\mathrm{p}}} e^{-qY_i} \geq 1 - q + 4q^2\right]. \quad (10)$$

This arithmetic average can be analyzed using Hoeffding bound, but it doesn't give the best result. Instead, we derive a custom moment generating function

for the summand. For any $t > 0$,

$$\mathbb{E}\left[e^{te^{-qY_i}}\right] =$$
$$\sum_{i=0}^{\infty} \frac{e^{te^{-qi}}}{ei!} =$$
$$e^{t-1} \sum_{i=0}^{\infty} \frac{e^{t(e^{-qi}-1)}}{i!} \leq$$
$$e^{t-1} \sum_{i=0}^{\infty} \frac{1 + t(e^{-qi} - 1) + \frac{t^2(1-e^{-qi})^2}{2}}{i!} \leq$$
$$e^{t-1} \sum_{i=0}^{\infty} \frac{1 + t(1 - qi + \frac{(qi)^2}{2} - 1) + \frac{t^2(1-e^{-qi})^2}{2}}{i!} =$$
$$e^{t-1} \sum_{i=0}^{\infty} \frac{1 - tqi + t\frac{(qi)^2}{2} + \frac{t^2(1-e^{-qi})^2}{2}}{i!} [\leq]$$

Since $0 < 1 - e^{-qi} \leq 1 - (1 - qi) = qi$,

$$[\leq] e^{t-1} \sum_{i=0}^{\infty} \frac{1 - tqi + t\frac{(qi)^2}{2} + t^2\frac{(qi)^2}{2}}{i!} =$$
$$e^{t-1} \left(\sum_{i=0}^{\infty} \frac{1}{i!} - tq \sum_{i=0}^{\infty} \frac{i}{i!} + (t + t^2)\frac{q^2}{2} \sum_{i=0}^{\infty} \frac{i^2}{i!} \right) [=]$$

By lemma 4,

$$[=] e^{t-1} \left(e - tqe + (t + t^2)\frac{q^2}{2} \cdot 2e \right) =$$
$$e^t \left(1 - tq + (t + t^2)q^2\right) \leq$$
$$e^t \cdot e^{-tq+(t+t^2)q^2} =$$
$$e^{t(1-q+(1+t)q^2)}.$$

Combining this bound, equation 10 and Markov's inequality, for any $s > 0$ we get

$$\Pr\left[\frac{1}{n_{\mathrm{p}}}\sum_{i=1}^{n_{\mathrm{p}}} e^{-qX_i} \geq 1-q+4q^2\right] \leq$$
$$2\cdot\Pr\left[e^{\frac{s}{n_{\mathrm{p}}}\sum_{i=1}^{n_{\mathrm{p}}} e^{-qY_i}} \geq e^{s(1-q+4q^2)}\right] \leq$$
$$2\cdot\frac{\mathbb{E}\left[e^{\frac{s}{n_{\mathrm{p}}}\sum_{i=1}^{n_{\mathrm{p}}} e^{-qY_i}}\right]}{e^{s(1-q+4q^2)}} =$$
$$2\cdot\frac{\mathbb{E}\left[\prod_{i=1}^{n_{\mathrm{p}}} e^{\frac{s}{n_{\mathrm{p}}}e^{-qY_i}}\right]}{e^{s(1-q+4q^2)}} =$$
$$2\cdot\frac{\prod_{i=1}^{n_{\mathrm{p}}}\mathbb{E}\left[e^{\frac{s}{n_{\mathrm{p}}}e^{-qY_i}}\right]}{e^{s(1-q+4q^2)}} \leq$$
$$2\cdot\frac{\prod_{i=1}^{n_{\mathrm{p}}} e^{\frac{s}{n_{\mathrm{p}}}\left(1-q+(1+\frac{s}{n_{\mathrm{p}}})q^2\right)}}{e^{s(1-q+4q^2)}} =$$
$$2\cdot\frac{e^{s\left(1-q+(1+\frac{s}{n_{\mathrm{p}}})q^2\right)}}{e^{s(1-q+4q^2)}} =$$
$$2e^{-\left(4-1-\frac{s}{n_{\mathrm{p}}}\right)tq^2}.$$

Setting $s = \frac{3}{2}n_{\mathrm{p}}$, we get

$$\Pr\left[\frac{1}{n_{\mathrm{p}}}\sum_{i=1}^{n} e^{-qX_i} \geq 1-q+4q^2\right] \leq 2e^{-\frac{9}{4}n_{\mathrm{p}}q^2}.$$

D.4 Lemmas for Section 3.2

Lemma 14. *Let $\lambda, \alpha, c > 0$,*

$$\delta = e^{cu\alpha}\left(\frac{\lambda}{d\alpha}+1\right),$$

$X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i and let E be the event that $\frac{1}{n_p}\sum_{i=1}^{n_p} e^{\alpha\cdot X_i} \leq e^{\alpha+c\alpha^2}$ with $\Pr[E] > 0$. Then $\Pr[Z \geq \delta du|E] \leq e^{-\lambda}$.

Proof. Set

$$t = \frac{\alpha}{e^{cu\alpha}\cdot u} \tag{11}$$

and define the random sequence $\{G_k\}$ as follows: let $G_0 = 1$ and for $k \geq 0$, let

$$G_{k+1} = \frac{1}{n_{\mathrm{p}}}\sum_{i=1}^{n_{\mathrm{p}}}\left(G_k\cdot e^t\right)^{X_i}.$$

By lemma 15, $\mathbb{E}\left[e^{tZ}|H_0\right] = G_u^d$.

For all $0 \leq k \leq u$, define $y_k = kte^{ck\alpha}$. We will prove by induction that given event E, for $0 \leq k \leq u$, $G_k \leq e^{y_k}$.
Basis case: $G_0 = 1 \leq 1 = e^{y_0}$.
Inductive step:

$$G_{k+1} = \frac{1}{n_{\mathrm{p}}} \sum_{i=1}^{n_{\mathrm{p}}} \left(G_k \cdot e^t\right)^{X_i} \leq \frac{1}{n_{\mathrm{p}}} \sum_{i=1}^{n_{\mathrm{p}}} e^{(y_k+t)X_i} =$$
$$\frac{1}{n_{\mathrm{p}}} \sum_{i=1}^{n_{\mathrm{p}}} \exp\left((kte^{ck\alpha}+t)X_i\right) \leq$$
$$\frac{1}{n_{\mathrm{p}}} \sum_{i=1}^{n_{\mathrm{p}}} \exp\left((k+1)te^{ck\alpha}X_i\right)[\leq].$$

Since $(k+1)te^{ck\alpha} \leq ute^{ck\alpha} \leq \alpha$, the function $f(x) = x^{\frac{(k+1)te^{ck\alpha}}{\alpha}}$ is concave and by Jensen's inequality,

$$[\leq]\left(\frac{1}{n_{\mathrm{p}}} \sum_{i=1}^{n_{\mathrm{p}}} e^{\alpha X_i}\right)^{\frac{(k+1)te^{ck\alpha}}{\alpha}} \leq$$
$$\left(e^{\alpha+c\alpha^2}\right)^{\frac{(k+1)te^{ck\alpha}}{\alpha}} =$$
$$\exp\left(\alpha(1+c\alpha)\frac{(k+1)te^{ck\alpha}}{\alpha}\right) =$$
$$\exp\left((k+1)te^{ck\alpha}(1+c\alpha)\right) \leq$$
$$\exp\left((k+1)te^{ck\alpha}e^{c\alpha}\right) =$$
$$\exp\left((k+1)te^{c(k+1)\alpha}\right) =$$
$$e^{y_{k+1}}.$$

Hence,

$$\mathbb{E}\left[e^{tZ}\middle|E\right] = \mathbb{E}\left[\mathbb{E}\left[e^{tZ}|H_0,E\right]\middle|E\right] = \mathbb{E}\left[\mathbb{E}\left[e^{tZ}|H_0\right]\middle|E\right] = \mathbb{E}\left[G_u^d\middle|E\right] \leq$$
$$\mathbb{E}\left[(e^{y_u})^d\middle|E\right] = e^{dy_u} \leq e^{d\alpha}.$$

By Markov's inequality,

$$\Pr[Z \geq \delta du|E] = \Pr\left[e^{tZ} \geq e^{\delta tdu}\middle|E\right] \leq$$
$$\frac{\mathbb{E}\left[e^{tZ}\middle|E\right]}{e^{\delta tdu}} \leq \frac{e^{d\alpha}}{e^{\delta tdu}} = \exp\left(-d(\delta tu-\alpha)\right).$$

This is at most $e^{-\lambda}$ if and only if

$$d(\delta tu - \alpha) \geq \lambda \iff$$
$$\delta tu - \alpha \geq \frac{\lambda}{d} \iff$$
$$\delta \geq \frac{\frac{\lambda}{d} + \alpha}{tu} [\iff]$$

Substituting the value of t from equation 11,

$$[\iff] \delta \geq \left(\frac{\lambda}{d} + \alpha\right) \frac{e^{cu\alpha}}{\alpha} \iff$$
$$\delta \geq e^{cu\alpha} \left(\frac{\lambda}{d\alpha} + 1\right)$$

which is true by the statement of the lemma.

Lemma 15. *Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i, let $t > 0$ and define the random sequence $\{G_k\}$ as follows: let $G_0 = 1$ and for $k \geq 0$, let*

$$G_{k+1} = \frac{1}{n_p} \sum_{i=1}^{n_p} \left(G_k \cdot e^t\right)^{X_i}.$$

Then $\mathbb{E}\left[e^{tZ} | H_0\right] = G_u^d$.

Proof. For $1 \leq j \leq d$, $1 \leq i \leq u$, $s_1, ..., s_i \in S_\text{p}$ and $1 \leq k \leq i$, let the indicator random variable

$$I_{j,s_1,...,s_i,k} = \begin{cases} 1 & \text{if for all } k \leq r \leq i,\ H_1(j, s_1, ..., s_{r-1}) = H_0(s_r) \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$Z = \sum_{\substack{1 \leq j \leq d, \\ 1 \leq i \leq u, \\ s_1,...,s_i \in S_\text{p}}} I_{j,s_1,...,s_i,1}.$$

Also for $1 \leq j \leq d$, $0 \leq i \leq u$ and $s_1, ..., s_i \in S_\text{p}$, let

$$F(j, s_1, ..., s_i) = \sum_{\substack{i+1 \leq k \leq u, \\ s_{i+1},...,s_k \in S_\text{p}}} I_{j,s_1,...,s_k,i+1}.$$

Then $Z = \sum_{j=1}^{d} F(j)$ and

$$\begin{gathered}\mathbb{E}[e^{tZ}|H_0] = \\ \mathbb{E}\left[\exp\left(t \cdot \sum_{j=1}^{d} F(j)\right)\middle| H_0\right] = \\ \mathbb{E}\left[\prod_{j=1}^{d} e^{tF(j)}\middle| H_0\right] = \\ \prod_{j=1}^{d} \mathbb{E}\left[e^{tF(j)}\middle| H_0\right].\end{gathered} \tag{12}$$

Now we will prove by induction that for all $1 \leq j \leq d$, $0 \leq k \leq u$ and $s_1, ..., s_{u-k} \in S_{\mathrm{p}}$,

$$\mathbb{E}\left[\exp\left(t \cdot F(j, s_1, ..., s_{u-k})\right)\middle| H_0\right] = G_k.$$

Basis case ($k = 0$): $\mathbb{E}\left[\exp\left(t \cdot F(j, s_1, ..., s_u)\right)\middle| H_0\right] = \mathbb{E}\left[\exp\left(t \cdot 0\right)\middle| H_0\right] = 1 = G_0$.

Inductive step:

$$\mathbb{E}\left[\exp\left(t\cdot F(j,s_1,...,s_{u-k-1})\right)\middle|H_0\right]=$$

$$\mathbb{E}\left[\exp\left(t\cdot\sum_{\substack{u-k\le r\le u,\\ s_{u-k},...,s_r\in S_{\mathrm{p}}}}I_{j,s_1,...,s_r,u-k}\right)\middle|H_0\right]=$$

$$\sum_{b=1}^{n_{\mathrm{p}}}\mathbb{E}\left[\exp\left(t\cdot\sum_{\substack{u-k\le r\le u,\\ s_{u-k},...,s_r\in S_{\mathrm{p}}}}I_{j,s_1,...,s_r,u-k}\right)\middle|H_1(j,s_1,...,s_{u-k-1})=b,H_0\right]\times$$

$$\Pr\left[H_1(j,s_1,...,s_{u-k-1})=b\middle|H_0\right]=$$

$$\frac{1}{n_{\mathrm{p}}}\sum_{b=1}^{n_{\mathrm{p}}}\mathbb{E}\left[\exp\left(t\cdot\sum_{\substack{u-k\le r\le u,\\ s_{u-k},...,s_r\in S_{\mathrm{p}},\\ H_0(s_{u-k})=b}}I_{j,s_1,...,s_r,u-k}\right)\middle|H_1(j,s_1,...,s_{u-k-1})=b,H_0\right]=$$

$$\frac{1}{n_{\mathrm{p}}}\sum_{b=1}^{n_{\mathrm{p}}}\mathbb{E}\left[\exp\left(t\cdot\sum_{\substack{s_{u-k}\in S_{\mathrm{p}},\\ H_0(s_{u-k})=b}}\sum_{\substack{u-k\le r\le u,\\ s_{u-k+1},...,s_r\in S_{\mathrm{p}}}}I_{j,s_1,...,s_r,u-k}\right)\middle|H_1(j,s_1,...,s_{u-k-1})=b,H_0\right]=$$

$$\frac{1}{n_{\mathrm{p}}}\sum_{b=1}^{n_{\mathrm{p}}}\mathbb{E}\left[\prod_{\substack{s_{u-k}\in S_{\mathrm{p}},\\ H_0(s_{u-k})=b}}\exp\left(t\cdot\sum_{\substack{u-k\le r\le u,\\ s_{u-k+1},...,s_r\in S_{\mathrm{p}}}}I_{j,s_1,...,s_r,u-k}\right)\middle|H_1(j,s_1,...,s_{u-k-1})=b,H_0\right]=$$

$$\frac{1}{n_{\mathrm{p}}}\sum_{b=1}^{n_{\mathrm{p}}}\mathbb{E}\left[\prod_{\substack{s_{u-k}\in S_{\mathrm{p}},\\ H_0(s_{u-k})=b}}\exp\left(t\cdot\left(I_{j,s_1,...,s_{u-k},u-k}+\sum_{\substack{u-k+1\le r\le u,\\ s_{u-k+1},...,s_r\in S_{\mathrm{p}}}}I_{j,s_1,...,s_r,u-k}\right)\right)\middle|\right.$$

$$\left.H_1(j,s_1,...,s_{u-k-1})=b,H_0\right]=$$

$$\frac{1}{n_{\mathrm{p}}}\sum_{b=1}^{n_{\mathrm{p}}}\mathbb{E}\left[\prod_{\substack{s_{u-k}\in S_{\mathrm{p}},\\ H_0(s_{u-k})=b}}\exp\left(t\cdot\left(1+\sum_{\substack{u-k+1\le r\le u,\\ s_{u-k+1},...,s_r\in S_{\mathrm{p}}}}I_{j,s_1,...,s_r,u-k+1}\right)\right)\middle|\right.$$

$$\left.H_1(j,s_1,...,s_{u-k-1})=b,H_0\right]=$$

$$\frac{1}{n_{\mathrm{p}}}\sum_{b=1}^{n_{\mathrm{p}}}\mathbb{E}\left[\prod_{\substack{s_{u-k}\in S_{\mathrm{p}},\\ H_0(s_{u-k})=b}}\exp\left(t\cdot\left(1+F(j,s_1,...,s_{u-k})\right)\right)\middle|H_0\right][=]$$

Since with fixed H_0, $F(j, s_1, ..., s_{u-k})$ for $s_{u-k} \in S_\mathrm{p}$ are all independent,

$$[=]\frac{1}{n_\mathrm{p}} \sum_{b=1}^{n_\mathrm{p}} \prod_{\substack{s_{u-k} \in S_\mathrm{p}, \\ H_0(s_{u-k})=b}} \mathbb{E}\left[\exp\left(t \cdot \left(1 + F(j, s_1, ..., s_{u-k})\right)\right) \middle| H_0 \right] =$$

$$\frac{1}{n_\mathrm{p}} \sum_{b=1}^{n_\mathrm{p}} \prod_{\substack{s_{u-k} \in S_\mathrm{p}, \\ H_0(s_{u-k})=b}} \left(e^t \cdot \mathbb{E}\left[\exp\left(t \cdot F(j, s_1, ..., s_{u-k})\right) \middle| H_0 \right] \right) =$$

$$\frac{1}{n_\mathrm{p}} \sum_{b=1}^{n_\mathrm{p}} \prod_{\substack{s_{u-k} \in S_\mathrm{p}, \\ H_0(s_{u-k})=b}} \left(e^t \cdot G_k \right) =$$

$$\frac{1}{n_\mathrm{p}} \sum_{b=1}^{n_\mathrm{p}} \left(G_k \cdot e^t \right)^{X_b} =$$

$$G_{k+1}$$

which concludes the inductive step.

Therefore by equation 12, $\mathbb{E}[e^{tZ}|H_0] = \prod_{j=1}^{d} \mathbb{E}\left[e^{tF(j)} \middle| H_0 \right] = \prod_{j=1}^{d} G_u = G_u^d$.

Lemma 16. *Let $\lambda > 0$, $u \in \mathbb{N}$, $0 < \alpha \leq \frac{1}{2u}$, $n \geq \frac{u^2 \lambda}{2}$,*

$$c = \frac{3}{\alpha} \cdot \sqrt{\frac{2\lambda}{n}} + 2,$$

Y_i be Poisson random variables with expectation 1 and let

$$A_i = \begin{cases} e^{\alpha Y_i} & \text{if } Y_i \leq u \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\Pr\left[\frac{1}{n} \sum_{i=1}^{n} A_i \geq 1 + \alpha + c\alpha^2 \right] \leq e^{-\lambda}.$$

Proof. Let $0 < r \leq \frac{2}{3\alpha u}$. We calculate the following:

$$\mathbb{E}\left[e^{rA_i}\right] =$$
$$\mathbb{E}\left[e^{rA_i}\middle|Y_i \leq u\right] \cdot \Pr[Y_i \leq u] + \mathbb{E}\left[e^{rA_i}\middle|Y_i > u\right] \cdot \Pr[Y_i > u] =$$
$$\mathbb{E}\left[\exp\left(re^{\alpha Y_i}\right)\middle|Y_i \leq u\right] \cdot \Pr[Y_i \leq u] + \Pr[Y_i > u] =$$
$$\Pr[Y_i \leq u] \cdot \sum_{j=0}^{u} \exp\left(re^{\alpha j}\right) \cdot \Pr[Y_i = j | Y_i \leq u] + \Pr[Y_i > u] =$$
$$\Pr[Y_i \leq u] \cdot \sum_{j=0}^{u} \exp\left(re^{\alpha j}\right) \cdot \frac{\Pr[Y_i = j]}{\Pr[Y_i \leq u]} + \Pr[Y_i > u] =$$
$$\sum_{j=0}^{u} \frac{\exp\left(re^{\alpha j}\right)}{ej!} + \Pr[Y_i > u] =$$
$$e^{r-1} \sum_{j=0}^{u} \frac{\exp\left(r\left(e^{\alpha j} - 1\right)\right)}{j!} + \Pr[Y_i > u][\leq]$$

The next step uses the fact that when $x \leq 1$, $e^x \leq 1 + x + x^2$. First note that $\alpha u \leq \frac{1}{2}$ by the statement of the lemma. Therefore, $r\left(e^{\alpha j} - 1\right) \leq r\left(e^{\alpha u} - 1\right) \leq r(1 + \alpha u + (\alpha u)^2 - 1) = r(\alpha u + (\alpha u)^2) \leq r \cdot \frac{3}{2}\alpha u \leq \frac{2}{3\alpha u} \cdot \frac{3}{2}\alpha u = 1$. Thus,

$$[\leq] e^{r-1} \sum_{j=0}^{u} \frac{1 + r\left(e^{\alpha j} - 1\right) + r^2\left(e^{\alpha j} - 1\right)^2}{j!} + \Pr[Y_i > u] \leq$$
$$e^{r-1} \sum_{j=0}^{u} \frac{1 + r\left(\alpha j + \alpha^2 j^2\right) + r^2\left(\alpha j + \alpha^2 j^2\right)^2}{j!} + \Pr[Y_i > u] \leq$$
$$e^{r-1} \sum_{j=0}^{u} \frac{1 + r\left(\alpha j + \alpha^2 j^2\right) + r^2\left(\frac{3}{2}\alpha j\right)^2}{j!} + \Pr[Y_i > u] =$$
$$e^{r-1} \sum_{j=0}^{u} \frac{1 + r\alpha j + \left(r + \frac{9}{4}r^2\right)\alpha^2 j^2}{j!} + \Pr[Y_i > u] =$$
$$e^{r-1} \sum_{j=0}^{u} \frac{1 + r\alpha j + \left(r + \frac{9}{4}r^2\right)\alpha^2 j^2}{j!} + \sum_{u+1}^{\infty} \frac{1}{ej!} \leq$$
$$e^{r-1} \sum_{j=0}^{\infty} \frac{1 + r\alpha j + \left(r + \frac{9}{4}r^2\right)\alpha^2 j^2}{j!} =$$
$$e^{r-1}\left(\sum_{j=0}^{\infty} \frac{1}{j!} + r\alpha \sum_{j=0}^{\infty} \frac{j}{j!} + \left(r + \frac{9}{4}r^2\right)\alpha^2 \sum_{j=0}^{\infty} \frac{j^2}{j!}\right)[=]$$

By lemma 4,

$$[=]e^{r-1}\left(e + r\alpha \cdot e + \left(r + \frac{9}{4}r^2\right)\alpha^2 \cdot 2e\right) =$$
$$e^r\left(1 + r\alpha + 2\left(r + \frac{9}{4}r^2\right)\alpha^2\right) \leq$$
$$e^r e^{r\alpha + 2\left(r+\frac{9}{4}r^2\right)\alpha^2} =$$
$$e^{r + r\alpha + 2\left(r+\frac{9}{4}r^2\right)\alpha^2} =$$
$$e^{r\left(1+\alpha+2\left(1+\frac{9}{4}r\right)\alpha^2\right)}.$$

We are now ready to bound $\frac{1}{n}\sum_{i=1}^n A_i$. Assume $s > 0$ and $\frac{s}{n} \leq \frac{2}{3\alpha u}$. By Markov's inequality,

$$\Pr\left[\frac{1}{n}\sum_{i=1}^n A_i \geq 1 + \alpha + c\alpha^2\right] =$$
$$\Pr\left[e^{\frac{s}{n}\sum_{i=1}^n A_i} \geq e^{s(1+\alpha+c\alpha^2)}\right] \leq$$
$$\frac{\mathbb{E}\left[e^{\frac{s}{n}\sum_{i=1}^n A_i}\right]}{e^{s(1+\alpha+c\alpha^2)}} =$$
$$\frac{\mathbb{E}\left[\prod_{i=1}^n e^{\frac{s}{n}A_i}\right]}{e^{s(1+\alpha+c\alpha^2)}} =$$
$$\frac{\prod_{i=1}^n \mathbb{E}\left[e^{\frac{s}{n}A_i}\right]}{e^{s(1+\alpha+c\alpha^2)}} \leq$$
$$\frac{\prod_{i=1}^n e^{\frac{s}{n}\left(1+\alpha+2\left(1+\frac{9s}{4n}\right)\alpha^2\right)}}{e^{s(1+\alpha+c\alpha^2)}} =$$
$$\frac{e^{s\left(1+\alpha+2\left(1+\frac{9s}{4n}\right)\alpha^2\right)}}{e^{s(1+\alpha+c\alpha^2)}} =$$
$$\exp\left(-\left(c - 2 - \frac{9s}{2n}\right)s\alpha^2\right).$$

This is at most $e^{-\lambda}$ if and only if

$$\left(c - 2 - \frac{9s}{2n}\right)s\alpha^2 \geq \lambda \iff c \geq \frac{\lambda}{s\alpha^2} + \frac{9s}{2n} + 2;$$

thus we set $c := \frac{\lambda}{s\alpha^2} + \frac{9s}{2n} + 2$. Differentiating with respect to s we find that the minimum is achieved when $s = \frac{\sqrt{2\lambda n}}{3\alpha}$. Then c becomes

$$\frac{3}{\alpha} \cdot \sqrt{\frac{2\lambda}{n}} + 2.$$

The requirement that $\frac{s}{n} \leq \frac{2}{3\alpha u}$ follows from $n \geq \frac{u^2\lambda}{2}$.

Lemma 17. *Let $\lambda > 0$, $u, n \in \mathbb{N}$, Y_i be Poisson random variables with expectation 1,*

$$B_i = \begin{cases} 0 & \text{if } Y_i \leq u \\ e^{\frac{Y_i}{3u}} & \text{otherwise} \end{cases}$$

and assume

$$u! \cdot \frac{u - e^{\frac{1}{3u}}}{u^3} \geq 9e^{-\frac{2}{3}} \cdot 2^{\lambda}.$$

Then

$$\Pr\left[\frac{1}{n}\sum_{i=1}^{n} B_i \geq \frac{1}{9u^2}\right] \leq 2^{-\lambda}.$$

Proof. First we bound the following:

$$\mathbb{E}[B_i] =$$
$$\mathbb{E}[B_i | Y_i \leq u] \cdot \Pr[Y_i \leq u] + \mathbb{E}[B_i | Y_i > u] \cdot \Pr[Y_i > u] =$$
$$\mathbb{E}\left[e^{\frac{Y_i}{3u}} \middle| Y_i > u\right] \cdot \Pr[Y_i > u] =$$
$$\Pr[Y_i > u] \cdot \sum_{i=u+1}^{\infty} e^{\frac{j}{3u}} \cdot \Pr[Y_i = j | Y_i > u] =$$
$$\Pr[Y_i > u] \cdot \sum_{i=u+1}^{\infty} e^{\frac{j}{3u}} \cdot \frac{\Pr[Y_i = j]}{\Pr[Y_i > u]} =$$
$$\sum_{j=u+1}^{\infty} \frac{e^{\frac{j}{3u}}}{ej!} \leq \sum_{j=u}^{\infty} \frac{e^{\frac{j}{3u}}}{ej!} \leq \frac{e^{-\frac{2}{3}}}{u!} \sum_{j=0}^{\infty} \left(\frac{e^{\frac{1}{3u}}}{u}\right)^j =$$
$$\frac{e^{-\frac{2}{3}}}{u!} \cdot \frac{1}{1 - \frac{\exp\left(\frac{1}{3u}\right)}{u}} = \frac{e^{-\frac{2}{3}} u}{\left(u - e^{\frac{1}{3u}}\right) u!}.$$

Then by Markov's inequality,

$$\Pr\left[\frac{1}{n}\sum_{i=1}^{n} B_i \geq \frac{1}{9u^2}\right] \leq \frac{\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} B_i\right]}{\frac{1}{9u^2}} = \frac{9u^2}{n} \sum_{i=1}^{n} \mathbb{E}[B_i] \leq$$
$$9u^2 \cdot \frac{e^{-\frac{2}{3}} u}{\left(u - e^{\frac{1}{3u}}\right) u!} = \frac{9e^{-\frac{2}{3}} u^3}{\left(u - e^{\frac{1}{3u}}\right) u!}.$$

This is at most $2^{-\lambda}$ if and only if

$$u! \cdot \frac{u - e^{\frac{1}{3u}}}{u^3} \geq 9e^{-\frac{2}{3}} \cdot 2^{\lambda}.$$

D.5 Lemmas for Section 4.2

Lemma 18. *Assume*

$$\mu \leq \frac{(\frac{4}{e})^{\lambda_{rel}}}{4e^{10}}; \delta = \sqrt{\frac{\lambda_{rel}}{4\mu}}; n_p \geq 2\mu$$

and X_i be Bernoulli random variables with probability $\frac{\mu}{n_p}$ for $1 \leq i \leq n_p$. Then

$$\Pr\left[\sum_{i=1}^{n_p} X_i \leq (1-\delta)\mu\right] \geq 2^{-\lambda_{rel}+1}.$$

Proof. Let $n = n_{\mathrm{p}}$, $k = (1-\delta)\mu$, $Y_i = 1 - X_i$ and $p = \frac{\mu}{n_{\mathrm{p}}}$. Then

$$\Pr\left[\sum_{i=1}^{n_{\mathrm{p}}} X_i \leq (1-\delta)\mu\right] =$$
$$\Pr\left[\sum_{i=1}^{n}(1-Y_i) \leq k\right] =$$
$$\Pr\left[\sum_{i=1}^{n} Y_i \geq n-k\right] =$$
$$\sum_{i=\lceil n-k\rceil}^{n} C(n,i)\cdot(1-p)^i p^{n-i}[\geq]$$

Define KL divergence $D(a \parallel p) = a\ln\frac{a}{p} + (1-a)\ln\frac{1-a}{1-p}$. By [Ash90], page 115,

$$[\geq]\frac{1}{\sqrt{8n\cdot\frac{\lceil n-k\rceil}{n}\left(1-\frac{\lceil n-k\rceil}{n}\right)}}\cdot\exp\left(-nD\left(\frac{\lceil n-k\rceil}{n}\parallel 1-p\right)\right) =$$
$$\frac{1}{\sqrt{8n\cdot\frac{\lceil n-k\rceil}{n}\left(1-\frac{\lceil n-k\rceil}{n}\right)}}\cdot\exp\left(-nD\left(1-\frac{\lceil n-k\rceil}{n}\parallel p\right)\right) =$$
$$\frac{1}{\sqrt{8n\cdot\frac{n-\lfloor k\rfloor}{n}\left(1-\frac{n-\lfloor k\rfloor}{n}\right)}}\cdot\exp\left(-nD\left(1-\frac{n-\lfloor k\rfloor}{n}\parallel p\right)\right) =$$
$$\frac{1}{\sqrt{8n\cdot\frac{\lfloor k\rfloor}{n}\left(1-\frac{\lfloor k\rfloor}{n}\right)}}\cdot\exp\left(-nD\left(\frac{\lfloor k\rfloor}{n}\parallel p\right)\right) \geq$$
$$\frac{1}{\sqrt{k}}\cdot\exp\left(-nD\left(\frac{\lfloor k\rfloor}{n}\parallel p\right)\right) \geq$$
$$\frac{1}{\sqrt{k}}\cdot\exp\left(-nD\left(\frac{k-1}{n}\parallel p\right)\right).$$

This is at least $2^{-\lambda_{\mathrm{rel}}+1}$ if and only if

$$\frac{1}{2}\ln k + nD\left(\frac{k-1}{n} \parallel p\right) \leq (\lambda_{\mathrm{rel}} - 1)\ln 2.$$

$$\frac{1}{2}\ln k + nD\left(\frac{k-1}{n} \parallel p\right) =$$

$$\frac{1}{2}\ln k + nD\left((1-\delta)p - \frac{1}{n} \parallel p\right) =$$

$$\frac{1}{2}\ln k + n\left(\left((1-\delta)p - \frac{1}{n}\right)\ln\frac{(1-\delta)p - \frac{1}{n}}{p} + \left(1-(1-\delta)p + \frac{1}{n}\right)\ln\frac{1-(1-\delta)p+\frac{1}{n}}{1-p}\right) \leq$$

$$\frac{1}{2}\ln k + n\left((1-\delta)p\cdot\ln(1-\delta) + \left(1-(1-\delta)p + \frac{1}{n}\right)\ln\frac{1-(1-\delta)p+\frac{1}{n}}{1-p}\right) =$$

$$\frac{1}{2}\ln k + n\left((1-\delta)p\cdot\ln(1-\delta) + \left(1-p+\delta p + \frac{1}{n}\right)\ln\left(1+\frac{\delta p+\frac{1}{n}}{1-p}\right)\right) \leq$$

$$\frac{1}{2}\ln k + n\left((1-\delta)p\cdot\ln(1-\delta) + \left(1-p+\delta p + \frac{1}{n}\right)\cdot\frac{\delta p+\frac{1}{n}}{1-p}\right) =$$

$$\frac{1}{2}\ln k + n\left((1-\delta)p\cdot\ln(1-\delta) + \left(1+\frac{\delta p+\frac{1}{n}}{1-p}\right)\cdot\left(\delta p + \frac{1}{n}\right)\right) =$$

$$\frac{1}{2}\ln k + n\left((1-\delta)p\cdot\ln(1-\delta) + \left(1+\frac{p}{1-p}\delta + \frac{1}{(1-p)n}\right)\cdot\left(\delta p + \frac{1}{n}\right)\right)[\leq]$$

Since $p = \frac{\mu}{n_{\mathrm{p}}} \leq \frac{1}{2}$,

$$
\begin{aligned}
[\leq] \frac{1}{2}\ln k + n\left((1-\delta)p \cdot \ln(1-\delta) + \left(1+\delta+\frac{2}{n}\right)\cdot\left(\delta p + \frac{1}{n}\right)\right) = \\
\frac{1}{2}\ln k + pn\left((1-\delta)\cdot\ln(1-\delta) + \left(1+\delta+\frac{2}{n}\right)\cdot\left(\delta + \frac{1}{pn}\right)\right) \leq \\
\frac{1}{2}\ln k + pn\left((1-\delta)\cdot(-\delta) + \left(1+\delta+\frac{2}{n}\right)\cdot\left(\delta + \frac{1}{pn}\right)\right) = \\
\frac{1}{2}\ln k + \mu\left((1-\delta)\cdot(-\delta) + \left(1+\delta+\frac{2}{n}\right)\cdot\left(\delta + \frac{1}{\mu}\right)\right) = \\
\frac{1}{2}\ln k + \mu\left(-\delta+\delta^2+\delta+\frac{1}{\mu}+\delta^2+\frac{\delta}{\mu}+\frac{2\delta}{n}+\frac{2}{\mu n}\right) = \\
\frac{1}{2}\ln k + \mu\big(-\delta+\delta^2+\delta+\delta^2\big) + 1 + \delta + \frac{2\delta\mu}{n} + \frac{2}{n} \leq \\
\frac{1}{2}\ln k + 2\delta^2\mu + 1 + \delta + \frac{2\delta\mu}{n} + \frac{2}{n} \leq \\
\frac{1}{2}\ln k + 2\delta^2\mu + 1 + 1 + 1 + 2 = \\
\frac{1}{2}\ln k + 2\delta^2\mu + 5.
\end{aligned}
$$

This is at most $(\lambda_{\mathrm{rel}} - 1)\ln 2 = \frac{\lambda_{\mathrm{rel}}-1}{\log e}$ if and only if $2\delta^2\mu \leq \frac{\lambda_{\mathrm{rel}}-1-5\log e}{\log e} - \frac{1}{2}\ln\mu$. We claim $2\delta^2\mu \leq \frac{\lambda_{\mathrm{rel}}}{2} \leq \frac{\lambda_{\mathrm{rel}}-1-5\log e}{\log e} - \frac{1}{2}\ln\mu$. The first inequality follows from the definition of δ. The second follows from

$$
\begin{aligned}
\frac{1}{2}\ln\mu \leq \left(\ln 2 - \frac{1}{2}\right)\lambda_{\mathrm{rel}} - \ln 2 - 5 \iff \\
\ln\mu \leq (2\ln 2 - 1)\lambda_{\mathrm{rel}} - 2\ln 2 - 10 \iff \\
\mu \leq \frac{e^{(2\ln 2-1)\lambda_{\mathrm{rel}}}}{e^{2\ln 2+10}} \iff \\
\mu \leq \frac{(\frac{4}{e})^{\lambda_{\mathrm{rel}}}}{4e^{10}}
\end{aligned}
$$

which is true by the assumption about μ.