

2024-05-26

Approximate lower bound arguments

L. Reyzin, P. Chaidos, A. Kiayias, A. Zinovyev. 2024. "Approximate Lower Bound Arguments"

<https://hdl.handle.net/2144/49988>

"Downloaded from OpenBU. Boston University's institutional repository."

Approximate Lower Bound Arguments

Pyrros Chaidos^{1,4}, Aggelos Kiayias^{2,4}, Leonid Reyzin^{*3}, and
Anatoliy Zinovyev³

¹National & Kapodistrian University of Athens

²University of Edinburgh

³Boston University

⁴IOG

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 Random 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 working together, 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 achieving universal composability in general-purpose succinct proof systems (SNARKs).

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.

^{*}Work done while visiting the Blockchain Technology Lab at the University of Edinburgh.

More formally, let R be a predicate. We explore succinct arguments of knowledge for a prover who knows a set S_p of values that satisfy R such that $|S_p| \geq n_p$ and wants to convince a verifier that $|S_p| > n_f$, where n_f is somewhat smaller than n_p . Because $n_f < n_p$, the verifier obtains a lower bound approximation to the actual cardinality of S_p ; hence we call this primitive an *Approximate Lower Bound Argument* or ALBA.

This problem has a long history. In 1983, in order to prove that $\text{BPP} \subseteq \text{RP}^{\text{NP}}$, 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, \dots, h_m [CW79] and the prover shows s, s' such that $s \neq s'$ and $h_i(s) = h_i(s')$ for some $i \in \{1, \dots, m\}$. If S is small (of size at most n_f), then such hash collisions are very unlikely to exist, and if S is big (of size at least n_p), 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 (see 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].

In designing ALBAs, we will aim to minimize communication and computational complexity; these metrics improve as the “gap” n_p/n_f increases. The proof size and verifier time in classical techniques above are far from optimal. While this does not affect the classical applications of ALBAs, which were theoretical (for example, the proof 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), it is an important concern for using ALBAs in practice.

1.1 Our Setting

The prover and verifier have access to a predicate R ; the prover possesses a set S_p whose elements satisfy R . The prover will show just a few elements of S_p to the verifier, which will convince the verifier that the prover possesses more than n_f elements that satisfy R . The goal is to find some property that is unlikely to hold for small sets S_f of size n_f , likely to hold for large sets S_p of size n_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 weight function W that takes a set element and outputs its nonnegative integer weight. 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 $\sum_{s \in S} W(s) > n_f$. When W is $\{0, 1\}$ -valued, we are in the setting of a predicate, and we call this case “unweighted.”

We emphasize that R or W 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 total weight exceeding n_f by simply observing the random oracle queries; in other words, the protocol is straight-line extractable in the nonprogrammable random oracle model. Our security is information-theoretic as long as the predicate R (or the weight function W) is independent of the random oracle; by the standard technique of adding a commitment to R (or W) to every random oracle query, we obtain computational security even if this function is adaptively chosen to depend on the oracle.

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 are non-adaptively secure — i.e., they require that the predicate R is independent of the URS or the verifier’s first message.

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

$$u = \frac{\lambda + \log \lambda}{\log \frac{n_p}{n_f}} \quad (1)$$

elements of S_p . Moreover, we show that this number is essentially tight, by proving that at least

$$u = \frac{\lambda}{\log \frac{n_p}{n_f}}$$

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 $\varepsilon \cdot n_f^u \leq 2^{-\lambda}$ for soundness, but $(1 - \varepsilon)^{n_p^u} \leq 2^{-\lambda}$ for completeness. A bit of algebra shows that $u = \frac{\lambda + \log \lambda}{\log(n_p/n_f)}$ suffices to satisfy both soundness and completeness constraints, so the proof is short.¹ However, in this scheme, the prover has to do an exhaustive search of $1/\varepsilon$ 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.

Its core idea 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 unfolding 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 $O(n_p \cdot \lambda^2)$. We then show how to drop further the prover complexity to $O(n_p + \lambda^2)$ by prehashing all elements and expressing the prefix invariant property as a hash collision. We also establish that our constructions are essentially optimal in terms of proof size by proving a lower bound in the number of elements than must be communicated by any ALBA scheme that satisfies the extractability requirements of Definition 4.

Weights and Decentralized Provers. In the case where all elements have an integer weight, 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, the prover’s running time becomes linear in the input’s total weight n_p which could be in the order of 2^{64} (number of coins in popular cryptocurrencies). 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 then proceed with the Telescope construction as above and with only a (poly)logarithmic penalty due to the weights. We detail this technique in Section 5.

Turning our attention to the decentralized setting we present two constructions. In the first one, each party performs a private random-oracle-based coin flip to decide whether to share her value. The aggregator produces a proof by concatenating

¹Let $\varepsilon = 2^{-\lambda} n_f^{-u}$ to satisfy soundness. Then $(1 - \varepsilon)^{n_p^u} < \exp(-2^{-\lambda} n_f^{-u})^{n_p^u} = \exp(-2^{-\lambda} (n_p/n_f)^u)$ is needed for completeness, so it suffices to have $\exp(-2^{-\lambda} (n_p/n_f)^u) \leq 2^{-\lambda}$, i.e., $2^{-\lambda} (n_p/n_f)^u \cdot \log e \geq \lambda$, i.e. $(n_p/n_f)^u \geq 2^\lambda \cdot \lambda / \log e$. Taking logarithm gives the desired result.

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 keys². Alternatively, if the knowledge extractor is allowed to rewind (need not be straight-line), one can use an arbitrary (not necessarily unique) signature scheme as follows: treat the public keys as set elements and for every selected public key in the ALBA proof, add its signature. 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

²The 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.

transform [BR93]). Compared to compact certificates, our Telescope 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, compact certificates cleverly tie the lottery to public keys rather than signatures and support an arbitrary signature scheme (not necessarily unique) while still providing straight-line knowledge extraction.

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 a 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 8.

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 scheme 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 the 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. 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,³ whereas

³This 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 $2^b = d2^{\lambda/r}$.

our Telescope construction from Section 3 requires only $d \cdot \lambda^{1/u} \cdot 2^{\lambda/u}$ of those.⁴ Thus, our approach speeds up this part of prover’s work by a factor of about u (which is close to the security parameter λ).

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. First, the proving time can become impractical when the number of set elements in the witness is large. Second, given that the weight function W must be encoded as a circuit, the proving cost also depends heavily on the complexity of W . Moreover, W cannot always be specified as a circuit, but is evaluated by a more complex process — 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 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] with straight-line extraction. To introduce arbitrary weights, we use a weight oracle $W : \{0, 1\}^* \rightarrow \mathbb{N} \cup \{0\}$ and denote for a set S , $W(S) = \sum_{s \in S} W(s)$.

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

- **Prove** ^{H, W} *is a probabilistic program that has access to the random oracle H and a weight oracle W ;*
- **Verify** ^{H, W} *is a program that has access to the random oracle H and a weight oracle W ;*
- **Extract** ^{H, W, \mathcal{A}} *is a probabilistic program that has access to the random oracle H , a weight oracle W and an adversary program \mathcal{A} ;*

⁴This value is obtained by setting $n_f = d$ and solving (1) for n_p .

- *completeness*: for all weight oracles W and all S_p such that $W(S_p) \geq n_p$, $\Pr[\text{Verify}^{H,W}(\text{Prove}^{H,W}(S_p)) = 1] \geq 1 - 2^{-\lambda_{\text{rel}}}$;
- *proof of knowledge*: consider the following experiment $\text{ExtractExp}(\mathcal{A}^{H,W}, W)$:
 $S_f \leftarrow \text{Extract}^{H,W,\mathcal{A}}()$;
output 1 iff $W(S_f) > n_f$;

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

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

moreover, $\text{Extract}^{H,W,\mathcal{A}}()$ is only allowed to run $\mathcal{A}^{H,W}$ once with the real H and W and only observes the transcript with its oracles (straight-line extraction property), Extract runs in time polynomial in the size of this transcript.

As presented, this definition is non-adaptive; i.e., it does not allow W to depend on H ; adaptivity can be added if it is possible to commit to W ; see Section 6 for further discussion.

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 $s \in S_p$ wish to act in coordination towards convincing the verifier. We now define a decentralized ALBA.

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

- $\text{Prove}^{H,W}$ is a probabilistic program that has access to the random oracle H and a weight oracle W ;
- $\text{Aggregate}^{H,W}$ is a probabilistic program that has access to the random oracle H and a weight oracle W ;
- $\text{Verify}^{H,W}$ is a program that has access to the random oracle H and a weight oracle W ;
- $\text{Extract}^{H,W,\mathcal{A}}$ is a probabilistic program that has access to the random oracle H , a weight oracle W and an adversary program \mathcal{A} ;
- *completeness*: consider the following experiment $\text{CompExp}(S_p, W)$:

$S := \emptyset$;
for $s \in S_p$ **do**
 $m \leftarrow \text{Prove}^{H,W}(s)$;
 if $m \neq \varepsilon$ **then** \triangleright if m is not empty string
 $S := S \cup \{m\}$;
 $\pi \leftarrow \text{Aggregate}^{H,W}(S)$;
 $r \leftarrow \text{Verify}^{H,W}(\pi)$;
return r ;

we require that for all weight oracles W and all S_p such that $W(S_p) \geq n_p$, $\Pr[\text{CompExp}(S_p, W) = 1] \geq 1 - 2^{-\lambda_{\text{rel}}}$;

- *proof of knowledge: consider the following experiment $\text{ExtractExp}(\mathcal{A}^{H,W}, W)$:*
 $S_f \leftarrow \text{Extract}^{H,W,\mathcal{A}}()$;
output 1 iff $W(S_f) > n_f$;

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

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

moreover, $\text{Extract}^{H,W,\mathcal{A}}()$ is only allowed to run $\mathcal{A}^{H,W}$ once with the real H and W and only observes the transcript with its oracles (straight-line extraction property), Extract runs in time polynomial in the size of this transcript.

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.”

Finally, we present a proof of knowledge ALBA definition in the CRS model. Unlike for NIROPK, the knowledge extractor here is allowed to rewind the adversary \mathcal{A} and is given it as regular input. Note that the definition crucially requires the CRS to be independent of W ; see Section 7 for further discussion.

Definition 3. (Prove, Verify, Extract, GenCRS) is a $(\lambda_{\text{sec}}, \lambda_{\text{rel}}, n_p, n_f)$ -CRS proof of knowledge ALBA scheme if and only if

- Prove^W is a probabilistic program;
- Verify^W is a program having access to a weight oracle W ;
- Extract^W is a probabilistic program having access to a weight oracle W ;
- GenCRS is a probabilistic program;
- *completeness: consider the following experiment $\text{CompExp}(W, S_p)$:*
 $\text{crs} \leftarrow \text{GenCRS}()$;
 $\pi \leftarrow \text{Prove}(\text{crs}, S_p)$;
 $r \leftarrow \text{Verify}^W(\text{crs}, \pi)$;
return r ;

we require that for all weight oracles W and all S_p such that $W(S_p) \geq n_p$, $\Pr[\text{CompExp}(W, S_p) = 1] \geq 1 - 2^{-\lambda_{\text{rel}}}$;

- *proof of knowledge: consider the following experiment $\text{SoundExp}(\mathcal{A}^W, W)$:*
 $\text{crs} \leftarrow \text{GenCRS}()$;
 $\pi \leftarrow \mathcal{A}^W(\text{crs})$;
 $r \leftarrow \text{Verify}^W(\text{crs}, \pi)$;

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 $\varepsilon = \Pr[\text{SoundExp}(\mathcal{A}^W, W) = 1] - 2^{-\lambda_{\text{sec}}} > 0$, then $S_f \leftarrow \text{Extract}^W(\mathcal{A})$ runs in expected time $\text{poly}(T, 1/\varepsilon)$ and $\Pr[W(S_f) > n_f] = 1$.

3 Telescope ALBA

In this section we present two ALBA schemes in sequence. 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 using a single random oracle which outputs binary strings in Appendix B. Further, we initially restrict weights to be either 0 or 1, and generalize to integers in Section 5. Finally, we postpone showing the proof of knowledge property and instead consider a simpler notion of soundness: given n_f elements fixed in advance, what is the probability that a valid proof exists containing only those elements? Sections 6 and 7 will then show how a knowledge extractor can be constructed.

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 $n_p d$ pairs with probability $1/n_p$. 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 $1/n_p$. 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 $H_1 \sim \text{Bernoulli}(1/n_p)$, $H_2 \sim \text{Bernoulli}(q)$ be random functions returning 1 with probability $1/n_p$ and q respectively, and returning 0 otherwise. Any tuple (t, s_1, \dots, s_u) such that

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

is a valid proof (see Section 3.3 how to implement H_1 efficiently).

Intuitively, this works because the honest prover maintains d tuples in expectation at each stage, while the malicious prover's tuples decrease n_p/n_f times with each stage. However, to implement and analyze the prover algorithm, it will be convenient to represent all tuples (t, s_1, \dots, s_i) , where $1 \leq t \leq d$, $0 \leq i \leq u$ and $s_1, \dots, s_i \in S_p$, as d trees of height u with $\{(1), \dots, (d)\}$ being the roots of the trees

and $\{(t, s_1, \dots, s_u)\}_{1 \leq t \leq d, s_1, \dots, s_u \in S_p}$ being the leaves. To implement Prove, simply run depth first search (DFS) to find a “valid” path from a root to a leaf.

```

procedure DFSH1,H2(Sp, t, s1, ..., sk)
  if k = u then
    if H2(t, s1, ..., su) = 1 then
      return (t, s1, ..., su)
    return ⊥
  for sk+1 ∈ Sp do
    if H1(t, s1, ..., sk+1) = 1 then
      π ← DFSH1,H2(Sp, t, s1, ..., sk+1);
      if π ≠ ⊥ then
        return π;
  return ⊥;
procedure ProveH1,H2(Sp)
  for t ∈ [d] do
    π ← DFSH1,H2(Sp, t);
    if π ≠ ⊥ then
      return π;
  return ⊥;
procedure VerifyH1,H2(t, s1, ..., su)
  if t ∉ [d] then
    return 0;
  for i ∈ [u] do
    if H1(t, s1, ..., si) ≠ 1 then
      return 0;
  return H2(t, s1, ..., su);

```

We will now analyze soundness of this construction. As mentioned above, the soundness error is defined to be the probability that a valid proof exists containing only elements from a fixed set S_f of size n_f .

Lemma 1. *The soundness error is at most $\left(\frac{n_f}{n_p}\right)^u \cdot qd$.*

Proof. By union bound, the probability that a valid proof can be constructed using n_f elements is at most

$$\left(\frac{1}{n_p}\right)^u \cdot q \cdot d \cdot n_f^u = \left(\frac{n_f}{n_p}\right)^u \cdot qd.$$

□

Theorem 1. *Let*

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

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

Proof. Follows from Lemma 1.

□

We now analyze completeness.

Lemma 2. *The probability that there does not exist a valid proof starting with a particular integer t is at most $\exp\left(-\left(q - u \cdot \frac{q^2}{2}\right)\right)$.*

Proof. We can make 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, \dots, s_{u-k} , there is no suffix of honest player's elements that works, meaning there is no $s_{u-k+1}, \dots, s_u \in S_p$ such that for all $u - k + 1 \leq i \leq u$, $H_1(t, s_1, \dots, s_i) = 1$, and $H_2(t, s_1, \dots, s_u) = 1$. Then one can see that

- $f(0) = 1 - q$;
- for $0 \leq k < u$, $f(k + 1) = \left(\left(1 - \frac{1}{n_p}\right) + \frac{1}{n_p} \cdot f(k)\right)^{n_p}$;
- the probability that there does not exist a valid proof with a particular integer t is $f(u)$;

This recursive formula can be approximated:

$$f(k + 1) = \left(1 + \frac{1}{n_p}(f(k) - 1)\right)^{n_p} \leq \left(e^{\frac{1}{n_p}(f(k)-1)}\right)^{n_p} = e^{f(k)-1}. \quad (2)$$

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,

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

Using the values for d and q , one can see that $k \cdot \frac{q^2}{2} \leq u \cdot \frac{q^2}{2} \leq q$, then

$$\begin{aligned} [\geq] 1 - \left(1 - \left(q - k \cdot \frac{q^2}{2}\right) + \frac{\left(q - k \cdot \frac{q^2}{2}\right)^2}{2}\right) &\geq \\ \left(q - k \cdot \frac{q^2}{2}\right) - \frac{q^2}{2} &= q - (k + 1) \cdot \frac{q^2}{2}. \end{aligned}$$

Hence, $-\ln f(u) \geq q - u \cdot \frac{q^2}{2}$ which proves the lemma. \square

Theorem 2. *Let*

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

Then completeness error is $\leq 2^{-\lambda_{rel}}$.

Proof. From Lemma 2, the probability that the honest prover fails is at most

$$\exp\left(-\left(q - u \cdot \frac{q^2}{2}\right)d\right).$$

Using the values for d and q , one can see that this is at most $2^{-\lambda_{rel}}$. \square

Corollary 1. *Let*

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

Then soundness error is $\leq 2^{-\lambda_{\text{sec}}}$ and completeness error is $\leq 2^{-\lambda_{\text{rel}}}$.

It is worth noting that the constant in d , and thus algorithm's running time, can be reduced. We show how to do this 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.

3.1.1 Running time

In this section we analyze the prover's running time, measured in terms of the number of invocations of the random (hash) functions.

Assume S_p is a set with cardinality n_p . As mentioned above, all tuples (j, s_1, \dots, s_i) can be represented as d trees. We would like to analyze the number of "accessible" vertices in these trees. Let the indicator random variable

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

If $A_{j,s_1,\dots,s_i} = 1$ we say the vertex (j, s_1, \dots, 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 3. *For any j and $0 \leq i \leq u$,*

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

We present the proof on page 57 of the Appendix.

Assuming the prover runs DFS, Theorem 2 gives a bound on the expected number of evaluated trees. And by the above theorem, the algorithm invokes H_1 $n_p u$ 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 $(n_p u + 1)$. This, however, needs a more careful proof.

Theorem 4. *Assume $uq < 2$. The expected number of hash evaluations is at most*

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

We present the proof on page 57 of the Appendix.

Taking parameter values from Corollary 1 and letting $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$ and $n_p/n_f = \text{const}$, we thus obtain an expected number of hash evaluations of $O(n_p \cdot \lambda^2)$.

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 non-root vertices in all d trees

$$Z = \sum_{\substack{1 \leq j \leq d, \\ 1 \leq i \leq u, \\ s_1, \dots, s_i \in S_p}} A_{j, s_1, \dots, s_i}.$$

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

Theorem 5. *For any $\delta \geq 0$,*

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

We present the proof on page 58 of the Appendix.

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

3.2 Construction with Prehashing

The basic scheme described above has prover expected running time $O(n_p \cdot \lambda^2)$, worst case running time $O(n_p \cdot \lambda^3)$ and verification time $O(\lambda)$ if we let $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$ and $n_p/n_f = \text{const}$. The modification described in this section has prover expected running time $n_p + O(\lambda^2)$, worst case running time $n_p + O(\lambda^3)$ and verification time is unchanged.

The improvement is inspired by balls-and-bins 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 $[n_p]$ and hash individual set elements to a uniform value in $[n_p]$, and consider a valid extension to be such that the tuple and the extension both hash to the same value. In the terminology of balls and bins, we treat the n_p individual elements as balls and put each of them randomly into one of the n_p bins as determined by the random function. Then, when trying to extend a partial tuple, we hash it to obtain the bin number and the permitted extensions will be exactly those in that bin.

More formally, we have random functions $H_0, H_1 \sim \text{Unif}([n_p])$ producing a uniformly random value in $[n_p]$ and hash function $H_2 \sim \text{Bernoulli}(q)$ returning 1 with probability q and 0 otherwise, and consider a tuple (t, s_1, \dots, s_u) a valid proof if and only if

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

(see Section 3.3 how to implement H_1 efficiently).

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. Below is the pseudocode implementation of the prover and verifier algorithms.

```

procedure DFSH0,H1,H2(bins, t, s1, ..., sk)
  if k = u then
    if H2(t, s1, ..., su) = 1 then
      return (t, s1, ..., su)
    return ⊥
  for sk+1 ∈ bins[H1(t, s1, ..., sk)] do
    π ← DFSH0,H1,H2(bins, t, s1, ..., sk+1);
    if π ≠ ⊥ then
      return π;
  return ⊥
procedure ProveH0,H1,H2(Sp)
  for i ∈ [np] do
    bins[i] ← ∅;
  for s ∈ Sp do
    bins[H0(s)] ← bins[H0(s)] ∪ {s};
  for t ∈ [d] do
    π ← DFSH0,H1,H2(bins, t);
    if π ≠ ⊥ then
      return π;
  return ⊥
procedure VerifyH0,H1,H2(t, s1, ..., su)
  if t ∉ [d] then
    return 0;
  for i ∈ [u] do
    if H1(t, s1, ..., si-1) ≠ H0(si) then
      return 0;
  return H2(t, s1, ..., su);

```

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 arrangement of balls to bins is well distributed, and thus another extension is likely to succeed. 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 “good”, then with high probability the honest player succeeds. The second part proves that we get a “good” distribution of balls with high probability. The “good” 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.

Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i , and let $c > 0$ be

some constant. The property we care about is the following:

$$\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq 1 - q + cq^2.$$

To show that it holds with good probability, we use Poisson approximation which lets us get rid of dependencies between different X_i and treat them as independent Poisson random variables with mean 1, which significantly simplifies the proof. After applying Poisson approximation, we use either Markov's inequality or custom tailored Chernoff analysis to bound the random sum. The Chernoff analysis lets us prove that the property holds with overwhelming probability but requires n_p , the number of balls, to be large enough (on the order of λ^3). If n_p is very small (smaller than λ^2), then we use the Markov approach to get completeness 1/2. If n_p is somewhere in between, we still use the Chernoff approach, but get completeness error that is only moderately small. In either of the two cases, completeness must be amplified as we explain later in Section 3.2.2. Looking ahead, we get average case running time $n_p + O(\lambda^2)$ and worst case running time $n_p + O(\lambda^3)$ regardless of n_p .

We first formally analyze soundness. As mentioned previously, we define soundness error to be the probability that a valid proof can be constructed using elements S_f with $|S_f| = n_f$ (simple soundness).

Lemma 3. *The soundness error is at most $\left(\frac{n_f}{n_p}\right)^u \cdot qd$.*

Proof. By union bound, the probability that a valid proof can be constructed using n_f elements is at most

$$\left(\frac{1}{n_p}\right)^u \cdot q \cdot d \cdot n_f^u = \left(\frac{n_f}{n_p}\right)^u \cdot qd.$$

□

Theorem 6. *Let*

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

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

Proof. Follows from Lemma 3. □

We now analyze completeness. The following lemma uses Markov's inequality to establish that the “good” arrangement of balls into bins holds with moderate probability. It will be useful later.

Lemma 4. *Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i , and let $c > 0$. Then*

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

We present the proof on page 59 of the Appendix.

The next lemma uses the Chernoff approach to analyze the same event. As one can notice, n_p needs to be large for it to be meaningful.

Lemma 5. *Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i . Then*

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

We present the proof on page 60 of the Appendix.

Finally, the following lemma establishes that as long as the “good” arrangement of balls holds, the honest prover succeeds with good probability.

Lemma 6. *Let $c > 0$, 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 $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+cq^2}$ and let F be the event that the honest prover fails. Then $\Pr[F|E] \leq e^{-(q-cuq^2)d}$.*

We present the proof on page 62 of the Appendix.

By combining the above two lemmas, we get sufficient Telescope parameter values to guarantee good completeness.

Theorem 7. *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 error is $\leq 2^{-\lambda_{rel}}$.

We present the proof on page 63 of the Appendix.

Corollary 2. *Assume*

$$u \geq \frac{\lambda_{sec} + \log(\lambda_{rel} + \log 3) + 1 - \log \log e}{\log \frac{n_p}{n_f}}; d \geq \frac{16u(\lambda_{rel} + \log 3)}{\log e};$$

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

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

Proof. Combine Theorems 6 and 7. □

The above corollary is ready to be used as is, provided that n_p is large. As we will see in Section 3.2.1, the average case running time of the prover in this scheme is $n_p + O(\lambda^2)$ and worst case running time is $n_p + O(\lambda^3)$. We explain how to handle the case when n_p is small in Section 3.2.2.

3.2.1 Running time

In this section we analyze the prover's running time. The results here will establish a bound on the average prover running time and a tight bound on the prover running time of the scheme with parameters in Corollary 2, as well as, serve as basis for more general running time analysis in Section 3.2.2. We measure the prover running time in terms of the number of invocations of the random (hash) functions.

Assume S_p is a set with cardinality n_p . As described in Section 3.1, all tuples (j, s_1, \dots, s_i) can be represented as d trees of height u . We would like to analyze the number of "accessible" vertices in these trees. Let the indicator random variable

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

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

Similarly to Section 3.1.1, one can prove that the expected number of accessible vertices in a single tree at a particular height is 1. This holds independently of the value of H_0 !

Theorem 8. *For any j and $0 \leq i \leq u$,*

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

Combining the expected number of accessible vertices in a single tree and a lower bound on the probability that a tree contains a valid proof, we can establish the following.

Lemma 7. *Let $c > 0$, assume $c u q < 1$, 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 $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+cq^2}$ and let V be the number of visited vertices by the (DFS) algorithm. Then*

$$\mathbb{E} [V|E] \leq \frac{2(u+1)}{q - cuq^2}.$$

We present the proof on page 63 of the Appendix.

Theorem 9. *Suppose $8uq \leq 1$. The expected number of visited vertices by the (DFS) algorithm is at most*

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

We present the proof on page 63 of the Appendix.

The above theorem lets us see that when taking parameters from Corollary 2 and letting $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$ and $n_p/n_f = \text{const}$, the scheme has average prover running time $n_p + O(\lambda^2)$. Note that n_p is outside of the big O since we prehash each of the n_p elements using H_0 exactly once.

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

$$Z = \sum_{\substack{1 \leq j \leq d, \\ 1 \leq i \leq u, \\ s_1, \dots, s_i \in S_p}} A_{j, s_1, \dots, s_i}$$

which serves as an upper bound on the DFS running time. A tight bound is useful for proving worst case running time, but it also lets us argue that an adversary cannot exploit an imperfect hash function or a PRF by making too many queries.

Below is a Chernoff style theorem in its general form. It features a variable w that needs to be large enough and that affects the final bound, but no requirement on u is imposed. It is useful for formally proving average and worst case prover running time in Section 3.2.2, but a more practical and better bound is given in Theorem 11.

For technical reasons, Theorem 10 works better with and Theorem 11 works only with large u . When u is small, an alternative way to prove a tight bound on the running time exists and is given in Lemma 8. It uses a very different approach and works well when u is not large. Another limitation that all Theorem 10, Theorem 11 and Lemma 8 have is that they only work well when n_p is large. When n_p is small, we simply use the expected running time analysis in Lemma 7 and apply Markov's inequality to get a bound on the running time. Section 3.2.2 demonstrates how to combine all these approaches.

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

Theorem 10. *Let $u, w, n_p \in \mathbb{N}$, $\lambda > 0$, $\lambda' = \frac{\lambda+2}{\log e}$ and assume*

$$\frac{8 \cdot w^2 \cdot (w+2) \cdot e^{\frac{w+1}{w}}}{e \cdot (w+2 - e^{1/w}) \cdot (w+1)!} \leq 2^{-\lambda}.$$

Also define

$$\delta = \left(\frac{w\lambda'}{d} + 1 \right) \cdot \exp \left(\frac{2uw\lambda'}{n_p} + \frac{7u}{w} \right).$$

Then

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

We present the proof on page 64 of the Appendix.

In the above theorem, δ is a, perhaps large, constant when letting $\lambda = \lambda_{\text{rel}}$ and using parameters from Corollary 2. When letting $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$ and $n_p/n_f = \text{const}$, one can see that the scheme with parameters in Corollary 2 does $n_p + O(\lambda^3)$ hashings with overwhelming probability. The following, however, is an optimized bound that should be used in practice.

Theorem 11. *Let $u, n_p \in \mathbb{N}$, $\lambda > 0$, $\lambda' = \frac{\lambda+2}{\log e}$ and assume*

$$\frac{24ud(u+2) \cdot e^{\frac{u+1}{u}}}{e \cdot \lambda' \cdot (u+2 - e^{1/u}) \cdot (u+1)!} \leq 2^{-\lambda}; \quad d \geq \frac{u\lambda'}{3}; \quad n_p \geq \frac{u^2\lambda'}{2}.$$

Also define

$$\delta = \left(\sqrt{\frac{3u\lambda'}{d}} + 1 \right) \cdot \exp \left(2 \left(1 + \sqrt{\frac{u\lambda'}{3d}} \right) u \sqrt{\frac{2\lambda'}{n_p}} + \sqrt{\frac{3u\lambda'}{d}} \right).$$

Then

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

We present the proof on page 65 of the Appendix.

As mentioned before, the tight bounds above are not suitable when u is small. To overcome this issue, the following bound is introduced that works well when u is not large. By combining the two, we get a bound that works well for any u .

Lemma 8. *Let $\lambda, c > 0$, $u \in \mathbb{N}$, 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 $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+cq^2}$, let V be the number of non-root vertices that the (DFS) algorithm visits, assume $cuq < 1$, and define*

$$B = \frac{u(\lambda + \log u)}{2 \log e} \left(\frac{1}{q} + \frac{1}{q - cuq^2} \right) + u.$$

Then $\Pr[V > B|E] \leq 2^{-\lambda}$.

We present the proof on page 66 of the Appendix.

3.2.2 Generalization to small n_p

In this section, we show our most advanced Telescope scheme, supporting both large and small values of n_p , and analyze its expected and worst case running times. The following useful theorems combine the running time analysis with the analysis of completeness from previous subsections to argue that the DFS search will succeed within a bounded number of steps. Looking ahead, our scheme will possess a **deterministic** worst case running time.

Theorem 12. *Assume*

$$d \geq (32 \ln 12)u; \quad q = \frac{2 \ln 12}{d}.$$

Then the (DFS) algorithm visits less than

$$\frac{8(u+1)d}{\ln 12}$$

vertices **and** finds a valid proof with probability at least $1/2$.

Proof. Apply Lemma 52 with $c_0 := 8$, $c_1 := 4$ and $\lambda := \ln 12$. □

Theorem 13. *Let $u, w, n_p \in \mathbb{N}$ and assume*

$$\lambda'_{rel} = \frac{\lambda_{rel} + 7}{\log e}; \quad d \geq 16u\lambda'_{rel}; \quad q = \frac{2\lambda'_{rel}}{d}; \quad n_p \geq \frac{d^2}{9\lambda'_{rel}};$$

$$\frac{14 \cdot w^2 \cdot (w + 2) \cdot e^{\frac{w+1}{w}}}{e \cdot (w + 2 - e^{1/w}) \cdot (w + 1)!} \leq 2^{-\lambda_{rel}}.$$

Then the (DFS) algorithm visits less than

$$\left(\frac{w\lambda'_{rel}}{d} + 1 \right) \cdot \exp \left(\frac{2uw\lambda'_{rel}}{n_p} + \frac{7u}{w} \right) \cdot du + d$$

*vertices **and** finds a valid proof with probability $\geq 1 - 2^{-\lambda_{rel}}$.*

Proof. Apply Theorem 7 with $\lambda_{rel} := \lambda_{rel} + \log \frac{7}{3}$ and Theorem 10 with $\lambda := \lambda_{rel} + \log \frac{7}{4}$. \square

Theorem 14. *Assume*

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

Then the (DFS) algorithm visits at most

$$\frac{\lambda_{rel} + 2 + \log u}{\lambda_{rel} + 2} \cdot \frac{3ud}{4} + d + u$$

*vertices **and** finds a valid proof with probability $\geq 1 - 2^{-\lambda_{rel}}$.*

We present the proof on page 67 of the Appendix.

We will now give an intuitive description of the new scheme and ideas behind it. For simplicity, assume $\lambda = \lambda_{sec} = \lambda_{rel}$ and $n_p/n_f = \text{const}$. The Telescope with Prehashing scheme with parameters in Corollary 2 have completeness and soundness $2^{-\lambda}$. Moreover, as shown in Section 3.2.1, the prover has expected running time $n_p + O(\lambda^2)$ and worst case running time $n_p + O(\lambda^3)$. The limitation, however, is that n_p needs to be large — at least λ^3 . One way to overcome it is to use Markov analysis of the “good” arrangement of balls into bins (Lemma 4) to achieve completeness $1/2$, which works for any n_p . We must then amplify completeness. We basically allow the prover to make multiple attempts, at the expense of worsened soundness. Therefore, we add another integer index v to the proof and restrict it to be between 1 and r . In general terms, in order to reduce completeness error $1/2$ to $2^{-\lambda_{rel}}$, we set $r := \lambda_{rel}$, and to compensate for the loss in soundness, we set $\lambda_{sec} := \lambda_{sec} + \log \lambda_{rel}$. Overall, the proof size u is not changed, except for an additive small constant.

Formally, the new proof object looks as follows. We have random functions $H_0, H_1 \sim \text{Unif}([n_p])$ producing a uniformly random value in $[n_p]$ and hash function $H_2 \sim \text{Bernoulli}(q)$ returning 1 with probability q and 0 otherwise. A tuple (v, t, s_1, \dots, s_u) is a valid proof if and only if

- $1 \leq v \leq r$;

n_p	$\leq \lambda^2$		$\lambda^{2+\varepsilon}, 0 < \varepsilon < 1$		$\geq \lambda^3$	
DFS completeness error	1/2	Theorem 12	$2^{-\lambda^\varepsilon}$	Theorem 13, 14, or 7 with 11	$2^{-\lambda}$	Theorem 14
DFS tight running time bound	$B = O(\lambda^2)$		$B = O(\lambda^{2+\varepsilon})$		$B = O(\lambda^3)$	
DFS expected running time	$O(\lambda^2)$		$O(\lambda^2)$	Theorem 9	$O(\lambda^2)$	Theorem 9
DFS bound	$B = O(\lambda^2)$		$B = O(\lambda^{2+\varepsilon})$		$B = O(\lambda^3)$	
Max. # of prove repetitions r	λ		$\lambda^{1-\varepsilon}$		1	
Expected # of prove repetitions	2		$1/(1 - 2^{-\lambda^\varepsilon})$		1	
Total expected running time			$n_p + O(\lambda^2)$			
Total worst case running time			$n_p + O(\lambda^3)$			

Figure 1: Optimal parameters for different n_p

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

(see Section 3.3 how to implement H_1 efficiently).

Theorem 12 combined with Theorem 6 shows that an attempt to find a proof under a single index v succeeds with probability $1/2$ within $O(\lambda^2)$ DFS steps. We restrict the DFS from running longer than that (parameter B in the pseudocode below) because if the arrangement of balls into bins happens to be bad, DFS might run for a long time ruining the worst case running time of the prover; we must stop it from doing so. As a consequence, we also get *deterministic* worst case running time. Since in expectation only two indices are checked before a valid proof is found, we get average prover running time of the resulting scheme $2n_p + O(\lambda^2)$ and worst case prover running time $O((n_p + \lambda^2) \cdot r) = O((n_p + \lambda^2) \cdot \lambda) = O(n_p \lambda + \lambda^3)$. While this might already suffice for many applications, for best efficiency we set our goal to get average prover running time $n_p + O(\lambda^2)$ and worst case prover running time $n_p + O(\lambda^3)$. For large n_p , the the expected time is improved by a factor of 2 and the worst case time is improved by a factor of λ !

Since we have already accomplished this when $n_p \geq \lambda^3$, we only need to consider the case where $\lambda^2 < n_p < \lambda^3$ (ignoring the constants). Consider, for example, $n_p = \lambda^{2+1/3}$. The Chernoff analysis (Corollary 2) cannot give us completeness error $2^{-\lambda}$, but setting $\lambda_{\text{rel}} := \lambda^{1/3}$, it can establish completeness error of $2^{-\lambda^{1/3}}$ when setting $d := \lambda^{1+1/3}$. Similarly, Theorem 10 shows that the DFS runs longer than $O(\lambda^{2+1/3})$ steps with probability at most $2^{-\lambda^{1/3}}$. Combining the two facts, conveniently formalized in Theorem 13, we know that the DFS fails to find a valid proof within $O(\lambda^{2+1/3})$ steps with probability at most $2^{-\lambda^{1/3}}$. We restrict the DFS to run for at most that number of steps (parameter B in the pseudocode below) and allow the prover to make $\lambda^{2/3}$ attempts to get average prover running time $O(n_p + \lambda^2)$ and worst case prover running time $O((n_p + \lambda^{2+1/3}) \cdot \lambda^{2/3}) = O(\lambda^{2+1/3} \cdot \lambda^{2/3}) = O(\lambda^3) = n_p + O(\lambda^3)$. With more careful calculation, one can also prove the average prover running time of $n_p + O(\lambda^2)$. We include optimal parameters and properties of the scheme for all values of n_p in Figure 1.

Below we present the full pseudocode implementation of the prover and verifier.

procedure BoundedDFS^{H₀,H₁,H₂}(bins, v, t, s_1, \dots, s_k , limit)

```

if  $k = u$  then
  if  $H_2(v, t, s_1, \dots, s_u) = 1$  then
    return  $(v, t, s_1, \dots, s_u)$ ;
  return  $\perp$ 
for  $s_{k+1} \in \text{bins}[H_1(v, t, s_1, \dots, s_k)]$  do
  if  $*\text{limit} = 0$  then
    return  $\perp$ ;
     $*\text{limit} \leftarrow *\text{limit} - 1$ ;
     $\pi \leftarrow \text{BoundedDFS}^{H_0, H_1, H_2}(\text{bins}, v, t, s_1, \dots, s_{k+1})$ ;
    if  $\pi \neq \perp$  then
      return  $\pi$ ;
return  $\perp$ 

```

procedure ProvelIndex^{H₀,H₁,H₂}(S_p, v)

```

for  $i \in [n_p]$  do
   $\text{bins}[i] \leftarrow \emptyset$ ;
for  $s \in S_p$  do
   $\text{bins}[H_0(v, s)] \leftarrow \text{bins}[H_0(v, s)] \cup \{s\}$ ;
limit  $\leftarrow B$ ;
for  $t \in [d]$  do
  if limit = 0 then
    return  $\perp$ ;
    limit  $\leftarrow$  limit - 1;
     $\pi \leftarrow \text{BoundedDFS}^{H_0, H_1, H_2}(\text{bins}, v, t, \&\text{limit})$ ;
    if  $\pi \neq \perp$  then
      return  $\pi$ ;
return  $\perp$ ;

```

procedure Prove^{H₀,H₁,H₂}(S_p)

```

for  $v \in [r]$  do
   $\pi \leftarrow \text{ProvelIndex}^{H_0, H_1, H_2}(S_p, v)$ ;
  if  $\pi \neq \perp$  then
    return  $\pi$ ;
return  $\perp$ ;

```

procedure Verify^{H₀,H₁,H₂}(v, t, s_1, \dots, s_u)

```

if  $v \notin [r]$  then
  return 0;
if  $t \notin [d]$  then
  return 0;
for  $i \in [u]$  do
  if  $H_1(v, t, s_1, \dots, s_{i-1}) \neq H_0(v, s_i)$  then
    return 0;
return  $H_2(v, t, s_1, \dots, s_u)$ ;

```

We finally present the main result of this section. The formal proof is quite cumbersome but we hope that the informal discussion above explains well how parameters are chosen to support all values of n_p while minimizing the average and

worst case prover running time.

Corollary 3. *For all $\lambda_{sec} \geq 0$, $\lambda_{rel} \geq 1$ and $n_p > n_f \geq 1$, there is an ALBA scheme with soundness error $\leq 2^{-\lambda_{sec}}$, completeness error $\leq 2^{-\lambda_{rel}}$, proof size*

$$u = \left\lceil \frac{\lambda_{sec} + \log \lambda_{rel} + 5 - \log \log e}{\log \frac{n_p}{n_f}} \right\rceil,$$

expected prover running time

$$n_p + O(u^2)$$

and worst case prover running time

$$n_p + O(u^2 \cdot \lambda_{rel}).$$

We present the proof on page 69 of the Appendix.

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 $H_1(j)$, $H_1(j, s_1)$, $H_1(j, s_1, s_2)$, $H_1(j, s_1, s_2, \dots, s_u)$, 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 $H_1(j, s_1, \dots, s_{i+1})$ to reuse most of the computation of $H_1(j, s_1, \dots, s_i)$. 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 Appendix B for how we implement g). Inductively define $H'_1(j, s_1, \dots, s_{i+1}) = f(H'_1(j, s_1, \dots, s_i), s_{i+1})$ 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 5 and 11).

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** (see the definition below) 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 also note that the following definition can be used for upper bound results too, as demonstrated in Section 7 for the CRS model.

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

- **Prove^H** is a probabilistic random oracle access program;
- **Verify^H** is a random oracle access program;
- **Read** is a program;
- *completeness: consider the following experiment $CompExp(S_p)$:*
 $\pi \leftarrow \text{Prove}^H(S_p)$;
output 1 iff $\text{Read}(\pi) \subseteq S_p$ and $\text{Verify}^H(\pi) = 1$;
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 $\exists \pi, \text{Read}(\pi) \subseteq S_f \wedge \text{Verify}^H(\pi) = 1$;
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 $\alpha = \frac{\lambda_{sec} - 3}{\log(n_p/n_f)}$, assume $n_f \geq 3\alpha^2$, let S_p be an arbitrary set of size n_p , and let **(Prove, Read, Verify)** be a $(\lambda_{sec}, \lambda_{rel}, n_p, n_f)$ -ALBA scheme. Then*

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

We present the proof on page 75 of the Appendix.

4 ALBAs with Decentralized Prover

In the previous section we assumed 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 still use weights of 0 or 1, they can be generalized to integer weights as explained in Section 5. Finally, as in Section 3, instead of proof of knowledge we consider a simpler notion of soundness: the probability that a valid proof exists containing only elements from set S_f of size n_f . Sections 6 and 7 demonstrate how to do knowledge extraction.

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 $p = \frac{\mu}{n_p}$ and 0 otherwise. Each set element s is sent to the aggregator over the network if and only if $H(s) = 1$. Now let $r_s, r_c > 1$ such that $r_s r_c = \frac{n_p}{n_f}$ and set $u = r_s \cdot p n_f$ (or equivalently $u = \frac{p n_p}{r_c}$). 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 9. *Assuming*

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

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

We present the proof on page 77 of the Appendix.

Lemma 10. *Assuming*

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

completeness error of the scheme is $\leq 2^{-\lambda_{rel}}$.

We present the proof on page 77 of the Appendix.

Thus, to minimize u , we need to minimize

$$\max \left\{ \frac{\lambda_{sec} \cdot \ln 2}{\ln r_s - 1 + \frac{1}{r_s}}, \frac{\lambda_{rel} \cdot \ln 2}{r_c - 1 - \ln r_c} \right\}.$$

Noting 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 $\lambda_{sec} = \lambda_{rel} = \lambda$, then setting $r_c = \frac{n_p}{n_p - n_f} \cdot \ln \frac{n_p}{n_f}$ and $r_s = \frac{n_p - n_f}{n_f} \cdot \frac{1}{\ln \frac{n_p}{n_f}}$ gives the smallest u .

We note the interesting fact that choosing r_s and r_c that minimize u also minimizes μ . Since $\mu = p n_p = u r_c$, we have

$$\mu \geq \max \left\{ \frac{\lambda_{sec} \cdot \ln 2}{\ln r_s - 1 + \frac{1}{r_s}} \cdot r_c, \frac{\lambda_{rel} \cdot \ln 2}{r_c - 1 - \ln r_c} \cdot r_c \right\}.$$

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, and calculate an appropriate increase to the security and reliability parameters. While combining a simple lottery with an ALBA aggregator is a generic technique, but the generic analysis requires one to calculate two lottery tail bounds: one for soundness and one for completeness. By using Telescope for the aggregator, we benefit from omitting the soundness tail bounds from analysis; this section has all details.

As previously, we have parameter μ and select each element to be transmitted over the network with probability μ/n_p . After receiving enough elements selected by the simple lottery, the aggregator runs the algorithm from Section 3.2.

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 $1 - 2^{-\lambda_{\text{rel}}-1}$ and 2) the Telescope aggregator will produce a valid certificate with probability $1 - 2^{-\lambda_{\text{rel}}-1}$. For the aggregator, we use the Telescope scheme with parameters from Theorem 12 and $\lambda_{\text{rel}} + 1$ repetitions as described in Section 3.2.2, yielding expected aggregator running time $O(\mu + u^2)$ and worst case running time $O((\mu + u^2)\lambda_{\text{rel}})$; the worst case running time can be improved following the approach in Corollary 3.

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

$$\frac{\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}}$$

(we have $\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12$ instead of $\lambda_{\text{sec}} + \log \lambda_{\text{rel}} + 1 + \log e + \log \ln 12$ in Theorem 12 because we instantiate it with $\lambda_{\text{sec}} := \lambda_{\text{sec}} + \log e$ and $\lambda_{\text{rel}} := \lambda_{\text{rel}} + 1$ for technical reasons).

One can think of the gap $\frac{\rho n_p}{\mu n_f}$ as $\frac{(1-\delta)n_p}{n_f}$ if we set $\rho = (1-\delta)\mu$, and a formula for δ can be derived using a Chernoff bound. Note that we only decrease n_p in the $\frac{n_p}{n_f}$ gap. n_f remains the same since the union bound argument for soundness still works, but with some modifications. Particularly, it requires a somewhat large μ .

Let Lottery : $\{0, 1\}^* \rightarrow \{0, 1\}$ be an oracle returning 1 with probability $\frac{\mu}{n_p}$ and assume $H = (H_0, H_1, H_2, \text{Lottery})$ where H_0, H_1, H_2 are as defined in Section 3.2. Also let $A.\text{Prove}^H, A.\text{Verify}^H$ be as in Section 3.2.2 and define the following.

<pre> procedure $B.\text{Prove}^H(s)$ if Lottery(s) = 1 then return s; else return empty string; </pre>	<pre> procedure $B.\text{Aggregate}^H(S)$ return $A.\text{Prove}^H(S)$; procedure $B.\text{Verify}^H(\pi)$ parse (t, s_1, \dots, s_u) = π; return 1 iff $A.\text{Verify}^H(\pi) = 1 \wedge$ $\forall 1 \leq i \leq u, \text{Lottery}(s_i) = 1$; </pre>
--	---

Theorem 16. *Assume*

$$\mu > \frac{2(\lambda_{rel} + 1)}{\log e}; \quad \delta = \sqrt{\frac{2(\lambda_{rel} + 1)}{\mu \log e}}; \quad \rho = \lceil (1 - \delta)\mu \rceil$$

and instantiate the algorithm in Section 3.2.2 with $r := \lambda_{rel} + 1$, $d \geq (32 \ln 12)u$, $q := \frac{2 \ln 12}{d}$ and $n_p := \rho$. Then completeness error is $\leq 2^{-\lambda_{rel}}$.

Proof. By Chernoff bound (Lemma 40), the simple lottery chooses at least $\rho > 0$ set elements with probability at least $1 - 2^{-\lambda_{rel}-1}$. Given this event, by Theorem 12, the algorithm outputs a valid certificate with probability at least $1 - 2^{-\lambda_{rel}-1}$. Therefore, completeness error is $\leq 2^{-\lambda_{rel}}$. \square

We now calculate soundness error defined as the probability that a valid proof can be constructed using elements S_f with $|S_f| = n_f$.

Lemma 11. *The soundness error is at most*

$$qdr \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u^2 n_p}{\mu n_f} \right).$$

We present the proof on page 78 of the Appendix.

We also include an additional improved soundness bound in Section C.2 that we use to calculate actual numbers.

Theorem 17. *Assume*

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

$$u \geq \frac{\lambda_{sec} + \log(\lambda_{rel} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}}.$$

and instantiate the algorithm in Section 3.2.2 with $r := \lambda_{rel} + 1$, $d \geq (32 \ln 12)u$, $q := \frac{2 \ln 12}{d}$ and $n_p := \rho$. Then soundness error is $\leq 2^{-\lambda_{sec}}$.

We present the proof on page 78 of the Appendix.

Using Theorems 16 and 17, we can see how big μ needs to be if we increase $u \log \frac{n_p}{n_f}$ only by some amount C .

Corollary 4. *Assume*

$$C > 0; \quad u \geq \frac{\lambda_{sec} + \log(\lambda_{rel} + 1) + 1 + \log e + \log \ln 12 + C}{\log \frac{n_p}{n_f}};$$

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

$$\delta = \sqrt{\frac{2(\lambda_{rel} + 1)}{\mu \log e}}; \quad \rho = \lceil (1 - \delta)\mu \rceil$$

and instantiate the algorithm in Section 3.2.2 with $r := \lambda_{rel} + 1$, $d \geq (32 \ln 12)u$, $q := \frac{2 \ln 12}{d}$ and $n_p := \rho$. Then soundness error is $\leq 2^{-\lambda_{sec}}$ and completeness error is $\leq 2^{-\lambda_{rel}}$.

We present the proof on page 79 of the Appendix.

Thus, if we let $\lambda = \lambda_{\text{sec}} = \lambda_{\text{rel}}$ and let u only be a constant larger than optimal, we have $\mu = O(\lambda^3)$; moreover, μ is proportional to $\frac{1}{C^2}$. Additionally, setting $C := \sqrt{\lambda}$, we get a slightly larger proof size u with communication complexity $O(\lambda^2)$. One could also amplify the completeness via repetitions as described in Section 3.2.2, not only on the aggregator side, but applied to the lottery as well. This can improve the proof size - communication tradeoff, but it requires some network engineering to avoid redundant communication. Specifically, one needs to delay lottery repetitions until the previous ones have probably failed.

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

Corollary 5. *Assume*

$$C > 0;$$

$$u \geq \left(1 + \frac{3\sqrt{2\log e} \cdot \sqrt{\lambda_{\text{rel}} + 1}}{\sqrt{\mu} \cdot \log \frac{n_p}{n_f}}\right) \cdot \frac{\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_p}{n_f}};$$

$$\mu \geq \max \left\{ \frac{8(\lambda_{\text{rel}} + 1)}{\log e}, \frac{n_p u^2}{n_f}, \frac{18 \log e \cdot (\lambda_{\text{rel}} + 1)}{\log^2 \frac{n_p}{n_f}} \right\};$$

$$\delta = \sqrt{\frac{2(\lambda_{\text{rel}} + 1)}{\mu \log e}}; \quad \rho = \lceil (1 - \delta)\mu \rceil$$

and instantiate the algorithm in Section 3.2.2 with $r := \lambda_{\text{rel}} + 1$, $d \geq (32 \ln 12)u$, $q := \frac{2 \ln 12}{d}$ and $n_p := \rho$. Then soundness error is $\leq 2^{-\lambda_{\text{sec}}}$ and completeness error is $\leq 2^{-\lambda_{\text{rel}}}$.

We present the proof on page 80 of the Appendix.

4.3 Optimality of the certificate size - communication trade-off

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

- Prove^H is a probabilistic random oracle access program;
- Verify^H is a random oracle access program;
- Read is a program;
- if L is a random 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 $\text{CompExp}(S_p)$:

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

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

we require that for all sets S_p with size $\geq n_p$, $\Pr[\text{CompExp}(S_p) = 1] \geq 1 - 2^{-\lambda_{rel}}.$

– soundness: consider the following experiment $\text{SoundExp}(S_f):$

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

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

The following theorem presents our lower bound.

Theorem 18. Assume ρ satisfies $\Pr[B(n_p, \frac{\mu}{n_p}) \leq \rho] \geq 2^{-\lambda_{rel}+1}$ 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 $(\text{Prove}, \text{Read}, \text{Verify})$ be a $(\lambda_{sec}, \lambda_{rel}, n_p, n_f, \mu)$ -lottery based ALBA scheme such that

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

Then

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

We present the proof on page 81 of the Appendix.

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

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

$$n_f \geq 2\mu.$$

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

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

We present the proof on page 84 of the Appendix.

Alternatively, we also present a corollary showing a lower bound on the certificate size as a function of μ . Compare it 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

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

$$n_f \geq 2\mu.$$

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

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

We present the proof on page 86 of the Appendix.

5 Adding Weights

We will assume, without loss of generality, that the weight function W outputs integers. A naive way to handle weights other than 0 and 1 is to interpret each set element s as $W(s)$ elements $(s, 1), \dots, (s, W(s))$ and apply schemes designed for the unweighted case to (s, i) pairs. Unfortunately, this approach makes the prover running time linear in the total weight which could be in the order of 2^{64} .

Fortunately, 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 $(s, 1), \dots, (s, W(s))$ pairs as $W(s)$ 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 $\text{Binom}(W(s), p)$ (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 $(s, 1), \dots, (s, k)$ 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 element.

Since the Decentralized Telescope scheme remains unchanged when weights are introduced, we now focus on constructing a weighted Telescope scheme with a single centralized prover as in Section 3.

Using Corollary 4 with $\lambda_{sec} := \lambda_{sec} + \log \lceil \lambda_{rel} \rceil$, $\lambda_{rel} := 1$, $C := 1$ and allowing the prover to make $\lceil \lambda_{rel} \rceil$ attempts as described in Section 3.2.2, we get a weighted

$(\lambda_{\text{sec}}, \lambda_{\text{rel}}, n_{\text{p}}, n_{\text{f}})$ -ALBA scheme with proof size

$$\frac{\lambda_{\text{sec}} + \log \lceil \lambda_{\text{rel}} \rceil + 3 + \log e + \log \ln 12}{\log \frac{n_{\text{p}}}{n_{\text{f}}}} \quad (3)$$

and expected prover running time $O(n + \lambda_{\text{sec}}^2)$, where n is the number of weighted elements in the input.

One can take the approach in Section 3.2.2 to also minimize the worst case prover running time, but in addition to carefully choosing the parameters d , q , r and the DFS bound B , one also needs to choose the optimal μ . The key difference, though, is that while in the unweighted case the size of prover's input n_{p} is fixed and known in advance, in the weighted case the size of prover's input n is only known at runtime, since elements can have large or small weight. A solution is to let the prover choose the appropriate d , q , r and B dynamically based on the size of its input n . The number of sets of parameters (d_i, q_i, r_i, B_i) should not be large to not affect soundness too much. We estimate that by making $\approx \log \lambda_{\text{rel}}$ sets of parameters where r_i are powers of two between 1 and λ_{rel} , one can construct a scheme with proof size

$$u = \frac{\lambda_{\text{sec}} + \log \lambda_{\text{rel}} + \log \log \lambda_{\text{rel}} + C}{\log \frac{n_{\text{p}}}{n_{\text{f}}}}$$

where C is a small constant, expected prover running time

$$n + O(u^2)$$

and worst case prover running time

$$n + O(u^2 \cdot \lambda_{\text{rel}}).$$

The additional additive $\log \log \lambda_{\text{rel}}$ factor in u comes from compensating the small loss in soundness.

6 Knowledge Extraction for NIROPK

In this section we show how Definitions 1 and 2 can be realized. While Sections 3 and 4 provide intuitive constructions with clean combinatorial analysis, they have a missing piece — a knowledge extractor. As we will see, the simple soundness proven there does not immediately imply proof of knowledge, and more reasoning is needed. We also remind that the knowledge extractor must be straight-line; i.e., rewinding of the prover is not allowed but observing its queries to the oracles is. Here we describe the full NIROPK scheme including its knowledge extractor for the case of the basic Telescope construction from Section 3.1 while other Telescope constructions can be made NIROPK in a similar fashion. For $H = (H_1, H_2)$, define

<pre> procedure Prove^{H,W}(S_p) run DFS as described in Section 3.1 procedure Verify^{H,W}(π) parse (t, s₁, ..., s_u) = π return 1 iff • 1 ≤ t ≤ d; • ∀1 ≤ i ≤ u, H₁(t, s₁, ..., s_i) = 1; • H₂(t, s₁, ..., s_u) = 1; • ∀1 ≤ i ≤ u, W(s_i) = 1 </pre>	<pre> procedure Extract^{H,W,A} function A₁^{H,W} π ← A^{H,W}(); v ← Verify^{H,W}(π); return π; run A₁^{H,W}() and observe its oracles transcript τ; S_f := ∅; for x queried to H₁ or H₂ in τ do if W(x) = 1 then add x to S_f; return S_f. </pre>
--	--

Theorem 19. *Define parameters as in Theorem 1. Then algorithms Verify^{H,W} with Extract^{H,W,A} satisfy the proof of knowledge property of Definition 1.*

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 at most $2^{-\lambda_{\text{sec}}}$ by the following lemma. Thus, the proof of knowledge property follows by the union bound. \square

See full proof on page 87.

The following lemma resembles the simple soundness result in Theorem 1, but unfortunately is harder to prove. Whereas the proof of Theorem 1 is a simple application of union bound, the fact that the adversary can choose what weight-1 elements to query adaptively based on past RO responses makes the “vanilla” union bound argument inapplicable. Fortunately, there exists a way around this problem.

Lemma 12. *Define parameters as in Theorem 1 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 $\Pr[E] \leq 2^{-\lambda_{\text{sec}}}$.*

We present the proof on page 88 of the Appendix.

Combining the proof of knowledge property with completeness proven in Section 3.1, we now state the main result of this section.

Corollary 8. *Using parameters from Corollary 1, (Prove, Verify, Extract) is a $(\lambda_{\text{sec}}, \lambda_{\text{rel}}, n_p, n_f)$ -NIROPK ALBA scheme.*

In summary, we achieve information-theoretic but non-adaptive security; i.e., additional computational power does not help the adversary avoid knowledge extraction but he is not allowed to choose the predicate / weight function based on the random oracle. Adaptive security can be achieved the traditional way: rerandomize the random oracle by including a commitment to the weight function as additional input to the random oracle. However, the security downgrades to computational: assuming that adversary makes at most 2^q RO queries, we need to increase ALBA’s λ_{sec} parameter by q .

7 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). This is a novel feature of our scheme in comparison to compact certificates which inherently rely on the random oracle because of Fiat-Shamir. 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-theoretic security.

Assume (GenKey, F) is a PRF such that for any oracle access program \mathcal{A}^O with running time bounded by T ,

$$\left| \Pr [\mathcal{A}^H() = 1] - \Pr [\mathcal{A}^{F(\text{GenKey}(\cdot, \cdot))}() = 1] \right| \leq \varepsilon_{\text{prf}}(T). \quad (4)$$

We will assume the unweighted case, but the following can be 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 11) and instantiating the scheme with the standard random oracle (Appendix B), one can build a Telescope scheme such that for some $B \in O(\lambda^3)$,

- the honest prover's DFS visits at most B vertices and outputs a valid proof 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 $\text{Prove}^H(S_p)$ as the standard DFS that visits at most B vertices and define $\text{Verify}^H(\pi)$ in a natural way. We show an ALBA scheme under Definition 4 where the random oracle is replaced with CRS. Below is the new definition and a Telescope construction for it.

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

- *Prove* is a probabilistic program;
- *Verify* is a program;
- *Read* is a program;
- *GenCRS* is a probabilistic program;
- *completeness*: consider the following experiment $\text{CompExp}(S_p)$:
 $\text{crs} \leftarrow \text{GenCRS}()$;
 $\pi \leftarrow \text{Prove}(\text{crs}, S_p)$;
output 1 iff $\text{Read}(\pi) \subseteq S_p$ and $\text{Verify}(\text{crs}, \pi) = 1$;

we require that for all sets S_p with size $\geq n_p$, $\Pr[\text{CompExp}(S_p) = 1] \geq 1 - 2^{-\lambda_{\text{rel}}}$.

- *soundness*: consider the following experiment $\text{SoundExp}(S_f)$:

$\text{crs} \leftarrow \text{GenCRS}();$

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

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

<pre> procedure $R.\text{Prove}(\text{crs}, S_p)$ ┌ $\pi \leftarrow \text{Prove}^{F(\text{crs}, \cdot)}(S_p);$ └ return $\pi;$ procedure $R.\text{Verify}(\text{crs}, \pi)$ ┌ $r \leftarrow \text{Verify}^{F(\text{crs}, \cdot)}(\pi);$ └ return $r;$ </pre>	<pre> procedure $R.\text{Read}(\pi)$ ┌ parse $(t, s_1, \dots, s_u) = \pi;$ └ return $\{s_1, \dots, s_u\};$ procedure $R.\text{GenCRS}$ ┌ $k \leftarrow \text{GenKey}();$ └ return $k;$ </pre>
--	---

Theorem 20. R is a $(\lambda'_{\text{sec}}, \lambda'_{\text{rel}}, n_p, n_f)$ -CRS ALBA scheme where $\lambda'_{\text{sec}} = \lambda'_{\text{rel}} = -\log(2^{-\lambda} + \varepsilon_{\text{prf}}(O(n_p + \lambda^3)))$.

Proof. Completeness follows from the fact that Prove 's running time is bounded by $O(n_p + B) = O(n_p + \lambda^3)$ steps and that $\text{Prove}^H(S_p)$, 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 $\text{Prove}^{F(\text{GenKey}(), \cdot)}$ outputs a valid proof with probability $\geq 1 - 2^{-\lambda} - \varepsilon_{\text{prf}}(O(n_p + \lambda^3))$.

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 $\leq 2^{-\lambda} + \varepsilon_{\text{prf}}(O(n_p + \lambda^3))$. But the probability that there *exists* a valid proof in the PRF case cannot be larger. \square

We present the full version of the proof in Section C.3.

7.1 Knowledge Extraction for Definition 6 / Definition 4

In this section we show how to generically convert an ALBA scheme under Definition 6 to a proof of knowledge scheme under Definition 3. We still assume the unweighted scenario ($W : \{0, 1\}^* \rightarrow \{0, 1\}$) but the following can be generalized to add weights. Sometimes it will be convenient to treat W as a set: $\{s : W(s) = 1\}$.

Let $X = (X.\text{Prove}, X.\text{Read}, X.\text{Verify}, X.\text{GenCRS})$ be a $(\lambda_{\text{sec}}, \lambda_{\text{rel}}, n_p, n_f)$ -CRS ALBA scheme (as in Definition 6) and define $Y = (Y.\text{Prove}, Y.\text{Verify}, Y.\text{Extract}, Y.\text{GenCRS})$ as follows.

<pre> procedure $Y.\text{GenCRS}$ ┌ return $X.\text{GenCRS}();$ procedure $Y.\text{Prove}^W(\text{crs}, S_p)$ ┌ return $X.\text{Prove}(\text{crs}, S_p \cap W);$ procedure $Y.\text{Verify}^W(\text{crs}, \pi)$ ┌ $S := X.\text{Read}(\pi);$ └ return 1 iff $S \subseteq W \wedge$ └ $X.\text{Verify}(\text{crs}, \pi) = 1;$ </pre>	<pre> procedure $Y.\text{Extract}^W(\mathcal{A})$ ┌ $S_f := \emptyset;$ └ while $S_f \leq n_f$ do └ ┌ $\text{crs} \leftarrow X.\text{GenCRS}();$ └ └ $\pi \leftarrow \mathcal{A}^W(\text{crs});$ └ └ $S := X.\text{Read}(\pi);$ └ └ $S_f := S_f \cup (S \cap W);$ └ return $S_f;$ </pre>
---	--

Theorem 21. Y is $(\lambda_{\text{sec}}, \lambda_{\text{rel}}, n_p, n_f)$ -CRS proof of knowledge ALBA scheme.

Proof. It is easy to see that Y satisfies the completeness property. We are left to prove the proof of knowledge property.

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

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

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

$$\begin{aligned} \varepsilon &= (2^{-\lambda_{\text{sec}}} + \varepsilon) - 2^{-\lambda_{\text{sec}}} \leq \\ &\Pr[S \subseteq W \wedge X.\text{Verify}(\text{crs}, \pi) = 1] - \Pr[S \subseteq S_f \wedge X.\text{Verify}(\text{crs}, \pi) = 1] \leq \\ &\Pr[(S \subseteq W \wedge X.\text{Verify}(\text{crs}, \pi) = 1) \wedge \neg(S \subseteq S_f \wedge X.\text{Verify}(\text{crs}, \pi) = 1)] = \\ &\Pr[S \subseteq W \wedge S \not\subseteq S_f \wedge X.\text{Verify}(\text{crs}, \pi) = 1] \leq \\ &\Pr[S \subseteq W \wedge S \not\subseteq S_f] \leq \\ &\Pr[\exists x \in (S \cap W) \setminus S_f]. \end{aligned}$$

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 $(n_f + 1) \cdot \frac{1}{\varepsilon}$ iterations. Then it is easy to see that $Y.\text{Extract}$ runs in expected time $\text{poly}(T, 1/\varepsilon)$ (treating n_f as constant). \square

In summary, we achieve information-theoretic but non-adaptive security; i.e., additional computational power does not help the adversary avoid knowledge extraction but he is not allowed to choose the predicate / weight function based on the CRS. Even then, this can be useful; one example is applications where PRF seed is chosen by a randomness beacon after the statement to be proven is already decided. As a last resort, adaptive security can be achieved by rerandomizing the PRF using the random oracle: let the CRS be the output of the random oracle on the description of the weight function. This can be beneficial to instantiating ALBA purely in the random oracle model, for example, when the knowledge of an ALBA proof is proven by a SNARK. In that case, calculating the CRS outside of the SNARK circuit and using PRF inside the circuit lets one avoid heuristically instantiating RO in the circuit.

8 Performance Comparisons

In terms of prover computation, the Simple Lottery scheme requires negligible effort from the aggregator (apart from verifying membership and eligibility of the received set elements). Compact certificates require the prover to build a commitment to the set of received set items in the form of a Merkle tree, requiring $O(n)$ hash evaluations,

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^{80})	$356\sigma + 356\eta$		$208\sigma + 208\eta$		$104\sigma + 104\eta$	
C. Cert. [MRV ⁺ 21] (2^{128})	$438\sigma + 438\eta$		$256\sigma + 256\eta$		$128\sigma + 128\eta$	
Telescope, no weights (Sect. 3)	232σ		136σ		68σ	
Telescope, weights (Sect. 4.2,5)	241σ		141σ		71σ	
Simple Lottery (Sect. 4.1)	4157σ	5058σ	1428σ	1981σ	364σ	675σ
Simple Lottery ($\lambda_{\text{rel}} = 64$)	3060σ	3591σ	1069σ	1395σ	283σ	466σ
Decentralized Telescope (Sect. 4.2)	273σ	74105σ	159σ	28443σ	79σ	9068σ
	354σ	14919σ	205σ	5989σ	104σ	1987σ

Figure 2: Certificate sizes and expected communication cost, expressed in revealed/sent set elements (σ) and, in the case of [MRV⁺21], secondary reveals of the same elements in the form of Merkle Tree paths (η). The parameters $\lambda_{\text{sec}}, \lambda_{\text{rel}}$ are set to 128 unless otherwise indicated.

where n is the number of weighted elements in prover’s input. Telescope in turn requires $O(n + \lambda^2)$ hashes in expectation and Goldwasser-Sipser requires $O(n_p \cdot \lambda)$ hashes.

In terms of number of revealed elements, compact certificates need to reveal at least $\frac{\lambda_{\text{sec}}}{\log(n_p/n_f)}$ set elements (denoted by σ) but they additionally need to reveal the Merkle tree path of each element (denoted by η) with regards to the commitment constructed by the prover. The Simple Lottery scheme only reveals set elements and the number of reveals has the same, linear dependency on λ_{sec} , but has a more complex (and more costly) dependency on (n_p/n_f) . Telescope combines the best of both worlds, as it only needs to reveal close to $\frac{\lambda_{\text{sec}}}{\log(n_p/n_f)}$ set elements and integers v, t with no need for secondary openings. Goldwasser-Sipser requires $8\lambda \cdot (n_p/n_f)^4 \cdot (n_p/n_f - 1)^{-4}$ reveals.

In Figure 2 we compare proof sizes and communication costs of our constructions with those of existing protocols: 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. Compact certificates have computational security and we provide proof sizes secure against adversaries making 2^{80} and 2^{128} random oracle queries; Telescope, on the other hand, has information-theoretic security and smaller number of revealed elements, but becomes only computationally secure with number of revealed elements similar to compact certificates when the adversary is allowed to choose the weight function.

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 indices. For compact certificates, we derive values using the formula from [MRV⁺21]. For the simple lottery we use direct calculation, slightly improving on the bounds of Section 4.1. For Goldwasser-Sipser we use the analysis of Theorem 23 in the appendix. For Telescope we use the bounds from Corollary 2. For Decentralized Telescope we use parameters from Theorem 16 along with a soundness bound from

Theorem 25 and solving for u . There is a tradeoff between the proof size u and the expected communication μ , and we include two data points in each table column. For the weighted Telescope scheme we use the formula in Equation 3.

Acknowledgements

We are grateful to Mahak Pancholi and Akira Takahashi for discussion about UC SNARKs. This material is based upon work supported in part by a gift from Input Output - IOG and by DARPA under Agreements No. HR00112020021, HR00112020023 and HR001120C0085. 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.
- [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 Nikolai 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 Nikolai 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 22. *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 $n_f 2^{-\ell}$.

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

$$\begin{aligned} n_p 2^{-\ell} - \sum_{x, x'} \Pr[h(x) = y \wedge h(x') = y] &= \\ n_p 2^{-\ell} - \binom{n_p}{2} 2^{-2\ell} &\geq \\ n_p 2^{-\ell} - (n_p 2^{-\ell})^2/2 & \end{aligned}$$

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

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

Theorem 23. *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 $\gamma \in (0, 1 - n_f/n_p)$ and $\ell = \log(n_p/2\gamma)$. The expected number of adversarial successes is $\mu_f = 2^{-\ell}n_f = 2\gamma(n_f/n_p)u$. Similarly the expected number of honest party successes is $\mu_p = 2^{-\ell}(1 - \gamma)n_p = 2\gamma(1 - \gamma)u$. We set a threshold $T = \gamma u(1 - \gamma - n_f/n_p)$. Let $t = \gamma(1 - \gamma - n_f/n_p)u$. Observe that $\mu_f + t = T = \mu_p - t$. It follows by the Hoeffding bound that: (1) the probability that the adversarial parties reach $T = \mu_f + t$ 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 $\exp(-2t^2/u) \leq 2^{-\lambda_{sec}}$ and $\exp(-2t^2/u) \leq 2^{-\lambda_{rel}}$. Given that $2t^2/u = \gamma^2(1 - \gamma - n_f/n_p)^2u$ we obtain that it should hold

$$u \geq \gamma^{-2}(1 - \gamma - n_f/n_p)^{-2} \max\{\lambda_{sec}, \lambda_{rel}\}/2.$$

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

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 $[n_p]$ (or 1 with probability $1/[n_p]$, 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 $k = \lceil \log_2(n_p/\varepsilon_{\text{fail}}) \rceil$, and set $d = \lfloor 2^k/n_p \rfloor$. Use H to produce a k -bit string, interpret it as an integer $i \in [0, 2^k - 1]$, fail if $i \geq dn_p$, and output $i \bmod n_p$ 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 $n_p/2^k \leq \varepsilon_{\text{fail}}$. 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 1, 6, and 17 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 24. *Assume $0 \leq q \leq 1$ and*

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

Then completeness error is $\leq 2^{-\lambda_{rel}}$, 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, \dots, s_{u-k} , there is no suffix of honest player's elements that works, meaning there is no s_{u-k+1}, \dots, s_u such that for all $u-k+1 \leq i \leq u$, $H_1(t, s_1, \dots, s_i) = 1$, and $H_2(t, s_1, \dots, s_u) = 1$. Then

- $f(0) = 1 - q$;
- for $0 \leq k < u$, $f(k+1) = \left(1 - \frac{1}{n_p}\right) + \frac{1}{n_p} \cdot f(k)^{n_p}$;
- the probability that the algorithm fails in the honest case is $(f(u))^d$.

This recursive formula can be approximated:

$$\begin{aligned} f(k+1) &= \left(1 + \frac{1}{n_p}(f(k) - 1)\right)^{n_p} \leq \\ &= \left(e^{\frac{1}{n_p}(f(k)-1)}\right)^{n_p} = \\ &= e^{f(k)-1}. \end{aligned}$$

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 1. 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. \square

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 $f(u)^d \leq x_u^d = \exp(d \ln x_u) \leq \exp(-\lambda_{\text{rel}}/\log e) = 2^{-\lambda_{\text{rel}}}$, by the above claim and the definition of d . \square

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 1, we can make the following choice:

Corollary 9. *Let*

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

Then soundness error is $\leq 2^{-\lambda_{\text{sec}}}$ and completeness error is $\leq 2^{-\lambda_{\text{rel}}}$.

C.2 Improved soundness bound for Section 4.2

Theorem 25. *Assume $0 < \mu < n_p$ and $c = \frac{\mu n_f}{\mu n_p} \geq 1$. The soundness error is at most*

$$qrd \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u(c+2)}{2(c+1)^2} + 1 \right) < qrd \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u^2 n_p}{2\mu n_f} + 1 \right).$$

Proof. Let random variable N denote the number of adversarial elements chosen by the lottery and E be the event that a valid certificate can be formed using those N elements. Also define probability $p = \mu/n_p$. The soundness error is thus

$$\begin{aligned} & \sum_{i=0}^{n_f} \Pr[N = i] \cdot \Pr[E|N = i] = \\ & \sum_{i=0}^{n_f} C(n_f, i) \cdot p^i \cdot (1-p)^{n_f-i} \cdot \left(\frac{1}{\rho} \right)^u \cdot qrd \cdot i^u = \\ & qrd \cdot \left(\frac{pn_f}{\rho} \right)^u \cdot \sum_{i=1}^{n_f} \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u} \leq \end{aligned}$$

by Lemma 13,

$$\begin{aligned}
& [\leq] qrd \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u(c+2)}{2(c+1)^2} + 1 \right) = \\
& qrd \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u(c+2)}{2(c^2+2c+1)} + 1 \right) < \\
& qrd \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u}{2c} + 1 \right) = \\
& qrd \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u^2 n_p}{2\mu n_f} + 1 \right).
\end{aligned}$$

□

Lemma 13. Let $0 < p < 1$, assume $c = pn_f/u \geq 1$ and define

$$f(i) = \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u}.$$

Then

$$\sum_{i=1}^{n_f} f(i) \leq \exp \left(\frac{u(c+2)}{2(c+1)^2} + 1 \right).$$

Proof. Let $1 \leq i^* \leq n_f$ be an integer that maximizes $f(i^*)$ and for all $1 \leq i \leq n_f - 1$ define

$$\Delta(i) = \frac{f(i+1)}{f(i)}.$$

Suppose $i^* = n_f$. Then

$$\begin{aligned}
\sum_{i=1}^{n_f} f(i) &= f(n_f) + \sum_{j=0}^{n_f-2} f(n_f - j - 1) = \\
& f(n_f) \left(1 + \sum_{j=0}^{n_f-2} \prod_{k=0}^j \Delta^{-1}(n_f - k - 1) \right) [\leq]
\end{aligned}$$

by Lemma 14,

$$\begin{aligned}
[\leq] f(n_f) & \left(1 + \sum_{j=0}^{n_f-2} \prod_{k=0}^j \frac{n_f - k - 1}{(n_f - 1)(k + 1)} \right) \leq \\
& f(n_f) \left(1 + \sum_{j=0}^{n_f-2} \prod_{k=0}^j \frac{1}{k + 1} \right) = \\
& f(n_f) \left(1 + \sum_{j=0}^{n_f-2} \frac{1}{(j + 1)!} \right) \leq \\
& f(n_f) \sum_{j=0}^{\infty} \frac{1}{j!} [=]
\end{aligned}$$

by Taylor series (Lemma 33),

$$[=] e \cdot f(n_f) [\leq]$$

$f(n_f) = p^{n_f - u}$; since $pn_f/u \geq 1$, $n_f \geq u$ and $f(n_f) \leq 1$; then

$$[\leq] e \leq \exp\left(\frac{u(c+2)}{2(c+1)^2} + 1\right).$$

Now suppose $1 \leq i^* < n_f$. Combining Lemma 15 and Lemma 20,

$$\begin{aligned} \sum_{i=1}^{n_f} f(i) &\leq e \cdot \sqrt{\frac{2\pi i^*(n_f - i^*)}{n_f}} \cdot f(i^*) \leq \\ e \cdot \sqrt{\frac{2\pi i^*(n_f - i^*)}{n_f}} \cdot \sqrt{\frac{n_f}{2\pi i^*(n_f - i^*)}} \cdot \exp\left(\frac{u(c+2)}{2(c+1)^2}\right) &= \exp\left(\frac{u(c+2)}{2(c+1)^2} + 1\right). \end{aligned}$$

□

Lemma 14. *Let $0 < p < 1$, define*

$$f(i) = \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u}; \quad \Delta(i) = \frac{f(i+1)}{f(i)};$$

and let $1 \leq i^* \leq n_f$ be an integer that maximizes $f(i^*)$. Then for all $0 \leq z < n_f - i^*$,

$$\Delta(i^* + z) \leq \frac{i^*(n_f - i^* - z)}{(i^* + z)(n_f - i^*)}$$

and for all $0 \leq z < i^* - 1$,

$$\Delta^{-1}(i^* - z - 1) \leq \frac{(i^* - z - 1)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 1)}.$$

Proof.

$$\begin{aligned} \Delta(i) &= \frac{f(i+1)}{f(i)} = \\ \left(\frac{n_f! \cdot p^{i+1-u} \cdot (1-p)^{n_f-i-1} \cdot (i+1)^u}{(i+1)! \cdot (n_f-i-1)! \cdot n_f^u} \right) / \left(\frac{n_f! \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{i! \cdot (n_f-i)! \cdot n_f^u} \right) &= \\ \frac{p \cdot (n_f - i)}{(1-p) \cdot (i+1)} \cdot \left(\frac{i+1}{i} \right)^u &= \\ \frac{p \cdot (n_f - i)}{(1-p) \cdot i} \cdot \left(1 + \frac{1}{i} \right)^{u-1}. \end{aligned}$$

Then for $0 \leq z < n_f - i^*$,

$$\begin{aligned}
\Delta(i^* + z) &= \\
&\frac{p \cdot (n_f - i^* - z)}{(1-p) \cdot (i^* + z)} \cdot \left(1 + \frac{1}{i^* + z}\right)^{u-1} \leq \\
&\frac{p \cdot (n_f - i^* - z)}{(1-p) \cdot (i^* + z)} \cdot \left(1 + \frac{1}{i^*}\right)^{u-1} = \\
&\frac{p \cdot (n_f - i^*)}{(1-p) \cdot i^*} \cdot \left(1 + \frac{1}{i^*}\right)^{u-1} \cdot \frac{i^*(n_f - i^* - z)}{(i^* + z)(n_f - i^*)} = \\
&\Delta(i^*) \cdot \frac{i^*(n_f - i^* - z)}{(i^* + z)(n_f - i^*)} \leq \\
&\frac{i^*(n_f - i^* - z)}{(i^* + z)(n_f - i^*)}
\end{aligned}$$

since $\Delta(i^*) \leq 1$.

For $2 \leq i \leq n_f$,

$$\begin{aligned}
\Delta^{-1}(i-1) &= \\
&\left(\frac{p \cdot (n_f - i + 1)}{(1-p) \cdot (i-1)} \cdot \left(1 + \frac{1}{i-1}\right)^{u-1}\right)^{-1} = \\
&\frac{(1-p) \cdot (i-1)}{p \cdot (n_f - i + 1)} \cdot \left(1 - \frac{1}{i}\right)^{u-1}
\end{aligned}$$

and for $0 \leq z < i^* - 1$,

$$\begin{aligned}
\Delta^{-1}(i^* - z - 1) &= \\
&\frac{(1-p) \cdot (i^* - z - 1)}{p \cdot (n_f - i^* + z + 1)} \cdot \left(1 - \frac{1}{i^* - z}\right)^{u-1} \leq \\
&\frac{(1-p) \cdot (i^* - z - 1)}{p \cdot (n_f - i^* + z + 1)} \cdot \left(1 - \frac{1}{i^*}\right)^{u-1} = \\
&\frac{(1-p) \cdot (i^* - 1)}{p \cdot (n_f - i^* + 1)} \cdot \left(1 - \frac{1}{i^*}\right)^{u-1} \cdot \frac{(i^* - z - 1)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 1)} = \\
&\Delta^{-1}(i^* - 1) \cdot \frac{(i^* - z - 1)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 1)} \leq \\
&\frac{(i^* - z - 1)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 1)}
\end{aligned}$$

since $\Delta^{-1}(i^* - 1) \leq 1$. □

Lemma 15. Let $0 < p < 1$, define

$$f(i) = \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u};$$

let $1 \leq i^* \leq n_f$ be an integer that maximizes $f(i^*)$, and assume $i^* \neq n_f$. Then

$$\sum_{i=1}^{n_f} f(i) \leq e \cdot \sqrt{\frac{2\pi i^*(n_f - i^*)}{n_f}} \cdot f(i^*).$$

Proof. Define

$$\alpha = \frac{i^*(n_f - i^*)}{n_f}.$$

We need to prove that

$$\frac{\sum_{i=1}^{n_f} f(i)}{\sqrt{2\pi\alpha} \cdot f(i^*)} \leq e.$$

We consider multiple cases covering all possibilities. Throughout, we will use Lemmas 14, 16 and 18.

1. $i^* = 1$: Then $\alpha = \frac{n_f-1}{n_f}$. Since $n_f > i^*$, $\alpha \geq 1/2$.

$$\frac{\sum_{i=1}^{n_f} f(i)}{\sqrt{2\pi\alpha} \cdot f(i^*)} \leq \frac{1}{\sqrt{2\pi\alpha}} \left(1 + 2\sqrt{\alpha} + \frac{1}{\sqrt{\alpha}} \right) = \frac{2}{\sqrt{2\pi}} + \frac{1}{\sqrt{2\pi\alpha}} + \frac{1}{\sqrt{2\pi\alpha}} < 2.2.$$

2. $i^* = 2$: Then $\alpha = \frac{2(n_f-2)}{n_f}$. Since $n_f > i^*$, $\alpha \geq 2/3$.

$$\frac{\sum_{i=1}^{n_f} f(i)}{\sqrt{2\pi\alpha} \cdot f(i^*)} \leq \frac{1}{\sqrt{2\pi\alpha}} \left(1 + 1 + 2\sqrt{\alpha} + \frac{1}{\sqrt{\alpha}} \right) = \frac{2}{\sqrt{2\pi}} + \frac{2}{\sqrt{2\pi\alpha}} + \frac{1}{\sqrt{2\pi\alpha}} < 2.4.$$

3. $n_f - i^* = 1$, $i^* = 3$: Then $\alpha = 3/4$.

$$\frac{\sum_{i=1}^{n_f} f(i)}{\sqrt{2\pi\alpha} \cdot f(i^*)} \leq \frac{1}{\sqrt{2\pi\alpha}} \left(\frac{1}{2} + 1 + 1 + 1 \right) = \sqrt{\frac{2}{3\pi}} \cdot 3.5 < 1.7.$$

4. $n_f - i^* = 1$, $i^* \geq 4$: Then $\alpha = \frac{i^*}{i^*+1} \geq 4/5$.

$$\frac{\sum_{i=1}^{n_f} f(i)}{\sqrt{2\pi\alpha} \cdot f(i^*)} \leq \frac{1}{\sqrt{2\pi\alpha}} \left(2\sqrt{\alpha} + \frac{2}{\sqrt{\alpha}} + 1 + 1 \right) = \frac{2}{\sqrt{2\pi}} + \frac{2}{\sqrt{2\pi\alpha}} + \frac{2}{\sqrt{2\pi\alpha}} < 2.7.$$

5. $n_f - i^* = 2$, $i^* \geq 3$: Then $\alpha = \frac{2i^*}{i^*+2} \geq 6/5$.

$$\frac{\sum_{i=1}^{n_f} f(i)}{\sqrt{2\pi\alpha} \cdot f(i^*)} \leq \frac{1}{\sqrt{2\pi\alpha}} \left(2\sqrt{\alpha} + \frac{2}{\sqrt{\alpha}} + 1 + 1 + \frac{1}{2} \right) = \frac{2}{\sqrt{2\pi}} + \frac{2.5}{\sqrt{2\pi\alpha}} + \frac{2}{\sqrt{2\pi\alpha}} < 2.4.$$

6. $i^* \geq 3$, $n_f - i^* \geq 3$: First notice that

$$\alpha \geq \frac{\min\{i^*, n_f - i^*\}}{2} \geq \frac{3}{2}.$$

Then

$$\begin{aligned} & \frac{\sum_{i=1}^{n_f} f(i)}{\sqrt{2\pi\alpha} \cdot f(i^*)} \leq \\ & \frac{1}{\sqrt{2\pi\alpha}} \left(2\sqrt{\alpha} - 1 + \frac{2}{\sqrt{\alpha}} + \frac{i^*}{n_f} + 1 + 2\sqrt{\alpha} + \frac{1}{\sqrt{\alpha}} - \frac{i^*}{n_f} \right) = \\ & \frac{1}{\sqrt{2\pi\alpha}} \left(4\sqrt{\alpha} + \frac{3}{\sqrt{\alpha}} \right) = \\ & \frac{4}{\sqrt{2\pi}} + \frac{3}{\sqrt{2\pi\alpha}} < 2.4. \end{aligned}$$

□

Lemma 16. Let $0 < p < 1$, define

$$f(i) = \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u}; \quad \alpha = \frac{i^*(n_f - i^*)}{n_f};$$

and let $1 \leq i^* \leq n_f$ be an integer that maximizes $f(i)$. Then

$$\sum_{i=1}^{i^*-1} f(i) \leq f(i^*) \left(2\sqrt{\alpha} - 1 + \frac{2}{\sqrt{\alpha}} + \frac{i^*}{n_f} \right).$$

Proof. Let $z = \lfloor \sqrt{\alpha} \rfloor$. By Lemma 17,

$$\begin{aligned} & \sum_{i=1}^{i^*-1} f(i) \leq \\ & f(i^*) \left(z + \frac{i^*(n_f - i^*)}{(z+1)n_f} + \frac{i^*}{n_f} + \frac{1}{z+1} \right) = \\ & f(i^*) \left(\lfloor \sqrt{\alpha} \rfloor + \frac{\alpha}{\lfloor \sqrt{\alpha} \rfloor + 1} + \frac{i^*}{n_f} + \frac{1}{\lfloor \sqrt{\alpha} \rfloor + 1} \right) [=] \end{aligned}$$

letting $\lfloor \sqrt{\alpha} \rfloor = \sqrt{\alpha} - \varepsilon$ where $0 \leq \varepsilon < 1$,

$$\begin{aligned}
& [=] f(i^*) \left(\sqrt{\alpha} - \varepsilon + \frac{\alpha + 1}{\sqrt{\alpha} - \varepsilon + 1} + \frac{i^*}{n_f} \right) = \\
& f(i^*) \left(2\sqrt{\alpha} - \varepsilon + \frac{-\sqrt{\alpha} + \varepsilon\sqrt{\alpha} + 1}{\sqrt{\alpha} - \varepsilon + 1} + \frac{i^*}{n_f} \right) = \\
& f(i^*) \left(2\sqrt{\alpha} + \frac{-\sqrt{\alpha} + 1 + \varepsilon^2 - \varepsilon}{\sqrt{\alpha} - \varepsilon + 1} + \frac{i^*}{n_f} \right) = \\
& f(i^*) \left(2\sqrt{\alpha} - 1 + \frac{2 + \varepsilon^2 - 2\varepsilon}{\sqrt{\alpha} - \varepsilon + 1} + \frac{i^*}{n_f} \right) \leq \\
& f(i^*) \left(2\sqrt{\alpha} - 1 + \frac{2}{\sqrt{\alpha}} + \frac{i^*}{n_f} \right).
\end{aligned}$$

□

Lemma 17. *Let $0 < p < 1$, define*

$$f(i) = \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u}$$

and let $1 \leq i^* \leq n_f$ be an integer that maximizes $f(i^*)$. Then for all integers $z \geq 0$,

$$\sum_{i=1}^{i^*-1} f(i) \leq f(i^*) \left(z + \frac{i^*(n_f - i^*)}{(z+1)n_f} + \frac{i^*}{n_f} + \frac{1}{z+1} \right).$$

Proof. Define $\Delta(i) = \frac{f(i+1)}{f(i)}$. For all integers $0 \leq z < i^* - 2$,

$$\begin{aligned}
& \sum_{i=1}^{i^*-z-1} f(i) = \\
& \sum_{j=0}^{i^*-z-2} f(i^* - z - 1 - j) = \\
& \sum_{j=0}^{i^*-z-2} f(i^* - z - 1) \prod_{k=1}^j \Delta^{-1}(i^* - z - 1 - k) \leq \\
& f(i^*) \sum_{j=0}^{i^*-z-2} \prod_{k=1}^j \Delta^{-1}(i^* - z - 1 - k) \leq \\
& f(i^*) \sum_{j=0}^{i^*-z-2} \Delta^{-j}(i^* - z - 2) [\leq]
\end{aligned}$$

by Lemma 14,

$$\begin{aligned} [\leq] f(i^*) \sum_{j=0}^{\infty} \left(\frac{(i^* - z - 2)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 2)} \right)^j &= \\ f(i^*) \cdot \frac{1}{1 - \frac{(i^* - z - 2)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 2)}}. \end{aligned}$$

The restriction that $z < i^* - 2$ can be replaced with $z \leq i^* - 2$ since when $z = i^* - 2$,

$$\sum_{i=1}^{i^* - z - 1} f(i) = f(1) \leq f(i^*) = f(i^*) \cdot \frac{1}{1 - \frac{(i^* - z - 2)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 2)}}.$$

Hence, for all $0 \leq z \leq i^* - 2$,

$$\begin{aligned} \sum_{i=1}^{i^* - z - 1} f(i) &\leq f(i^*) \cdot \frac{1}{1 - \frac{(i^* - z - 2)(n_f - i^* + 1)}{(i^* - 1)(n_f - i^* + z + 2)}} = \\ f(i^*) \left(\frac{i^*(n_f - i^*)}{(z + 1)n_f} + \frac{i^*}{n_f} + \frac{2i^*}{(z + 1)n_f} - \frac{1}{z + 1} - \frac{1}{n_f} - \frac{1}{(z + 1)n_f} \right) &\leq \\ f(i^*) \left(\frac{i^*(n_f - i^*)}{(z + 1)n_f} + \frac{i^*}{n_f} + \frac{1}{z + 1} \right). \end{aligned}$$

If $z \leq i^* - 2$,

$$\begin{aligned} \sum_{i=1}^{i^* - 1} f(i) &= \sum_{i=1}^{i^* - z - 1} f(i) + \sum_{i=i^* - z}^{i^* - 1} f(i) = \\ f(i^*) \left(z + \frac{i^*(n_f - i^*)}{(z + 1)n_f} + \frac{i^*}{n_f} + \frac{1}{z + 1} \right). \end{aligned}$$

Otherwise if $z > i^* - 2$,

$$\sum_{i=1}^{i^* - 1} f(i) \leq (i^* - 1)f(i^*) \leq z f(i^*) \leq f(i^*) \left(z + \frac{i^*(n_f - i^*)}{(z + 1)n_f} + \frac{i^*}{n_f} + \frac{1}{z + 1} \right).$$

□

Lemma 18. Let $0 < p < 1$, define

$$f(i) = \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u}; \quad \alpha = \frac{i^*(n_f - i^*)}{n_f};$$

and let $1 \leq i^* \leq n_f$ be an integer that maximizes $f(i^*)$. Then

$$\sum_{i=i^*+1}^{n_f} f(i) \leq f(i^*) \cdot \left(2\sqrt{\alpha} + \frac{1}{\sqrt{\alpha}} - \frac{i^*}{n_f} \right).$$

Proof. Let $z = \lfloor \sqrt{\alpha} \rfloor$. By Lemma 19,

$$\begin{aligned} & \sum_{i=i^*+1}^{n_f} f(i) \leq \\ & f(i^*) \left(z + \frac{i^*(n_f - i^*)}{(z+1)n_f} + 1 - \frac{i^*}{n_f} \right) = \\ & f(i^*) \left(\lfloor \sqrt{\alpha} \rfloor + \frac{\alpha}{\lfloor \sqrt{\alpha} \rfloor + 1} + 1 - \frac{i^*}{n_f} \right) [=] \end{aligned}$$

letting $\lfloor \sqrt{\alpha} \rfloor = \sqrt{\alpha} - \varepsilon$ where $0 \leq \varepsilon < 1$,

$$\begin{aligned} [=] & f(i^*) \left(\sqrt{\alpha} - \varepsilon + \frac{\alpha}{\sqrt{\alpha} - \varepsilon + 1} + 1 - \frac{i^*}{n_f} \right) = \\ & f(i^*) \left(2\sqrt{\alpha} - \varepsilon + \frac{\varepsilon\sqrt{\alpha} - \sqrt{\alpha}}{\sqrt{\alpha} - \varepsilon + 1} + 1 - \frac{i^*}{n_f} \right) = \\ & f(i^*) \left(2\sqrt{\alpha} + \frac{-\sqrt{\alpha} + \varepsilon^2 - \varepsilon}{\sqrt{\alpha} - \varepsilon + 1} + 1 - \frac{i^*}{n_f} \right) = \\ & f(i^*) \left(2\sqrt{\alpha} + \frac{\varepsilon^2 - 2\varepsilon + 1}{\sqrt{\alpha} - \varepsilon + 1} - \frac{i^*}{n_f} \right) \leq \\ & f(i^*) \left(2\sqrt{\alpha} + \frac{1}{\sqrt{\alpha}} - \frac{i^*}{n_f} \right). \end{aligned}$$

□

Lemma 19. Let $0 < p < 1$, define

$$f(i) = \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u}$$

and let $1 \leq i^* \leq n_f$ be an integer that maximizes $f(i^*)$. Then for all integers $z \geq 0$,

$$\sum_{i=i^*+1}^{n_f} f(i) \leq f(i^*) \cdot \left(z + \frac{i^*(n_f - i^*)}{(z+1)n_f} + 1 - \frac{i^*}{n_f} \right).$$

Proof. Define $\Delta(i) = \frac{f(i+1)}{f(i)}$. For all integers $0 \leq z < n_f - i^* - 1$,

$$\begin{aligned}
& \sum_{i=i^*+z+1}^{n_f} f(i) = \\
& \sum_{j=0}^{n_f-i^*-z-1} f(i^* + z + 1 + j) = \\
& \sum_{j=0}^{n_f-i^*-z-1} f(i^* + z + 1) \prod_{k=1}^j \Delta(i^* + z + k) \leq \\
& f(i^*) \sum_{j=0}^{n_f-i^*-z-1} \prod_{k=1}^j \Delta(i^* + z + k) \leq \\
& f(i^*) \sum_{j=0}^{n_f-i^*-z-1} \Delta^j(i^* + z + 1) [\leq]
\end{aligned}$$

by Lemma 14,

$$\begin{aligned}
[\leq] f(i^*) \sum_{j=0}^{\infty} \left(\frac{i^*(n_f - i^* - z - 1)}{(i^* + z + 1)(n_f - i^*)} \right)^j &= \\
f(i^*) \cdot \frac{1}{1 - \frac{i^*(n_f - i^* - z - 1)}{(i^* + z + 1)(n_f - i^*)}}. &
\end{aligned}$$

The restriction that $z < n_f - i^* - 1$ can be replaced with $z \leq n_f - i^* - 1$ since when $z = n_f - i^* - 1$,

$$\sum_{i=i^*+z+1}^{n_f} f(i) = f(n_f) \leq f(i^*) = f(i^*) \cdot \frac{1}{1 - \frac{i^*(n_f - i^* - z - 1)}{(i^* + z + 1)(n_f - i^*)}}.$$

Hence, for all $0 \leq z \leq n_f - i^* - 1$,

$$\begin{aligned}
& \sum_{i=i^*+z+1}^{n_f} f(i) \leq \\
& f(i^*) \cdot \frac{1}{1 - \frac{i^*(n_f - i^* - z - 1)}{(i^* + z + 1)(n_f - i^*)}} = \\
& f(i^*) \cdot \left(\frac{i^*(n_f - i^*)}{(z + 1)n_f} + 1 - \frac{i^*}{n_f} \right).
\end{aligned}$$

If $z \leq n_f - i^* - 1$, then

$$\sum_{i=i^*+1}^{n_f} f(i) = \sum_{i=i^*+1}^{i^*+z} f(i) + \sum_{i=i^*+z+1}^{n_f} f(i) \leq f(i^*) \left(z + \frac{i^*(n_f - i^*)}{(z + 1)n_f} + 1 - \frac{i^*}{n_f} \right).$$

Otherwise if $z > n_f - i^* - 1$, also

$$\sum_{i=i^*+1}^{n_f} f(i) \leq (n_f - i^*)f(i^*) \leq zf(i^*) \leq f(i^*) \left(z + \frac{i^*(n_f - i^*)}{(z+1)n_f} + 1 - \frac{i^*}{n_f} \right).$$

□

Lemma 20. Assume $0 < p < 1$, $1 \leq i \leq n_f - 1$ and $c = pn_f/u \geq 1$. Then

$$\frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u} \leq \sqrt{\frac{n_f}{2\pi i(n_f-i)}} \cdot \exp\left(\frac{u(c+2)}{2(c+1)^2}\right).$$

Proof. By Stirling's approximation (Lemma 36),

$$\begin{aligned} & \frac{C(n_f, i) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{n_f^u} = \\ & \frac{n_f! \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{i! \cdot (n_f-i)! \cdot n_f^u} \leq \\ & \frac{\sqrt{2\pi n_f} \cdot \left(\frac{n_f}{e}\right)^{n_f} \cdot \exp\left(\frac{1}{12n_f}\right) \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{\sqrt{2\pi i} \cdot \left(\frac{i}{e}\right)^i \cdot \exp\left(\frac{1}{12i+1}\right) \cdot \sqrt{2\pi(n_f-i)} \cdot \left(\frac{n_f-i}{e}\right)^{n_f-i} \cdot \exp\left(\frac{1}{12(n_f-i)+1}\right) \cdot n_f^u} \quad [\leq] \end{aligned}$$

since $1 \leq i \leq n_f - 1$, $\exp\left(\frac{1}{12n_f}\right) \leq \exp\left(\frac{1}{12i+1}\right)$; thus,

$$\begin{aligned} & [\leq] \frac{\sqrt{2\pi n_f} \cdot \left(\frac{n_f}{e}\right)^{n_f} \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{\sqrt{2\pi i} \cdot \left(\frac{i}{e}\right)^i \cdot \sqrt{2\pi(n_f-i)} \cdot \left(\frac{n_f-i}{e}\right)^{n_f-i} \cdot n_f^u} = \\ & \sqrt{\frac{n_f}{2\pi i(n_f-i)}} \cdot \frac{n_f^{n_f} \cdot p^{i-u} \cdot (1-p)^{n_f-i} \cdot i^u}{i^i \cdot (n_f-i)^{n_f-i} \cdot n_f^u} = \\ & \sqrt{\frac{n_f}{2\pi i(n_f-i)}} \cdot \left(\frac{pn_f}{i}\right)^{i-u} \cdot \left(\frac{(1-p)n_f}{n_f-i}\right)^{n_f-i} = \\ & \sqrt{\frac{n_f}{2\pi i(n_f-i)}} \cdot \left(\frac{pn_f}{i}\right)^{i-u} \cdot \left(1 + \frac{i-pn_f}{n_f-i}\right)^{n_f-i} \leq \\ & \sqrt{\frac{n_f}{2\pi i(n_f-i)}} \cdot \left(\frac{pn_f}{i}\right)^{i-u} \cdot e^{i-pn_f} \quad [\leq] \end{aligned}$$

by Lemma 21,

$$[\leq] \sqrt{\frac{n_f}{2\pi i(n_f-i)}} \cdot \exp\left(\frac{u(c+2)}{2(c+1)^2}\right).$$

□

Lemma 21. Assume $i \geq 1$, $0 < p \leq 1$ and $c = pn_f/u \geq 1$. Then

$$\left(\frac{pn_f}{i}\right)^{i-u} \cdot e^{i-pn_f} \leq \exp\left(\frac{u(c+2)}{2(c+1)^2}\right).$$

Proof.

$$\ln \left(\left(\frac{pn_f}{i} \right)^{i-u} \cdot e^{i-pn_f} \right) = (i-u) \ln \frac{pn_f}{i} + i - pn_f [=]$$

Letting $x = i/u$ and $c = pn_f/u$,

$$\begin{aligned} [=] u \left((x-1) \ln \frac{pn_f}{xu} + x - \frac{pn_f}{u} \right) &= \\ u \left((x-1) \ln \frac{c}{x} + x - c \right) &[\leq] \end{aligned}$$

By Lemma 22,

$$[\leq] u \cdot \frac{c+2}{2(c+1)^2}.$$

□

Lemma 22. *Let $x > 0$ and $c \geq 1$. Then*

$$(1-x) \ln \frac{x}{c} + x - c \leq \frac{c+2}{2(c+1)^2}.$$

Proof. Define function

$$f(x) = (1-x) \ln \frac{x}{c} + x - c,$$

then

$$f'(x) = -\ln \frac{x}{c} + (1-x) \cdot \frac{c}{x} \cdot \frac{1}{c} + 1 = \frac{1}{x} - \ln \frac{x}{c}.$$

Thus, $f(x)$ is increasing on the interval $(0, c]$ and for any x in this interval, $f(x) \leq f(c) = 0$ and we only need to prove the bound for $x > c$.

Assume $z > c$, then

$$\begin{aligned} f(z) &= (1-z) \ln \frac{z}{c} + z - c = \\ &(z-1) \ln \frac{c}{z} + z - c = \\ &(z-1) \ln \left(1 - \frac{z-c}{z} \right) + z - c [\leq] \end{aligned}$$

By Lemma 30,

$$\begin{aligned} [\leq] &(z-1) \cdot \left(-\frac{z-c}{z} - \frac{(z-c)^2}{2z^2} \right) + z - c = \\ &(z-c) \left(1 - \frac{z-1}{z} - \frac{(z-1)(z-c)}{2z^2} \right) = \\ &(z-c) \left(\frac{1}{z} - \frac{(z-1)(z-c)}{2z^2} \right). \end{aligned}$$

Define function

$$g(x) = (x - c) \left(\frac{1}{x} - \frac{(x - 1)(x - c)}{2x^2} \right).$$

By Lemma 23, $g(x)$ is non-decreasing for $x \in [c, c + 1]$. Hence, $f(z) \leq g(z) \leq g(c + 1) = \frac{c+2}{2(c+1)^2}$. \square

Lemma 23. *Let $c \geq 1$ and define function*

$$g(x) = (x - c) \left(\frac{1}{x} - \frac{(x - 1)(x - c)}{2x^2} \right).$$

$g(x)$ is non-decreasing for $x \in [c, c + 1]$.

Proof. It can be checked that the derivative

$$g'(x) = -\frac{1}{2x^3} (x^3 - c(c + 4)x + 2c^2)$$

which is non-negative for $x \in [c, c + 1]$ by Lemma 24. \square

Lemma 24. *Let $c \geq 1$ and define function $h(x) = x^3 - c(c + 4)x + 2c^2$. Then $h(x) \leq 0$ for $x \in [c, c + 1]$.*

Proof. First notice that $h(c) = -2c^2 \leq 0$ and $h(c + 1) = 1 - c \leq 0$. Also the derivative $h'(x) = 3x^2 - c(c + 4)$. Then $h(x)$ is decreasing at $x = 0$ and changes direction only at one coordinate $x > 0$. Hence, $h(x) \leq 0$ for $x \in [c, c + 1]$. \square

C.3 Replacing the Random Oracle with PRF

Theorem 26. *Take any set S_p with $|S_p| \geq n_p$. Then the construction R satisfies $\Pr[\text{CompExp}(S_p) = 1] \geq 1 - 2^{-\lambda} - \varepsilon_{\text{prf}}(O(n_p + \lambda^3))$.*

Proof. Define

procedure \mathcal{A}^O
 $\left[\begin{array}{l} \pi \leftarrow \text{Prove}^O(S_p); \\ \text{return } 1 \text{ iff } \pi \neq \perp. \end{array} \right.$

By the assumption about our Telescope construction, $\Pr[\mathcal{A}^H() = 1] \geq 1 - 2^{-\lambda}$. Since the running time of \mathcal{A}^O is bounded by $O(n_p + B) = O(n_p + \lambda^3)$, equation 4 gives

$$\Pr \left[\mathcal{A}^{F(\text{GenKey}(), \cdot)}() = 1 \right] \geq 1 - 2^{-\lambda} - \varepsilon_{\text{prf}}(O(n_p + \lambda^3)).$$

But

$$\Pr \left[\mathcal{A}^{F(\text{GenKey}(), \cdot)}() = 1 \right] = \Pr[\text{CompExp}(S_p) = 1].$$

\square

Theorem 27. *Let S_f be any set with $|S_f| \leq n_f$. Then the construction R satisfies $\Pr[\text{SoundExp}(S_f) = 1] \leq 2^{-\lambda} + \varepsilon_{\text{prf}}(O(n_p + \lambda^3))$.*

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

By the assumption about our Telescope construction, $\Pr[\mathcal{A}^H() = 1] \leq 2^{-\lambda}$. Since the running time of \mathcal{A}^O is bounded by $O(n_f + B) = O(n_p + \lambda^3)$, equation 4 gives

$$\Pr \left[\mathcal{A}^{F(\text{GenKey}(), \cdot)}() = 1 \right] \leq 2^{-\lambda} + \varepsilon_{\text{prf}}(O(n_p + \lambda^3)).$$

But since $\text{SoundExp}(S_f) = 1$ implies $\mathcal{A}^{F(\text{GenKey}(), \cdot)}() = 1$,

$$\Pr[\text{SoundExp}(S_f) = 1] \leq \Pr \left[\mathcal{A}^{F(\text{GenKey}(), \cdot)}() = 1 \right].$$

□

D Proofs

Proof of Theorem 3.

$$\begin{aligned} \mathbb{E} \left[\sum_{s_1, \dots, s_i \in S_p} A_{j, s_1, \dots, s_i} \right] &= \\ \sum_{s_1, \dots, s_i \in S_p} \mathbb{E}[A_{j, s_1, \dots, s_i}] &= \\ \sum_{s_1, \dots, s_i \in S_p} \mathbb{E} \left[\prod_{j=1}^i H_1(j, s_1, \dots, s_j) \right] &= \\ \sum_{s_1, \dots, s_i \in S_p} \prod_{j=1}^i \mathbb{E} [H_1(j, s_1, \dots, s_j)] &= \\ \sum_{s_1, \dots, s_i \in S_p} \prod_{j=1}^i \frac{1}{n_p} &= \\ \sum_{s_1, \dots, s_i \in S_p} \left(\frac{1}{n_p} \right)^i &= \\ n_p^i \cdot \left(\frac{1}{n_p} \right)^i &= \\ 1. \end{aligned}$$

□

Proof of Theorem 4. Let X_i be the number of hash invocations in tree i and let Y_i be 1 if a valid proof starting with integer i exists and 0 otherwise. As described

earlier, $\mathbb{E}[X_i] \leq n_p u + 1$. Also by Lemma 2,

$$\Pr[Y_i] \geq 1 - \exp\left(-q + u \cdot \frac{q^2}{2}\right).$$

Therefore by Lemma 35, the expected total number of hash invocations is at most

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

The statement of the theorem then follows from Lemma 31. \square

Proof of Theorem 5. Let $t > 0$ and define the sequence $\{x_k\}$ as follows: let $x_0 = 1$ and for $k \geq 0$, let

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

By Lemma 41, $\mathbb{E}[e^{tZ}] = x_u^d$.

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: $x_{k+1} = \left(\frac{1}{n}x_k e^t + 1 - \frac{1}{n}\right)^{n_p} = \left(1 + \frac{1}{n}(x_k e^t - 1)\right)^{n_p} \leq \left(e^{\frac{1}{n}(x_k e^t - 1)}\right)^{n_p} = \exp(x_k e^t - 1) \leq \exp(y_k + t - 1)$. Since $y_k + t \leq y_k + t + (y_k + t)^2 = y_{k+1} \leq y_u \leq 1$, $x_{k+1} \leq \exp(1 + y_k + t + (y_k + t)^2 - 1) = \exp(y_{k+1})$. Hence, $\mathbb{E}[e^{tZ}] \leq e^{y_u d}$.

By Markov's inequality,

$$\begin{aligned} \Pr[Z \geq (1 + \delta)du] &= \Pr[e^{tZ} \geq e^{(1+\delta)tdu}] \leq \\ &= \frac{e^{y_u d}}{e^{(1+\delta)tdu}} = \exp\left(-d((1 + \delta)tu - y_u)\right). \end{aligned} \tag{5}$$

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 $y_k \geq y_{k+1} - y_{k+1}^2 - t \geq y_{k+1} - \alpha^2 - t$. 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: $y_{k+1} = y_k + t + (y_k + t)^2 \leq \frac{\alpha k}{u} + \frac{\alpha}{u} - \alpha^2 + \left(\frac{\alpha k}{u} + \frac{\alpha}{u} - \alpha^2\right)^2 = \frac{\alpha(k+1)}{u} - \alpha^2 + \left(\frac{\alpha(k+1)}{u} - \alpha^2\right)^2 \leq \frac{\alpha(k+1)}{u} - \alpha^2 + (\alpha - \alpha^2)^2 \leq \frac{\alpha(k+1)}{u} - \alpha^2 + \alpha^2 = \frac{\alpha(k+1)}{u}$.

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,

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

Hence by equation 5,

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

□

Proof of Lemma 4. We will use Markov's inequality with Poisson approximation.

Define $Z = \frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} - (1 - q)$. Then

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

Since $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq \frac{1}{n_p} \sum_{i=1}^{n_p} (1 - qX_i) = 1 - \frac{q}{n_p} \sum_{i=1}^{n_p} X_i = 1 - q$, Z is a non-negative random variable; by Markov's inequality,

$$\begin{aligned}
\leq \frac{\mathbb{E}[Z]}{cq^2} &= \frac{\mathbb{E} \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} - (1 - q) \right]}{cq^2} = \\
& \frac{\mathbb{E} \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} - \frac{1}{n_p} \sum_{i=1}^{n_p} (1 - qX_i) \right]}{cq^2} = \\
& \frac{\mathbb{E} \left[\frac{1}{n_p} \sum_{i=1}^{n_p} (e^{-qX_i} - (1 - qX_i)) \right]}{cq^2} \leq
\end{aligned}$$

Since $\frac{1}{n_p} \sum_{i=1}^{n_p} (e^{-qx_i} - (1 - qx_i)) \geq 0$ for any x_1, \dots, x_{n_p} and since the derivative $(e^{-qx} - (1 - qx))' = -qe^{-qx} + q = q(1 - e^{-qx}) \geq 0$ for any $x \geq 0$, one can see that all conditions for Poisson approximation in [MU05, Theorem 5.10] are satisfied. Then, letting Y_i be independent Poisson random variables with mean 1 (i.e., for all integers

$j \geq 0$, $\Pr[Y_i = j] = \frac{1}{e j!}$),

$$\begin{aligned} & \leq \frac{2 \cdot \mathbb{E} \left[\frac{1}{n_p} \sum_{i=1}^{n_p} (e^{-qY_i} - (1 - qY_i)) \right]}{cq^2} = \\ & \frac{\frac{2}{n_p} \sum_{i=1}^{n_p} \left(\mathbb{E} [e^{-qY_i}] - (1 - q) \right)}{cq^2}. \end{aligned}$$

Now,

$$\begin{aligned} \mathbb{E} [e^{-qY_i}] &= \sum_{j=0}^{\infty} e^{-qj} \cdot \frac{1}{e \cdot j!} \leq \\ & \frac{1}{e} \cdot \sum_{j=0}^{\infty} \left(1 - qj + \frac{(qj)^2}{2} \right) \cdot \frac{1}{j!} = \\ & \frac{1}{e} \cdot \left(\sum_{j=0}^{\infty} \frac{1}{j!} - q \sum_{j=0}^{\infty} \frac{j}{j!} + \frac{q^2}{2} \sum_{j=0}^{\infty} \frac{j^2}{j!} \right) [=] \end{aligned}$$

By Lemma 33,

$$\begin{aligned} [=] \frac{1}{e} \cdot \left(e - q \cdot e + \frac{q^2}{2} \cdot 2e \right) &= \\ 1 - q + q^2. \end{aligned}$$

Combining this with the previous inequality, we get

$$\begin{aligned} \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq 1 - q + cq^2 \right] &\leq \\ \frac{\frac{2}{n_p} \sum_{i=1}^{n_p} \left((1 - q + q^2) - (1 - q) \right)}{cq^2} &= \frac{2}{c}. \end{aligned}$$

□

Proof of Lemma 5. We use Poisson approximation: for $1 \leq i \leq n_p$, let Y_i be independent Poisson random variables with mean 1; i.e., for all integers $j \geq 0$, $\Pr[Y_i = j] = \frac{1}{e j!}$. Then by [MU05, Theorem 5.10],

$$\Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq 1 - q + 4q^2 \right] \leq 2 \cdot \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qY_i} \geq 1 - q + 4q^2 \right]. \quad (6)$$

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 $t > 0$,

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

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

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

By Lemma 33,

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

Combining this bound, equation 6 and Markov's inequality, for any $s > 0$ we get

$$\begin{aligned}
& \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq 1 - q + 4q^2 \right] \leq \\
& 2 \cdot \Pr \left[e^{\frac{s}{n_p} \sum_{i=1}^{n_p} e^{-qY_i}} \geq e^{s(1-q+4q^2)} \right] \leq \\
& 2 \cdot \frac{\mathbb{E} \left[e^{\frac{s}{n_p} \sum_{i=1}^{n_p} e^{-qY_i}} \right]}{e^{s(1-q+4q^2)}} = \\
& 2 \cdot \frac{\mathbb{E} \left[\prod_{i=1}^{n_p} e^{\frac{s}{n_p} e^{-qY_i}} \right]}{e^{s(1-q+4q^2)}} = \\
& 2 \cdot \frac{\prod_{i=1}^{n_p} \mathbb{E} \left[e^{\frac{s}{n_p} e^{-qY_i}} \right]}{e^{s(1-q+4q^2)}} \leq \\
& 2 \cdot \frac{\prod_{i=1}^{n_p} e^{\frac{s}{n_p} (1-q+(1+\frac{s}{n_p})q^2)}}{e^{s(1-q+4q^2)}} = \\
& 2 \cdot \frac{e^{s(1-q+(1+\frac{s}{n_p})q^2)}}{e^{s(1-q+4q^2)}} = \\
& 2e^{-\left(4-1-\frac{s}{n_p}\right)sq^2}.
\end{aligned}$$

Setting $s = \frac{3}{2}n_p$, we get

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

□

Proof of Lemma 6. Define random function $f(x) = \frac{1}{n_p} \sum_{i=1}^{n_p} x^{X_i}$. By Lemma 45,

$$\Pr[F|H_0] \leq \left(e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}} \right)^u \right)^d.$$

Therefore,

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

□

Proof of Theorem 7. 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 $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+4q^2}$ and let F be the event that the honest prover fails. By Lemma 43 with $\lambda := \frac{\lambda_{\text{rel}} + \log 3}{\log e}$ and $c := 4$, $\Pr[F|E] \leq \frac{1}{3} \cdot 2^{-\lambda_{\text{rel}}}$. Also by Lemma 5,

$$\begin{aligned} \Pr[\bar{E}] &= \\ \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} > e^{-q+4q^2}\right] &\leq \\ \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq 1 - q + 4q^2\right] &\leq \\ &2e^{-\frac{9}{4}n_p q^2}. \end{aligned}$$

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

$$\begin{aligned} 3 \cdot 2^{\lambda_{\text{rel}}} &\leq e^{\frac{9}{4}n_p q^2} \iff \\ \frac{9}{4} \log e \cdot n_p q^2 &\geq \lambda_{\text{rel}} + \log 3 \iff \\ n_p &\geq \frac{4(\lambda_{\text{rel}} + \log 3)}{9 \log e \cdot q^2} \iff \\ n_p &\geq \frac{4(\lambda_{\text{rel}} + \log 3)}{9 \log e \cdot \left(\frac{2(\lambda_{\text{rel}} + \log 3)}{d \log e}\right)^2} \iff \\ n_p &\geq \frac{4(\lambda_{\text{rel}} + \log 3)}{9 \log e \cdot \frac{4(\lambda_{\text{rel}} + \log 3)^2}{d^2 \log^2 e}} \iff \\ n_p &\geq \frac{d^2 \log e}{9(\lambda_{\text{rel}} + \log 3)} \end{aligned}$$

which is true by our assumption about n_p .

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

Proof of Lemma 7. By Lemma 44, a single tree does not contain a valid proof with probability at most $\exp(-q + cuq^2)$ given event E . Also by Theorem 8, the expected number of vertices that the algorithm visits in a single tree is $\leq u + 1$, given any arrangement of balls into bins. Thus, using Lemma 35 and Lemma 31,

$$\mathbb{E}[V|E] \leq \frac{u+1}{1 - e^{-q+cuq^2}} \leq \frac{2(u+1)}{q - cuq^2}.$$

\square

Proof of Theorem 9. Let V be the number of visited vertices, let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i and let E be the event that

$\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+4q^2}$. Then

$$\mathbb{E}[V] = \mathbb{E}[V|E] \cdot \Pr[E] + \mathbb{E}[V|\bar{E}] \cdot \Pr[\bar{E}].$$

Clearly, $\Pr[E] \leq 1$. By Theorem 8, there are $d(u+1)$ accessible vertices in expectation; thus, $\mathbb{E}[V|\bar{E}] \leq d(u+1)$. By Lemma 5, $\Pr[\bar{E}] \leq 2e^{-\frac{9}{4}n_p q^2}$. Finally, by Lemma 7 with $c := 4$,

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

□

Proof of Theorem 10. Let

$$\alpha = \frac{1}{w},$$

$$c = \left(\frac{1}{\alpha} + w\right) \cdot \frac{w\lambda'}{n_p} + 2\left(1 + \frac{1}{\alpha w}\right) + 3 = \frac{2w^2\lambda}{n_p} + 7,$$

let $X_i = |\{s \in S_p : H_0(s) = i\}|$ 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 $\Pr[E] > 0$. By Lemma 48 with $\lambda := \lambda'$,

$$\Pr[Z \geq \delta du | E] \leq e^{-\lambda'} = \frac{2^{-\lambda}}{4}. \quad (7)$$

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

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

and

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

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

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

By Poisson approximation [MU05, Theorem 5.10],

$$\begin{aligned}
& [\leq] 2 \cdot \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{\alpha Y_i} \geq 1 + \alpha + c\alpha^2 \right] = \\
& 2 \cdot \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} (A_i + B_i) \geq 1 + \alpha + c\alpha^2 \right] = \\
& 2 \cdot \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} A_i + \frac{1}{n_p} \sum_{i=1}^{n_p} B_i \geq 1 + \alpha + c\alpha^2 \right] \leq \\
& 2 \cdot \left(\Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} A_i \geq 1 + \alpha + (c-1)\alpha^2 \right] + \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} B_i \geq \alpha^2 \right] \right)
\end{aligned}$$

By Lemma 50 with $\lambda := \lambda'$ and $n := n_p$,

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

Finally, by Lemma 51 with $n := n_p$, $x := \alpha^2$ and by our assumption about w ,

$$\begin{aligned}
\Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} B_i \geq \alpha^2 \right] & \leq \frac{(w+2) \cdot e^{\alpha(w+1)}}{e \cdot (w+2 - e^\alpha) \cdot (w+1)! \cdot \alpha^2} = \\
& \frac{w^2 \cdot (w+2) \cdot e^{\frac{w+1}{w}}}{e \cdot (w+2 - e^{1/w}) \cdot (w+1)!} \leq \frac{2^{-\lambda}}{8}.
\end{aligned}$$

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 7, we conclude that $\Pr[Z \geq \delta du] \leq 2^{-\lambda}$. \square

Proof of Theorem 11. Let

$$\begin{aligned}
\alpha & = \sqrt{\frac{\lambda'}{3ud}}, \\
c & = 2 \left(\frac{1}{\alpha} + u \right) \sqrt{\frac{2\lambda'}{n_p}} + 3,
\end{aligned}$$

let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i , and let E be the event that $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{\alpha X_i} \leq e^{\alpha + c\alpha^2}$ with $\Pr[E] > 0$. By Lemma 48 with $\lambda := \lambda'$,

$$\Pr[Z \geq \delta du | E] \leq e^{-\lambda'} = \frac{2^{-\lambda}}{4}. \tag{8}$$

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

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

and

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

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

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

By Poisson approximation [MU05, Theorem 5.10],

$$\begin{aligned} & [\leq] 2 \cdot \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{\alpha Y_i} \geq 1 + \alpha + c\alpha^2\right] = \\ & 2 \cdot \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} (A_i + B_i) \geq 1 + \alpha + c\alpha^2\right] = \\ & 2 \cdot \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} A_i + \frac{1}{n_p} \sum_{i=1}^{n_p} B_i \geq 1 + \alpha + c\alpha^2\right] \leq \\ & 2 \cdot \left(\Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} A_i \geq 1 + \alpha + (c-1)\alpha^2\right] + \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} B_i \geq \alpha^2\right] \right) \end{aligned}$$

By Lemma 50 with $\lambda := \lambda'$, $w := u$ and $n := n_p$,

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

Finally, by Lemma 51 with $w := u$, $n := n_p$, $x := \alpha^2$ and by our assumption about u ,

$$\begin{aligned} \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} B_i \geq \alpha^2\right] & \leq \frac{(u+2) \cdot e^{\alpha(u+1)}}{e \cdot (u+2 - e^\alpha) \cdot (u+1)! \cdot \alpha^2} = \\ & \frac{3ud(u+2) \cdot e^{\frac{u+1}{u}}}{e \cdot \lambda' \cdot (u+2 - e^{1/u}) \cdot (u+1)!} \leq \frac{2^{-\lambda}}{8}. \end{aligned}$$

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 8, we conclude that $\Pr[Z \geq \delta du] \leq 2^{-\lambda}$. \square

Proof of Lemma 8. For $k \in \mathbb{Z}$ with $0 \leq k < u$, define

$$m_k = \left\lceil \frac{\lambda + \log u}{\log e} \cdot \frac{1}{q - ckq^2} \right\rceil,$$

let G_k be the event that the algorithm visits more than m_k vertices at height $u - k$ (root vertices, consisting of a single integer, being at height 0), and let F_k be the event that the algorithm entered at least m_k vertices at height $u - k$ and the first m_k of them were not prefixes of a valid certificate.

Since

$$\begin{aligned} \sum_{k=0}^{u-1} m_k &\leq \frac{\lambda + \log u}{\log e} \cdot \sum_{k=0}^{u-1} \frac{1}{q - ckq^2} + u \leq \\ \frac{\lambda + \log u}{\log e} \cdot \sum_{k=0}^{u-1} \left(\frac{1}{q} \cdot \frac{u-1-k}{u-1} + \frac{1}{q - c(u-1)q^2} \cdot \frac{k}{u-1} \right) + u &= \\ \frac{\lambda + \log u}{\log e} \cdot \frac{u}{2} \cdot \left(\frac{1}{q} + \frac{1}{q - c(u-1)q^2} \right) + u &\leq B, \end{aligned}$$

it follows that for all h_0 satisfying event E ,

$$\begin{aligned} \Pr[V > B | H_0 = h_0] &\leq \Pr \left[\bigvee_{k=0}^{u-1} G_k \middle| H_0 = h_0 \right] \leq \Pr \left[\bigvee_{k=0}^{u-1} F_k \middle| H_0 = h_0 \right] \leq \\ &\sum_{k=0}^{u-1} \Pr[F_k | H_0 = h_0] \leq \end{aligned}$$

and by Lemma 45,

$$\leq \sum_{k=0}^{u-1} \left(e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}} \right)^k \right)^{m_k}.$$

Hence,

$$\begin{aligned} \Pr[V > B | E] &= \mathbb{E} \left[\Pr[V > B | H_0, E] \middle| E \right] = \mathbb{E} \left[\Pr[V > B | H_0] \middle| E \right] = \\ &\mathbb{E} \left[\sum_{k=0}^{u-1} \left(e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}} \right)^k \right)^{m_k} \middle| f(e^{-q}) \leq e^{-q+ckq^2} \right] \leq \\ \sum_{k=0}^{u-1} \left(e^{-q+ckq^2} \right)^{m_k} &\leq \sum_{k=0}^{u-1} \exp \left(- \frac{\lambda + \log u}{\log e} \right) = \sum_{k=0}^{u-1} 2^{-\lambda}/u = 2^{-\lambda}. \end{aligned}$$

□

Proof of Theorem 14. Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin

i and let E be the event that $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+4q^2}$. By Lemma 5,

$$\begin{aligned} \Pr \left[\bar{E} \right] &= \\ \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} > e^{-q+4q^2} \right] &\leq \\ \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq 1 - q + 4q^2 \right] &\leq \\ &2e^{-\frac{9}{4}n_p q^2}. \end{aligned}$$

This is at most $\frac{1}{2} \cdot 2^{-\lambda_{\text{rel}}}$ if and only if

$$\begin{aligned} 4 \cdot 2^{\lambda_{\text{rel}}} &\leq e^{\frac{9}{4}n_p q^2} \iff \\ \frac{9}{4} \log e \cdot n_p q^2 &\geq \lambda_{\text{rel}} + 2 \iff \\ n_p &\geq \frac{4(\lambda_{\text{rel}} + 2)}{9 \log e \cdot q^2} \iff \\ n_p &\geq \frac{4(\lambda_{\text{rel}} + 2)}{9 \log e \cdot \left(\frac{2(\lambda_{\text{rel}} + 2)}{d \log e} \right)^2} \iff \\ n_p &\geq \frac{4(\lambda_{\text{rel}} + 2)}{9 \log e \cdot \frac{4(\lambda_{\text{rel}} + 2)^2}{d^2 \log^2 e}} \iff \\ n_p &\geq \frac{d^2 \log e}{9(\lambda_{\text{rel}} + 2)} \end{aligned}$$

which is true by our assumption about n_p .

Let F be the event that the honest prover fails. By Lemma 43 with $\lambda := \frac{\lambda_{\text{rel}} + 2}{\log e}$ and $c := 4$, $\Pr[F|E] \leq \frac{1}{4} \cdot 2^{-\lambda_{\text{rel}}}$.

Finally, let V be the number of non-root vertices that the algorithm visits and define

$$\begin{aligned} B &= \frac{u(\lambda_{\text{rel}} + 2 + \log u)}{2 \log e} \left(\frac{1}{q} + \frac{1}{q - 4uq^2} \right) + u; \\ B' &= \frac{\lambda_{\text{rel}} + 2 + \log u}{\lambda_{\text{rel}} + 2} \cdot \frac{3ud}{4} + u. \end{aligned}$$

By Lemma 8 with $\lambda := \lambda_{\text{rel}} + 2$ and $c := 4$, $\Pr[V > B|E] \leq \frac{1}{4} \cdot 2^{-\lambda_{\text{rel}}}$. Since

$$q - 4uq^2 = q(1 - 4uq) = q\left(1 - \frac{8u(\lambda_{\text{rel}}+2)}{d \log e}\right) \geq \frac{q}{2},$$

$$\begin{aligned} B &\leq \frac{u(\lambda_{\text{rel}} + 2 + \log u)}{2 \log e} \left(\frac{1}{q} + \frac{1}{\frac{q}{2}} \right) + u = \\ &\quad \frac{3u(\lambda_{\text{rel}} + 2 + \log u)}{2 \log e} \cdot \frac{1}{q} + u = \\ &\quad \frac{3u}{4} \cdot \frac{\lambda_{\text{rel}} + 2 + \log u}{\lambda_{\text{rel}} + 2} \cdot \frac{2(\lambda_{\text{rel}} + 2)}{q \log e} + u = B'. \end{aligned}$$

Therefore, $\Pr[V > B'|E] \leq \Pr[V > B|E] \leq \frac{1}{4} \cdot 2^{-\lambda_{\text{rel}}}$.

Combining the facts that $\Pr[\bar{E}] \leq \frac{1}{2} \cdot 2^{-\lambda_{\text{rel}}}$, $\Pr[F|E] \leq \frac{1}{4} \cdot 2^{-\lambda_{\text{rel}}}$, $\Pr[V > B'|E] \leq \frac{1}{4} \cdot 2^{-\lambda_{\text{rel}}}$, and given that there are exactly d root vertices, the theorem follows. \square

Proof of Corollary 3. Following Theorem 12, define

$$d^{(0)} = \lceil (32 \ln 12)u \rceil; \quad q^{(0)} = \frac{2 \ln 12}{d^{(0)}}; \quad B^{(0)} = \left\lfloor \frac{8(u+1)d^{(0)}}{\ln 12} \right\rfloor.$$

Following Theorem 13, define functions

$$\lambda_{\text{rel}}^{(1)'}(\lambda) = \frac{\lambda + 7}{\log e}; \quad d^{(1)}(\lambda) = \lceil 16u\lambda_{\text{rel}}^{(1)'}(\lambda) \rceil; \quad q^{(1)}(\lambda) = \frac{2\lambda_{\text{rel}}^{(1)'}(\lambda)}{d^{(1)}(\lambda)}.$$

Also define

$$\begin{aligned} S^{(1)} &= \left\{ \lambda : 1 \leq \lambda \leq \lambda_{\text{rel}} \wedge n_{\text{p}} \geq \frac{(17u\lambda_{\text{rel}}^{(1)'}(\lambda))^2}{9\lambda_{\text{rel}}^{(1)'}(\lambda)} \right\}; \\ \lambda_{\text{rel}}^{(1)} &= \begin{cases} \max S^{(1)} & \text{if } S^{(1)} \neq \emptyset \\ \perp & \text{otherwise.} \end{cases} \end{aligned}$$

If $\lambda_{\text{rel}}^{(1)} \neq \perp$, also define

$$\begin{aligned} \lambda_{\text{rel}}^{(1)'} &= \lambda_{\text{rel}}^{(1)'}(\lambda_{\text{rel}}^{(1)}); \quad d^{(1)} = d^{(1)}(\lambda_{\text{rel}}^{(1)}); \quad q^{(1)} = q^{(1)}(\lambda_{\text{rel}}^{(1)}); \\ w &= \min \left\{ w : w \in \mathbb{N} \wedge w \geq u \wedge \frac{14 \cdot w^2 \cdot (w+2) \cdot e^{\frac{w+1}{w}}}{e \cdot (w+2 - e^{1/w}) \cdot (w+1)!} \leq 2^{-\lambda_{\text{rel}}^{(1)}} \right\}; \\ B^{(1)} &= \left\lfloor \left(\frac{w\lambda_{\text{rel}}^{(1)'}}{d^{(1)}} + 1 \right) \cdot \exp \left(\frac{2uw\lambda_{\text{rel}}^{(1)'}}{n_{\text{p}}} + \frac{7u}{w} \right) \cdot d^{(1)}u + d^{(1)} \right\rfloor. \end{aligned}$$

Following Theorem 14, define functions

$$d^{(2)}(\lambda) = \left\lceil \frac{16u(\lambda+2)}{\log e} \right\rceil; \quad q^{(2)}(\lambda) = \frac{2(\lambda+2)}{d^{(2)}(\lambda) \log e}.$$

Also define

$$S^{(2)} = \left\{ \lambda : 1 \leq \lambda \leq \lambda_{\text{rel}} \wedge n_p \geq \left(\frac{17u(\lambda+2)}{\log e} \right)^2 \cdot \frac{\log e}{9(\lambda+2)} \right\};$$

$$\lambda_{\text{rel}}^{(2)} = \begin{cases} \max S^{(2)} & \text{if } S^{(2)} \neq \emptyset \\ \perp & \text{otherwise.} \end{cases}$$

If $\lambda_{\text{rel}}^{(2)} \neq \perp$, also define

$$d^{(2)} = d^{(2)}(\lambda_{\text{rel}}^{(2)}); \quad q^{(2)} = q^{(2)}(\lambda_{\text{rel}}^{(2)});$$

$$B^{(2)} = \left\lfloor \frac{\lambda_{\text{rel}}^{(2)} + 2 + \log u}{\lambda_{\text{rel}}^{(2)} + 2} \cdot \frac{3ud^{(2)}}{4} + d^{(2)} + u \right\rfloor.$$

Claim 2. *There exist constants $C_0 > 0$, $C_1 > 0$, $C_2 > 0$ independent of $\lambda_{\text{sec}}, \lambda_{\text{rel}}, n_p, n_f$ such that*

- if $\lambda_{\text{rel}}^{(1)} \neq \perp$, then
 - $\lambda_{\text{rel}}^{(1)} = \lambda_{\text{rel}}$ or $\lambda_{\text{rel}}^{(1)} \geq C_0 n_p / u^2$;
 - $\lambda_{\text{rel}}^{(1)} \leq C_1 n_p / u^2$;
- if $\lambda_{\text{rel}}^{(1)} = \perp$, then $n_p \leq C_2 u^2$.

Proof. Notice that

$$S^{(1)} = [1; \lambda_{\text{rel}}] \cap \left(-\infty; \frac{n_p}{Cu^2} - 7 \right]$$

where $C = \frac{17^2}{9 \log e}$.

Suppose $\lambda_{\text{rel}}^{(1)} \neq \perp$. Then, $\frac{n_p}{Cu^2} - 7 \geq \lambda_{\text{rel}}$ implies $\lambda_{\text{rel}}^{(1)} = \lambda_{\text{rel}}$. On the other hand, $\frac{n_p}{Cu^2} - 7 < \lambda_{\text{rel}}$ implies $\lambda_{\text{rel}}^{(1)} = \frac{n_p}{Cu^2} - 7$; combined with the fact that $\lambda_{\text{rel}}^{(1)} \geq 1$, it follows that $\lambda_{\text{rel}}^{(1)} \geq \frac{n_p}{8Cu^2}$. Additionally, $\lambda_{\text{rel}}^{(1)} \leq \frac{n_p}{Cu^2} - 7 \leq \frac{n_p}{Cu^2}$.

Now suppose $\lambda_{\text{rel}}^{(1)} = \perp$. Then $\frac{n_p}{Cu^2} - 7 < 1$ which implies $n_p \leq 8Cu^2$. \square

The following is proven in a similar way.

Claim 3. *There exist constants $C_3 > 0$, $C_4 > 0$, $C_5 > 0$ independent of $\lambda_{\text{sec}}, \lambda_{\text{rel}}, n_p, n_f$ such that*

- if $\lambda_{\text{rel}}^{(2)} \neq \perp$, then
 - $\lambda_{\text{rel}}^{(2)} = \lambda_{\text{rel}}$ or $\lambda_{\text{rel}}^{(2)} \geq C_3 n_p / u^2$;
 - $\lambda_{\text{rel}}^{(2)} \leq C_4 n_p / u^2$;
- if $\lambda_{\text{rel}}^{(2)} = \perp$, then $n_p \leq C_5 u^2$.

We now consider three cases.

Case 1. Suppose $(\lambda_{\text{rel}}^{(1)} = \perp) \vee (\lambda_{\text{rel}}^{(2)} = \perp)$. We set

$$d = d^{(0)}; \quad q = q^{(0)}; \quad B = B^{(0)}; \quad r = \lceil \lambda_{\text{rel}} \rceil.$$

By Theorem 12, the DFS fails with probability $\leq \frac{1}{2}$. Therefore, the scheme has completeness error $\leq \left(\frac{1}{2}\right)^r \leq 2^{-\lambda_{\text{rel}}}$. By Lemma 3, the soundness error of the scheme is at most $\left(\frac{n_f}{n_p}\right)^u \cdot qd = \left(\frac{n_f}{n_p}\right)^u \cdot 2 \ln 12 \leq 2^{-\lambda_{\text{sec}}}$. We will now analyze the expected prover running time.

$$B \leq \frac{8(u+1)d}{\ln 12} = O(ud) = O\left(u \cdot \lceil (32 \ln 12)u \rceil\right) = O(u^2).$$

Also by Claim 2 and Claim 3, $n_p \leq \max\{C_2, C_5\}u^2 = O(u^2)$. By Lemma 35, the expected running time is at most $2(n_p + B) = O(u^2)$. Finally, the worst case running time is bounded by

$$(n_p + B) \cdot r = O(u^2) \cdot \lceil \lambda_{\text{rel}} \rceil = O(u^2 \cdot \lambda_{\text{rel}}).$$

Case 2. Suppose $(\lambda_{\text{rel}}^{(1)} \neq \perp) \wedge (\lambda_{\text{rel}}^{(2)} \neq \perp) \wedge (u \geq \lambda_{\text{rel}}^{(2)})$. We set

$$d = d^{(1)}; \quad q = q^{(1)}; \quad B = B^{(1)}; \quad r = \lceil \lambda_{\text{rel}} / \lambda_{\text{rel}}^{(1)} \rceil.$$

By Theorem 13, the DFS fails with probability $\leq 2^{-\lambda_{\text{rel}}^{(1)}}$. Therefore, the completeness error of the scheme is at most

$$\left(2^{-\lambda_{\text{rel}}^{(1)}}\right)^r \leq 2^{-\lambda_{\text{rel}}}.$$

By Lemma 3, the soundness error of the scheme is at most

$$\begin{aligned} r \cdot \left(\frac{n_f}{n_p}\right)^u \cdot qd &= \\ \lceil \lambda_{\text{rel}} / \lambda_{\text{rel}}^{(1)} \rceil \cdot \left(\frac{n_f}{n_p}\right)^u \cdot 2\lambda_{\text{rel}}^{(1)'} &= \\ \lceil \lambda_{\text{rel}} / \lambda_{\text{rel}}^{(1)} \rceil \cdot \left(\frac{n_f}{n_p}\right)^u \cdot 2 \cdot \frac{\lambda_{\text{rel}}^{(1)} + 7}{\log e} &\leq \\ \frac{2\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(1)}} \cdot \left(\frac{n_f}{n_p}\right)^u \cdot 2 \cdot \frac{8\lambda_{\text{rel}}^{(1)}}{\log e} &= \\ \left(\frac{n_f}{n_p}\right)^u \cdot \lambda_{\text{rel}} \cdot \frac{32}{\log e} &\leq \\ 2^{-\lambda_{\text{sec}}}. \end{aligned}$$

We will now analyze the expected prover running time. Notice that

$$d = \lceil 16u\lambda_{\text{rel}}^{(1)'} \rceil = O\left(u\lambda_{\text{rel}}^{(1)'}\right) = O\left(u \cdot \frac{\lambda_{\text{rel}}^{(1)} + 7}{\log e}\right) = O\left(u \cdot \lambda_{\text{rel}}^{(1)}\right)$$

and

$$q = \frac{2\lambda_{\text{rel}}^{(1)'}}{d} = \frac{2\lambda_{\text{rel}}^{(1)'}}{\lceil 16u\lambda_{\text{rel}}^{(1)'} \rceil} = \frac{2\lambda_{\text{rel}}^{(1)'}}{O(u\lambda_{\text{rel}}^{(1)'})} = \Omega\left(\frac{1}{u}\right).$$

By Theorem 9, the expected number of vertices each DFS visits is at most

$$\begin{aligned} & \frac{4(u+1)}{q} + 2e^{-\frac{9}{4}n_p q^2} \cdot d(u+1) = \\ & O(u^2) + e^{-\frac{9}{4}n_p q^2} \cdot O(u^2 \cdot \lambda_{\text{rel}}^{(1)}) \quad [=] \end{aligned}$$

By Claim 2, $\lambda_{\text{rel}}^{(1)} \leq C_1 n_p / u^2$, and thus $n_p = \Omega(u^2 \lambda_{\text{rel}}^{(1)})$; therefore,

$$[=] O(u^2) + \exp\left(-\Omega\left(\lambda_{\text{rel}}^{(1)}\right)\right) \cdot O(u^2 \cdot \lambda_{\text{rel}}^{(1)}) = O(u^2).$$

If $\lambda_{\text{rel}}^{(1)} = \lambda_{\text{rel}}$, then $r = 1$ and the expected running time is at most $n_p + O(u^2)$. Otherwise, by Lemma 35, the expected running time is at most

$$\frac{n_p + O(u^2)}{1 - 2^{-\lambda_{\text{rel}}^{(1)}}} = \frac{n_p}{1 - 2^{-\lambda_{\text{rel}}^{(1)}}} + O(u^2) \quad [=]$$

By Lemma 32,

$$\begin{aligned} & [=] n_p \left(1 + 2 \cdot 2^{-\lambda_{\text{rel}}^{(1)}}\right) + O(u^2) = \\ & n_p + 2 \cdot n_p \cdot 2^{-\lambda_{\text{rel}}^{(1)}} + O(u^2) \quad [=] \end{aligned}$$

By Claim 2, $\lambda_{\text{rel}}^{(1)} \geq C_0 n_p / u^2$; thus,

$$\begin{aligned} & [=] n_p + \frac{2u^2 \lambda_{\text{rel}}^{(1)}}{C_0} \cdot 2^{-\lambda_{\text{rel}}^{(1)}} + O(u^2) = \\ & n_p + O(u^2). \end{aligned}$$

Finally, we will analyze the worst case running time. By Claim 3, $\lambda_{\text{rel}}^{(2)} = \lambda_{\text{rel}}$ or $\lambda_{\text{rel}}^{(2)} \geq C_3 n_p / u^2$. If $\lambda_{\text{rel}}^{(2)} = \lambda_{\text{rel}}$, then $w \geq u \geq \lambda_{\text{rel}}^{(2)} = \lambda_{\text{rel}} \geq \lambda_{\text{rel}}^{(1)}$; if $\lambda_{\text{rel}}^{(2)} \geq C_3 n_p / u^2$, then $w \geq u \geq \lambda_{\text{rel}}^{(2)} \geq C_3 n_p / u^2 = \frac{C_3}{C_1} \cdot C_1 n_p / u^2 \geq \frac{C_3}{C_1} \cdot \lambda_{\text{rel}}^{(1)}$. In either case, it can be seen by the definition of w that $w \leq u + C_6$ for some constant C_6 . Prover's worst

case running time is bounded by $(n_p + B) \cdot r$.

$$\begin{aligned}
B \cdot r &= \\
&\left[\left(\frac{w\lambda_{\text{rel}}^{(1)'}}{d} + 1 \right) \cdot \exp \left(\frac{2uw\lambda_{\text{rel}}^{(1)'}}{n_p} + \frac{7u}{w} \right) \cdot du + d \right] \cdot \left\lceil \lambda_{\text{rel}} / \lambda_{\text{rel}}^{(1)} \right\rceil = \\
&\left(\frac{w\lambda_{\text{rel}}^{(1)'}}{d} + 1 \right) \cdot \exp \left(\frac{2uw\lambda_{\text{rel}}^{(1)'}}{n_p} + \frac{7u}{w} \right) \cdot du \cdot O \left(\frac{\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(1)}} \right) = \\
&\left(\frac{w\lambda_{\text{rel}}^{(1)'}}{\left\lceil 16u\lambda_{\text{rel}}^{(1)'} \right\rceil} + 1 \right) \cdot \exp \left(\frac{2uw\lambda_{\text{rel}}^{(1)'}}{n_p} + \frac{7u}{w} \right) \cdot O(u^2 \lambda_{\text{rel}}^{(1)}) \cdot O \left(\frac{\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(1)}} \right) = \\
&\left(\frac{w\lambda_{\text{rel}}^{(1)'}}{16u\lambda_{\text{rel}}^{(1)'}} + 1 \right) \cdot \exp \left(\frac{2uw\lambda_{\text{rel}}^{(1)'}}{n_p} + \frac{7u}{w} \right) \cdot O(u^2 \cdot \lambda_{\text{rel}}) = \\
&\left(\frac{u + C_6}{16u} + 1 \right) \cdot \exp \left(\frac{2u(u + C_6)\lambda_{\text{rel}}^{(1)'}}{n_p} + \frac{7u}{u} \right) \cdot O(u^2 \cdot \lambda_{\text{rel}}) = \\
&\exp \left(\frac{2u(u + C_6)\lambda_{\text{rel}}^{(1)'}}{n_p} \right) \cdot O(u^2 \cdot \lambda_{\text{rel}}).
\end{aligned}$$

Since $1 \leq \lambda_{\text{rel}}^{(1)} \leq C_1 n_p / u^2$, $\lambda_{\text{rel}}^{(1)'}$ $= (\lambda_{\text{rel}}^{(1)} + 7) / \log e \leq 8\lambda_{\text{rel}}^{(1)} / \log e = O(n_p / u^2)$, and thus $B \cdot r = O(u^2 \cdot \lambda_{\text{rel}})$. Additionally, by Claim 2, $\lambda_{\text{rel}}^{(1)} = \lambda_{\text{rel}}$ or $\lambda_{\text{rel}}^{(1)} \geq C_0 n_p / u^2$. In the first case, $r = 1$ and $n_p \cdot r = n_p$. In the second case,

$$n_p \cdot r = n_p \cdot \left\lceil \lambda_{\text{rel}} / \lambda_{\text{rel}}^{(1)} \right\rceil \leq 2 \cdot n_p \cdot \frac{\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(1)}} \leq 2 \cdot n_p \cdot \frac{\lambda_{\text{rel}}}{C_0 n_p / u^2} = \frac{2u^2 \lambda_{\text{rel}}}{C_0}.$$

In either case, $n_p \cdot r = n_p + O(u^2 \cdot \lambda_{\text{rel}})$. Hence, the worst case running time is $(n_p + B) \cdot r = n_p + O(u^2 \cdot \lambda_{\text{rel}})$.

Case 3. Suppose $(\lambda_{\text{rel}}^{(1)} \neq \perp) \wedge (\lambda_{\text{rel}}^{(2)} \neq \perp) \wedge (u < \lambda_{\text{rel}}^{(2)})$. We set

$$d = d^{(2)}; \quad q = q^{(2)}; \quad B = B^{(2)}; \quad r = \left\lceil \lambda_{\text{rel}} / \lambda_{\text{rel}}^{(2)} \right\rceil.$$

By Theorem 14, the DFS fails with probability $\leq 2^{-\lambda_{\text{rel}}^{(2)}}$. Therefore, the completeness error of the scheme is at most

$$\left(2^{-\lambda_{\text{rel}}^{(2)}} \right)^r \leq 2^{-\lambda_{\text{rel}}}.$$

By Lemma 3, the soundness error of the scheme is at most

$$\begin{aligned}
& r \cdot \left(\frac{n_f}{n_p} \right)^u \cdot qd = \\
& \left\lceil \lambda_{\text{rel}} / \lambda_{\text{rel}}^{(2)} \right\rceil \cdot \left(\frac{n_f}{n_p} \right)^u \cdot \frac{2(\lambda_{\text{rel}}^{(2)} + 2)}{\log e} \leq \\
& \frac{2\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(2)}} \cdot \left(\frac{n_f}{n_p} \right)^u \cdot \frac{6\lambda_{\text{rel}}^{(2)}}{\log e} = \\
& \left(\frac{n_f}{n_p} \right)^u \cdot \lambda_{\text{rel}} \cdot \frac{12}{\log e} \leq \\
& 2^{-\lambda_{\text{sec}}}.
\end{aligned}$$

We will now analyze the expected prover running time. Notice that

$$d = \left\lceil \frac{16u(\lambda_{\text{rel}}^{(2)} + 2)}{\log e} \right\rceil = O\left(u \cdot \lambda_{\text{rel}}^{(2)}\right)$$

and

$$q = \frac{2(\lambda_{\text{rel}}^{(2)} + 2)}{d \log e} = \frac{2(\lambda_{\text{rel}}^{(2)} + 2)}{\left\lceil \frac{16u(\lambda_{\text{rel}}^{(2)} + 2)}{\log e} \right\rceil \cdot \log e} = \frac{2(\lambda_{\text{rel}}^{(2)} + 2)}{O\left(\frac{u(\lambda_{\text{rel}}^{(2)} + 2)}{\log e}\right) \cdot \log e} = \Omega\left(\frac{1}{u}\right).$$

By Theorem 9, the expected number of vertices each DFS visits is at most

$$\begin{aligned}
& \frac{4(u+1)}{q} + 2e^{-\frac{9}{4}n_p q^2} \cdot d(u+1) = \\
& O(u^2) + e^{-\frac{9}{4}n_p q^2} \cdot O\left(u^2 \cdot \lambda_{\text{rel}}^{(2)}\right) [=]
\end{aligned}$$

By Claim 3, $\lambda_{\text{rel}}^{(2)} \leq C_4 n_p / u^2$ and thus $n_p = \Omega(u^2 \cdot \lambda_{\text{rel}}^{(2)})$; therefore,

$$[=] O(u^2) + \exp\left(-\Omega\left(\lambda_{\text{rel}}^{(2)}\right)\right) \cdot O\left(u^2 \cdot \lambda_{\text{rel}}^{(2)}\right) = O(u^2).$$

If $\lambda_{\text{rel}}^{(2)} = \lambda_{\text{rel}}$, then $r = 1$ and the expected running time is at most $n_p + O(u^2)$. Otherwise, by Lemma 35, the expected running time is at most

$$\frac{n_p + O(u^2)}{1 - 2^{-\lambda_{\text{rel}}^{(2)}}} = \frac{n_p}{1 - 2^{-\lambda_{\text{rel}}^{(2)}}} + O(u^2) [=]$$

By Lemma 32,

$$\begin{aligned}
& [=] n_p \left(1 + 2 \cdot 2^{-\lambda_{\text{rel}}^{(2)}}\right) + O(u^2) = \\
& n_p + 2 \cdot n_p \cdot 2^{-\lambda_{\text{rel}}^{(2)}} + O(u^2) [=]
\end{aligned}$$

By Claim 2, $\lambda_{\text{rel}}^{(2)} \geq C_3 n_p / u^2$; thus,

$$\begin{aligned} [=] n_p + \frac{2u^2 \lambda_{\text{rel}}^{(2)}}{C_3} \cdot 2^{-\lambda_{\text{rel}}^{(2)}} + O(u^2) &= \\ n_p + O(u^2). \end{aligned}$$

Prover's worst case running time is bounded by $(n_p + B) \cdot r$.

$$\begin{aligned} B \cdot r &= \\ \left[\frac{\lambda_{\text{rel}}^{(2)} + 2 + \log u}{\lambda_{\text{rel}}^{(2)} + 2} \cdot \frac{3ud}{4} + d + u \right] \cdot \left[\lambda_{\text{rel}} / \lambda_{\text{rel}}^{(2)} \right] &= \\ \left(\frac{\lambda_{\text{rel}}^{(2)} + 2 + \log u}{\lambda_{\text{rel}}^{(2)} + 2} \cdot \frac{3ud}{4} + d + u \right) \cdot O\left(\frac{\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(2)}} \right) &= \\ \left(\frac{\lambda_{\text{rel}}^{(2)} + 2 + \log u}{\lambda_{\text{rel}}^{(2)} + 2} \cdot O(u^2 \cdot \lambda_{\text{rel}}^{(2)}) \right) \cdot O\left(\frac{\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(2)}} \right) &= \\ O\left(\frac{\lambda_{\text{rel}}^{(2)} + 2 + \log u}{\lambda_{\text{rel}}^{(2)} + 2} \cdot u^2 \cdot \lambda_{\text{rel}} \right) &= \\ O\left(\frac{\lambda_{\text{rel}}^{(2)} + 2 + \log \lambda_{\text{rel}}^{(2)}}{\lambda_{\text{rel}}^{(2)} + 2} \cdot u^2 \cdot \lambda_{\text{rel}} \right) &= \\ O(u^2 \cdot \lambda_{\text{rel}}). \end{aligned}$$

Additionally, by Claim 3, $\lambda_{\text{rel}}^{(2)} = \lambda_{\text{rel}}$ or $\lambda_{\text{rel}}^{(2)} \geq C_3 n_p / u^2$. In the first case, $r = 1$ and $n_p \cdot r = n_p$. In the second case,

$$n_p \cdot r = n_p \cdot \left[\lambda_{\text{rel}} / \lambda_{\text{rel}}^{(2)} \right] \leq 2 \cdot n_p \cdot \frac{\lambda_{\text{rel}}}{\lambda_{\text{rel}}^{(2)}} \leq 2 \cdot n_p \cdot \frac{\lambda_{\text{rel}}}{C_3 n_p / u^2} = \frac{2u^2 \lambda_{\text{rel}}}{C_3}.$$

In either case, $n_p \cdot r = n_p + O(u^2 \cdot \lambda_{\text{rel}})$. Hence, the worst case running time is $(n_p + B) \cdot r = n_p + O(u^2 \cdot \lambda_{\text{rel}})$. \square

Proof of Theorem 15. Suppose not and define $u = \lfloor \alpha \rfloor$. Then $\Pr \left[\left| \text{Read}(\text{Prove}^H(S_p)) \right| \leq u \right] \geq \frac{3}{4}$.

Let $\pi \leftarrow \text{Prove}^H(S_p)$, S_f be a uniformly random subset of S_p of size n_f , A be the event that $|\text{Read}(\pi)| \leq u$ and B the event that $\text{Read}(\pi) \subseteq S_p \wedge \text{Verify}^H(\pi) = 1$. By

the above, $\Pr[A] \geq \frac{3}{4}$, and by completeness, $\Pr[B] \geq \frac{1}{2}$. Then

$$\begin{aligned}
& \Pr[\text{Read}(\pi) \subseteq S_f \wedge \text{Verify}^H(\pi) = 1] \geq \\
& \Pr[\text{Read}(\pi) \subseteq S_f \wedge \text{Verify}^H(\pi) = 1 | A \wedge B] \cdot \Pr[A \wedge B] = \\
& \Pr[\text{Read}(\pi) \subseteq S_f | A \wedge B] \cdot \Pr[A \wedge B] \geq \\
& \Pr[\text{Read}(\pi) \subseteq S_f | A \wedge B] \cdot (\Pr[A] + \Pr[B] - 1) \geq \\
& \Pr[\text{Read}(\pi) \subseteq S_f | A \wedge B] \cdot \left(\frac{3}{4} + \frac{1}{2} - 1\right) = \\
& \frac{1}{4} \cdot \Pr[\text{Read}(\pi) \subseteq S_f | A \wedge B] \geq \\
& \frac{1}{4} \cdot \frac{n_f}{n_p} \cdot \frac{n_f - 1}{n_p - 1} \times \dots \times \frac{n_f - (u - 1)}{n_p - (u - 1)} \geq \\
& \frac{1}{4} \cdot \left(\frac{n_f - u}{n_p}\right)^u = \\
& \frac{1}{4} \cdot \left(\frac{n_f}{n_p}\right)^u \cdot \left(\frac{n_f - u}{n_f}\right)^u = \\
& \frac{1}{4} \cdot \left(\frac{n_f}{n_p}\right)^u \cdot \left(1 - \frac{u}{n_f}\right)^u [\geq]
\end{aligned}$$

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

$$\begin{aligned}
& [\geq] \frac{1}{4} \cdot \left(\frac{n_f}{n_p}\right)^u \cdot \left(e^{-\frac{3u}{2n_f}}\right)^u \geq \\
& \frac{1}{4} \cdot \left(\frac{n_f}{n_p}\right)^u \cdot e^{-\frac{3u^2}{6u^2}} > \\
& \frac{1}{8} \cdot \left(\frac{n_f}{n_p}\right)^u.
\end{aligned}$$

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

$$\Pr[\text{Read}(\pi) \subseteq S'_f \wedge \text{Verify}^H(\pi) = 1] > \frac{1}{8} \cdot \left(\frac{n_f}{n_p}\right)^u.$$

On the other hand, by soundness from Definition 4)

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

Thus,

$$\begin{aligned} \frac{1}{8} \cdot \left(\frac{n_f}{n_p} \right)^u &< 2^{-\lambda_{\text{sec}}} \iff \\ \left(\frac{n_p}{n_f} \right)^u &> 2^{\lambda_{\text{sec}}-3} \iff \\ u \log \frac{n_p}{n_f} &> \lambda_{\text{sec}} - 3 \iff \\ u &> \alpha, \end{aligned}$$

which is a contradiction. \square

Proof of Lemma 9. Let $S_f = \{s_1, \dots, s_{n_f}\}$ be malicious prover's set and define $X_i = H(s_i)$. To violate soundness, the malicious prover needs $\sum X_i \geq u = r_s \cdot pn_f$, while the expectation $\mathbb{E} \sum X_i = pn_f$. By Chernoff bound (Lemma 37) (with $\delta = r_s - 1$),

$$\Pr \left[\sum X_i \geq u \right] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^{pn_f} = \left(\frac{e^{r_s-1}}{r_s^{r_s}} \right).$$

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

$$\begin{aligned} pn_f(r_s - 1 - r_s \ln r_s) &\leq -\lambda_{\text{sec}} \cdot \ln 2 \iff \\ pn_f(r_s \ln r_s - r_s + 1) &\geq \lambda_{\text{sec}} \cdot \ln 2 \iff \\ u \left(\ln r_s - 1 + \frac{1}{r_s} \right) &\geq \lambda_{\text{sec}} \cdot \ln 2 \iff \\ u &\geq \frac{\lambda_{\text{sec}} \cdot \ln 2}{\ln r_s - 1 + \frac{1}{r_s}} \end{aligned}$$

which is true by our assumption about u . \square

Proof of Lemma 10. Let $S_p = \{s_1, \dots, s_{n_p}\}$ be honest prover's set and define $X_i = H(s_i)$. The honest prover fails whenever $\sum X_i < u = \frac{pn_p}{r_c}$, while the expectation $\mathbb{E} \sum X_i = pn_p$. By Chernoff bound (Lemma 39) (with $\delta = 1 - \frac{1}{r_c}$),

$$\Pr \left[\sum X_i < u \right] \leq \Pr \left[\sum X_i \leq u \right] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right)^{pn_p} = \left(\frac{e^{\frac{1}{r_c}-1}}{\left(\frac{1}{r_c}\right)^{\frac{1}{r_c}}} \right)^{pn_p}.$$

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

$$\begin{aligned} pn_p \left(\frac{1}{r_c} - 1 - \frac{1}{r_c} \cdot \ln \frac{1}{r_c} \right) &\leq -\lambda_{\text{rel}} \cdot \ln 2 \iff \\ pn_p \left(1 - \frac{1}{r_c} - \frac{1}{r_c} \cdot \ln r_c \right) &\geq \lambda_{\text{rel}} \cdot \ln 2 \iff \\ u(r_c - 1 - \ln r_c) &\geq \lambda_{\text{rel}} \cdot \ln 2 \iff \\ u &\geq \frac{\lambda_{\text{rel}} \cdot \ln 2}{r_c - 1 - \ln r_c} \end{aligned}$$

which is true by our assumption about u . \square

Proof of Lemma 11. 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, there are l choices for the other $(u-l)$ positions, there are d choices for the tuple's integer t and there are r choice's for the tuple's integer v . Overall, the number of tuples with exactly l distinct elements is at most

$$rd \cdot C(u, l) \cdot P(n_f, l) \cdot l^{u-l}.$$

Then by union bound the probability that a valid proof can be constructed using n_f elements is at most

$$\begin{aligned} & \sum_{l=1}^u \left(\left(\frac{\mu}{n_p} \right)^l \cdot \left(\frac{1}{\rho} \right)^u \cdot q \cdot rd \cdot C(u, l) \cdot P(n_f, l) \cdot l^{u-l} \right) = \\ & \left(\frac{1}{\rho} \right)^u \cdot qdr \cdot \sum_{l=1}^u \left(\left(\frac{\mu}{n_p} \right)^l \cdot \frac{u!}{l!(u-l)!} \cdot \frac{n_f!}{(n_f-l)!} \cdot l^{u-l} \right) \leq \\ & \left(\frac{1}{\rho} \right)^u \cdot qdr \cdot \sum_{l=1}^u \left(\left(\frac{\mu}{n_p} \right)^l \cdot \frac{u^{u-l}}{(u-l)!} \cdot n_f^l \cdot u^{u-l} \right) = \\ & \left(\frac{1}{\rho} \right)^u \cdot qdr \cdot \sum_{l=1}^u \left(\left(\frac{\mu n_f}{n_p} \right)^l \cdot \frac{u^{2(u-l)}}{(u-l)!} \right) = \\ & \left(\frac{1}{\rho} \right)^u \cdot qdr \cdot \left(\frac{\mu n_f}{n_p} \right)^u \cdot \sum_{l=1}^u \left(\left(\frac{\mu n_f}{n_p} \right)^{l-u} \cdot \frac{u^{2(u-l)}}{(u-l)!} \right) = \\ & qdr \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \sum_{l=1}^u \left(\left(\frac{u^2 n_p}{\mu n_f} \right)^{u-l} \cdot \frac{1}{(u-l)!} \right) \leq \\ & qdr \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \sum_{i=0}^{\infty} \left(\left(\frac{u^2 n_p}{\mu n_f} \right)^i \cdot \frac{1}{i!} \right) = \\ & qdr \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u^2 n_p}{\mu n_f} \right). \end{aligned}$$

□

Proof of Theorem 17. By Lemma 11 and our assumption about μ , the soundness error is at most

$$\begin{aligned} & qdr \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(\frac{u^2 n_p}{\mu n_f} \right) \leq \\ & eqdr \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u = \\ & 2e \ln 12 \cdot (\lambda_{\text{rel}} + 1) \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u. \end{aligned}$$

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

$$u \geq \frac{\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log \frac{\rho}{\mu}}.$$

□

Proof of Corollary 4. Completeness follows from Theorem 16. We only need to show that $\frac{\rho n_{\text{p}}}{\mu n_{\text{f}}} > 1$ and

$$u \geq \frac{\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log \frac{\rho}{\mu}}$$

to prove soundness using Theorem 17.

Since by our assumption $\mu \geq \frac{8(\lambda_{\text{rel}}+1)}{\log e}$, $\delta \leq \frac{1}{2}$ and $1 - \delta \geq e^{-\frac{3}{2}\delta}$ by Lemma 29. Then

$$\begin{aligned} \log \frac{\rho n_{\text{p}}}{\mu n_{\text{f}}} &= \log \frac{n_{\text{p}}}{n_{\text{f}}} + \log \frac{\rho}{\mu} \geq \log \frac{n_{\text{p}}}{n_{\text{f}}} + \log(1 - \delta) \geq \\ &\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log e^{-\frac{3}{2}\delta} = \log \frac{n_{\text{p}}}{n_{\text{f}}} - \frac{3}{2}\delta \log e > 0 \end{aligned} \tag{9}$$

where the last inequality follows from

$$\begin{aligned} \delta &< \frac{2}{3 \log e} \log \frac{n_{\text{p}}}{n_{\text{f}}} \iff \\ \delta^2 &< \left(\frac{2}{3 \log e} \log \frac{n_{\text{p}}}{n_{\text{f}}} \right)^2 \iff \\ \frac{2(\lambda_{\text{rel}} + 1)}{\mu \log e} &< \left(\frac{2}{3 \log e} \log \frac{n_{\text{p}}}{n_{\text{f}}} \right)^2 \iff \\ \mu &> \frac{2(\lambda_{\text{rel}} + 1)}{\left(\frac{2}{3 \log e} \log \frac{n_{\text{p}}}{n_{\text{f}}} \right)^2 \log e} \iff \\ \mu &> \frac{9 \log e}{2} \cdot \frac{\lambda_{\text{rel}} + 1}{\log^2 \frac{n_{\text{p}}}{n_{\text{f}}}}. \end{aligned}$$

Hence, $\frac{\rho n_{\text{p}}}{\mu n_{\text{f}}} > 1$ and we are left to show that

$$u \geq \frac{\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_{\text{p}}}{n_{\text{f}}} + \log \frac{\rho}{\mu}}.$$

Define $\lambda' = \lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12$. Then, also using Equation 9,

this inequality is equivalent to

$$\begin{aligned}
u &\geq \frac{\lambda'}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}} \iff \\
\frac{\lambda'}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}} &\leq \frac{\lambda' + C}{\log \frac{n_p}{n_f}} \iff \\
\frac{\lambda'}{\log \frac{n_p}{n_f} - \frac{3}{2}\delta \log e} &\leq \frac{\lambda' + C}{\log \frac{n_p}{n_f}} \iff \\
\lambda' \log \frac{n_p}{n_f} &\leq (\lambda' + C) \left(\log \frac{n_p}{n_f} - \frac{3}{2}\delta \log e \right) \iff \\
\frac{3}{2}\delta \log e (\lambda' + C) &\leq C \log \frac{n_p}{n_f} \iff \\
\delta &\leq \frac{2C \log \frac{n_p}{n_f}}{3(\lambda' + C) \log e} \iff \\
\delta^2 &\leq \left(\frac{2C \log \frac{n_p}{n_f}}{3(\lambda' + C) \log e} \right)^2 \iff \\
\frac{2(\lambda_{\text{rel}} + 1)}{\mu \log e} &\leq \left(\frac{2C \log \frac{n_p}{n_f}}{3(\lambda' + C) \log e} \right)^2 \iff \\
\mu &\geq \frac{2(\lambda_{\text{rel}} + 1)}{\log e} \left(\frac{3(\lambda' + C) \log e}{2C \log \frac{n_p}{n_f}} \right)^2 \iff \\
\mu &\geq \frac{9(\lambda_{\text{rel}} + 1) \log e}{2C^2} \left(\frac{\lambda' + C}{\log \frac{n_p}{n_f}} \right)^2 \iff \\
\mu &\geq \frac{9u^2(\lambda_{\text{rel}} + 1) \log e}{2C^2}
\end{aligned}$$

which is true by our assumption about μ . \square

Proof of Corollary 5. Completeness follows from Theorem 16. The proof of Corollary 4 shows that the assumption

$$\mu \geq \frac{18 \log e \cdot (\lambda_{\text{rel}} + 1)}{\log^2 \frac{n_p}{n_f}} > \frac{9 \log e}{2} \cdot \frac{\lambda_{\text{rel}} + 1}{\log^2 \frac{n_p}{n_f}}$$

implies

$$\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu} \geq \log \frac{n_p}{n_f} - \frac{3}{2}\delta \log e > 0$$

and thus $\frac{\rho n_p}{\mu n_f} > 1$. Hence, we only need to show

$$u \geq \frac{\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}} \quad (10)$$

to prove soundness using Theorem 17.

Additionally, one can see that $\frac{3\delta \log e}{2 \log(n_p/n_f)} \leq \frac{1}{2}$ is true since it is equivalent to

$$\begin{aligned} \delta &\leq \frac{\log \frac{n_p}{n_f}}{3 \log e} \iff \\ \frac{2(\lambda_{\text{rel}} + 1)}{\mu \log e} &\leq \left(\frac{\log \frac{n_p}{n_f}}{3 \log e} \right)^2 \iff \\ \mu &\geq \frac{18 \log e \cdot (\lambda_{\text{rel}} + 1)}{\log \frac{n_p}{n_f}}. \end{aligned}$$

Define $\lambda' = \lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12$. Equation 10 follows from

$$\begin{aligned} \frac{\lambda_{\text{sec}} + \log(\lambda_{\text{rel}} + 1) + 1 + \log e + \log \ln 12}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}} &= \\ \frac{\lambda'}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}} &\leq \\ \frac{\lambda'}{\log \frac{n_p}{n_f} - \frac{3}{2} \delta \log e} &= \\ \frac{\lambda'}{\log \frac{n_p}{n_f} \left(1 - \frac{3\delta \log e}{2 \log(n_p/n_f)} \right)} &[\leq] \end{aligned}$$

Lemma 32 implies

$$\begin{aligned} &[\leq] \left(1 + \frac{3\delta \log e}{\log \frac{n_p}{n_f}} \right) \cdot \frac{\lambda'}{\log \frac{n_p}{n_f}} = \\ &\left(1 + \frac{3 \log e}{\log \frac{n_p}{n_f}} \cdot \sqrt{\frac{2(\lambda_{\text{rel}} + 1)}{\mu \log e}} \right) \cdot \frac{\lambda'}{\log \frac{n_p}{n_f}} = \\ &\left(1 + \frac{3\sqrt{2} \log e \cdot \sqrt{\lambda_{\text{rel}} + 1}}{\sqrt{\mu} \cdot \log \frac{n_p}{n_f}} \right) \cdot \frac{\lambda'}{\log \frac{n_p}{n_f}} \leq \end{aligned}$$

u.

□

Proof of Theorem 18. By completeness, if $\pi \leftarrow \text{Prove}^H(\text{Lottery}(S_p))$, then

$$\begin{aligned}
& 2^{-\lambda_{\text{rel}}} \geq \\
& \Pr \left[\neg(\text{Read}(\pi) \subseteq \text{Lottery}(S_p) \wedge \text{Verify}^H(\pi) = 1) \right] \geq \\
& \Pr \left[\neg(\text{Read}(\pi) \subseteq \text{Lottery}(S_p) \wedge \text{Verify}^H(\pi) = 1) \mid |\text{Lottery}(S_p)| \leq \rho \right] \times \\
& \Pr \left[|\text{Lottery}(S_p)| \leq \rho \right] \geq \\
& \Pr \left[\neg(\text{Read}(\pi) \subseteq \text{Lottery}(S_p) \wedge \text{Verify}^H(\pi) = 1) \mid |\text{Lottery}(S_p)| \leq \rho \right] \cdot 2^{-\lambda_{\text{rel}}+1}.
\end{aligned}$$

Therefore,

$$\Pr \left[\neg(\text{Read}(\pi) \subseteq \text{Lottery}(S_p) \wedge \text{Verify}^H(\pi) = 1) \mid |\text{Lottery}(S_p)| \leq \rho \right] \leq \frac{1}{2}$$

and

$$\Pr \left[\text{Read}(\pi) \subseteq \text{Lottery}(S_p) \wedge \text{Verify}^H(\pi) = 1 \mid |\text{Lottery}(S_p)| \leq \rho \right] \geq \frac{1}{2}.$$

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

$$\Pr \left[\text{Read}(\pi) \subseteq \text{Lottery}(S_p) \wedge \text{Verify}^H(\pi) = 1 \mid |\text{Lottery}(S_p)| = m \right] \geq \frac{1}{2}. \quad (11)$$

Now for all $S_f \subseteq S_p$ of size n_f , define

procedure $\mathcal{A}_{S_f}^{L,H}$

- $S \leftarrow \text{Lottery}(S_f)$;
- if** $m < |S|$ **then**
- remove $(|S| - m)$ random elements from S ;
- else**
- add $(m - |S|)$ random elements from $S_f \setminus S$ to S ;
- $\pi \leftarrow \text{Prove}^H(S)$;
- output** π .

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

$$\begin{aligned}
& \Pr[\text{Read}(\pi) \subseteq \text{Lottery}(S_f) \wedge \text{Verify}^H(\pi) = 1] \geq \\
& \Pr \left[\text{Read}(\pi) \subseteq \text{Lottery}(S_f) \wedge \text{Verify}^H(\pi) = 1 \mid |\text{Lottery}(S_f)| \geq \frac{\mu n_f}{n_p} \right] \times \\
& \Pr \left[|\text{Lottery}(S_f)| \geq \frac{\mu n_f}{n_p} \right] [>]
\end{aligned}$$

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,

$$\begin{aligned} & [\geq] \frac{1}{4} \cdot \Pr \left[\text{Read}(\pi) \subseteq \text{Lottery}(S_f) \wedge \text{Verify}^H(\pi) = 1 \mid |\text{Lottery}(S_f)| \geq \frac{\mu n_f}{n_p} \right] \geq \\ & \quad \frac{1}{4} \cdot \Pr \left[\text{Read}(\pi) \subseteq \text{Lottery}(S_f) \wedge \text{Verify}^H(\pi) = 1 \mid \right. \\ & \quad \left. \text{Read}(\pi) \subseteq S \wedge \text{Verify}^H(\pi) = 1 \wedge |\text{Lottery}(S_f)| \geq \frac{\mu n_f}{n_p} \right] \times \\ & \quad \times \Pr \left[\text{Read}(\pi) \subseteq S \wedge \text{Verify}^H(\pi) = 1 \mid |\text{Lottery}(S_f)| \geq \frac{\mu n_f}{n_p} \right] [\geq] \end{aligned}$$

One can see that in \mathcal{A}_{S_f} , independent of $|\text{Lottery}(S_f)|$, S is a uniformly random subset of S_p of size m , and using equation 11,

$$\begin{aligned} & [\geq] \frac{1}{8} \cdot \Pr \left[\text{Read}(\pi) \subseteq \text{Lottery}(S_f) \wedge \text{Verify}^H(\pi) = 1 \mid \right. \\ & \quad \left. \text{Read}(\pi) \subseteq S \wedge \text{Verify}^H(\pi) = 1 \wedge |\text{Lottery}(S_f)| \geq \frac{\mu n_f}{n_p} \right] = \\ & \quad \frac{1}{8} \cdot \Pr \left[\text{Read}(\pi) \subseteq \text{Lottery}(S_f) \mid \right. \\ & \quad \left. \text{Read}(\pi) \subseteq S \wedge \text{Verify}^H(\pi) = 1 \wedge |\text{Lottery}(S_f)| \geq \frac{\mu n_f}{n_p} \right] [\geq] \end{aligned}$$

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

$$\begin{aligned} & [\geq] \frac{1}{8} \cdot \prod_{i=0}^{u-1} \frac{\binom{\frac{\mu n_f}{n_p}}{\rho} - i}{\rho - i} \geq \\ & \quad \frac{1}{8} \cdot \left(\frac{\frac{\mu n_f}{n_p} - u}{\rho} \right)^u = \\ & \quad \frac{1}{8} \cdot \left(\frac{\frac{\mu n_f}{n_p}}{\rho} \right)^u \cdot \left(\frac{\frac{\mu n_f}{n_p} - u}{\frac{\mu n_f}{n_p}} \right)^u = \\ & \quad \frac{1}{8} \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \left(1 - \frac{u n_p}{\mu n_f} \right)^u [\geq] \end{aligned}$$

Since $\frac{u n_p}{\mu n_f} \leq \frac{u n_p}{\frac{3u^2 n_p \log e}{2 n_f} \cdot n_f} = \frac{2}{3u \log e} \leq \frac{2}{3 \log e} \leq \frac{1}{2}$ and $1 - x \geq e^{-x-x^2} \geq e^{-\frac{3}{2}x}$ for

$$0 \leq x \leq \frac{1}{2},$$

$$\begin{aligned} & [\geq] \frac{1}{8} \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \left(e^{-\frac{3u n_p}{2\mu n_f}} \right)^u \geq \\ & \frac{1}{8} \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(-\frac{3u^2 n_p}{2 \cdot \frac{3u^2 n_p \log e}{2n_f} \cdot n_f} \right) = \\ & \frac{1}{8} \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u \cdot \exp \left(-\frac{1}{\log e} \right) = \\ & \frac{1}{16} \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u. \end{aligned}$$

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

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

On the other hand, by soundness from Definition 5,

$$\begin{aligned} \Pr[\text{Read}(\pi') \subseteq \text{Lottery}(S'_f) \wedge \text{Verify}^H(\pi') = 1] &\leq \\ \Pr[\text{SoundExp}(S'_f) = 1] &\leq 2^{-\lambda_{\text{sec}}}. \end{aligned}$$

Therefore,

$$\begin{aligned} \frac{1}{16} \cdot \left(\frac{\mu n_f}{\rho n_p} \right)^u &< 2^{-\lambda_{\text{sec}}} \iff \\ \left(\frac{\rho n_p}{\mu n_f} \right)^u &> 2^{\lambda_{\text{sec}}-4} \iff \\ u \left(\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu} \right) &> \lambda_{\text{sec}} - 4 \iff \\ u &> \frac{\lambda_{\text{sec}} - 4}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}}. \end{aligned}$$

□

Proof of Corollary 6. Suppose otherwise, then $\Pr \left[\left| \text{Read}(\text{Prove}^H(\text{Lottery}(S_p))) \right| \leq u \right] = 1$.

Let $\delta = \sqrt{\frac{\lambda_{\text{rel}}}{4\mu}}$ and $\rho = \lfloor (1 - \delta)\mu \rfloor$. By Lemma 53,

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

In order to use Theorem 18, we need to show that $\frac{\rho n_p}{\mu n_f} > 1$. First we show that $(1 - \delta)\mu - 1 \geq (1 - 2\delta)\mu$. This is equivalent to

$$\delta\mu \geq 1 \iff \sqrt{\frac{\lambda_{\text{rel}}}{4\mu}} \cdot \mu \geq 1 \iff \sqrt{\frac{\lambda_{\text{rel}} \cdot \mu}{4}} \geq 1 \iff \mu \geq \frac{4}{\lambda_{\text{rel}}}$$

which is true by our assumption. Then

$$\frac{\rho n_p}{\mu n_f} > \frac{((1 - \delta)\mu - 1)n_p}{\mu n_f} \geq \frac{(1 - 2\delta)\mu n_p}{\mu n_f} = \frac{(1 - 2\delta)n_p}{n_f}.$$

This is at least 1 if and only if

$$\begin{aligned} 1 - 2\delta &\geq \frac{n_f}{n_p} \iff \\ 2\delta &\leq 1 - \frac{n_f}{n_p} \iff \\ 4\delta^2 &\leq \left(1 - \frac{n_f}{n_p}\right)^2 \iff \\ \frac{\lambda_{\text{rel}}}{\mu} &\leq \left(1 - \frac{n_f}{n_p}\right)^2 \iff \\ \mu &\geq \frac{\lambda_{\text{rel}}}{\left(1 - \frac{n_f}{n_p}\right)^2} \end{aligned}$$

which is true by our assumption.

By Theorem 18, $u > \frac{\lambda_{\text{sec}} - 4}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}}$. We need to prove that $\frac{\lambda_{\text{sec}} - 4}{\log \frac{n_p}{n_f} + \log \frac{\rho}{\mu}} \geq \alpha$. Define

$\lambda' = \lambda_{\text{sec}} - 4$. This is equivalent to

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

which is true by our assumption.

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

Proof of Corollary 7. Suppose otherwise, then $\Pr \left[\left| \text{Read}(\text{Prove}^H(\text{Lottery}(S_p))) \right| \leq u \right] = 1$.

Let $\delta = \sqrt{\frac{\lambda_{\text{rel}}}{4\mu}}$ and $\rho = \lfloor (1 - \delta)\mu \rfloor$. By Lemma 53,

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

In order to use Theorem 18, we need to show that $\frac{\rho n_p}{\mu n_f} > 1$. The proof of Corollary 6 shows how the assumption $\mu \geq \max \left\{ \frac{4}{\lambda_{\text{rel}}}, \frac{\lambda_{\text{rel}}}{(1 - \frac{n_f}{n_p})^2} \right\}$ implies it.

By Theorem 18,

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

which is a contradiction. \square

Proof of Theorem 19. Referencing the internals of the extractor algorithm, 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

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

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

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

where the last step follows from Lemma 12. \square

Proof of Lemma 12. Theorem 1 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 1, we simply need to switch from thinking about set elements as input to H to thinking about indices as inputs. We will define a new function Q to do so.

Let X_1, \dots, X_N be the first n_f distinct weight-1 elements that are present in random oracle queries of \mathcal{A} . If $N < n_f$, pad the sequence X_1, \dots, 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 $Q(1, t, \dots)$, and $Q(2, t, \dots)$ to be the same as H_1 and H_2 , respectively, but operating on indices rather than values of the X s; that is, $Q(i, t, v_1, \dots, v_j) = H_i(t, X_{v_1}, \dots, X_{v_j})$. 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 k th part, and observe that Q_k is independent of Q_1, \dots, 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, \dots, 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 $1 \leq t \leq d$ and $v_1, \dots, v_u \in [n_f]$ such that for all $1 \leq i \leq u$, $Q(1, t, v_1, \dots, v_i) = 1$, and $Q(2, t, v_1, \dots, v_u) = 1$. $\Pr[E] \leq \Pr[\text{cert}]$, because cert happens whenever E happens (and may also happen using some of the dummy values $X_{N+1} \dots, X_{n_f}$). And $\Pr[\text{cert}] \leq 2^{-\lambda_{\text{sec}}}$ by the same exact argument as in Theorem 1. \square

E Additional Lemmas

E.1 Useful facts

Lemma 25. $e^x \geq 1 + x$ for all x .

Lemma 26. $e^{-x} \leq 1 - x + \frac{x^2}{2}$ for all $x \geq 0$.

Lemma 27. $e^{-x} \leq 1 - \frac{x}{2}$ for all $0 \leq x \leq 1$.

Proof. Follows from Lemma 26. \square

Lemma 28. $\ln(1 - x) \geq -x - x^2$ for all $0 \leq x \leq \frac{1}{2}$.

Lemma 29. $\ln(1 - x) \geq -\frac{3}{2}x$ for all $0 \leq x \leq \frac{1}{2}$.

Proof. Follows from Lemma 28. \square

Lemma 30. $\ln(1 - x) \leq -x - \frac{x^2}{2}$ for all $x \geq 0$.

Lemma 31. $\frac{1}{1 - e^{-x}} \leq \frac{2}{x}$ for all $0 < x \leq 1$.

Proof. From Lemma 27, $\frac{1}{1 - e^{-x}} \leq \frac{1}{1 - (1 - \frac{x}{2})} = \frac{2}{x}$. \square

Lemma 32. Suppose $0 \leq \epsilon \leq \frac{1}{2}$. Then $\frac{1}{1 - \epsilon} \leq 1 + 2\epsilon$.

Lemma 33.

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

Proof. It is known that for any $x \in \mathbb{R}$, $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$. 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:

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

□

Lemma 34. Let A, E be probabilistic events such that $\Pr[A|E] \leq x$ and $\Pr[\bar{E}] \leq y \leq 1$. Then $\Pr[A] \leq x + y - xy$.

Proof.

$$\begin{aligned} \Pr[A] &= \\ \Pr[A|E] \cdot \Pr[E] + \Pr[A|\bar{E}] \cdot \Pr[\bar{E}] &\leq \\ \Pr[A|E] \cdot (1 - \Pr[\bar{E}]) + \Pr[\bar{E}] &= \\ \Pr[A|E] + \Pr[\bar{E}] \cdot (1 - \Pr[A|E]) &\leq \\ \Pr[A|E] + y \cdot (1 - \Pr[A|E]) &= \\ y + \Pr[A|E] \cdot (1 - y) &\leq \\ y + x \cdot (1 - y) &= \\ x + y - xy & \end{aligned}$$

□

Lemma 35. Let $n \in \mathbb{N}$, let $\{X_i\}_{i \in [n]}$ be random variables with $X_i \in \mathbb{R}$, $\mathbb{E}[X_i] = \mu \geq 0$ and let $\{Y_i\}_{i \in \mathbb{N}}$ be random variables with $Y_i \in \{0, 1\}$, $\Pr[Y_i = 1] = p$ such that all (X_i, Y_i) are independent and identically distributed. Also define

$$N = \begin{cases} \min\{i : Y_i = 1\} & \exists i \in [n], Y_i = 1 \\ n & \text{otherwise.} \end{cases}$$

Then $\mathbb{E}\left[\sum_{i=1}^N X_i\right] \leq \mu/p$.

Proof. Let

$$\begin{aligned}
t &= \mathbb{E} \left[\sum_{j=1}^N X_j \right] = \\
&\mathbb{E} \left[\sum_{j=1}^N X_j \middle| N = 1 \right] \cdot \Pr[N = 1] + \mathbb{E} \left[\sum_{j=1}^N X_j \middle| N \neq 1 \right] \cdot \Pr[N \neq 1] = \\
&\mathbb{E}[X_1 | N = 1] \cdot \Pr[N = 1] + \mathbb{E} \left[X_1 + \sum_{j=2}^N X_j \middle| N \neq 1 \right] \cdot \Pr[N \neq 1] = \\
&\mathbb{E}[X_1 | N = 1] \cdot \Pr[N = 1] + \mathbb{E}[X_1 | N \neq 1] \cdot \Pr[N \neq 1] + \\
&\mathbb{E} \left[\sum_{j=2}^N X_j \middle| N \geq 2 \right] \cdot \Pr[N \neq 1] = \\
&\mu + \mathbb{E} \left[\sum_{j=2}^N X_j \middle| N \geq 2 \right] \cdot (1 - p) \leq \\
&\mu + t \cdot (1 - p).
\end{aligned}$$

Hence, $t \leq \frac{\mu}{p}$. □

Lemma 36 (Stirling's approximation). *For all $n \in \mathbb{N}$,*

$$\sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{\frac{1}{12n+1}} < n! < \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}.$$

E.2 Chernoff Bounds

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

Lemma 37 (Upper tail). *For any $\delta \geq 0$,*

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

Lemma 38 (Upper tail, simpler). *For any $\delta \in [0, 1]$,*

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

Lemma 39 (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 40 (Lower tail, simpler). *For any $\delta \geq 0$,*

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

E.3 Lemmas for Section 3.1.1

Lemma 41. 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 $1 \leq j \leq d$, $1 \leq i \leq u$, $s_1, \dots, s_i \in S_p$ and $1 \leq k \leq i$, let the indicator random variable

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

Then

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

Also for $1 \leq j \leq d$, $0 \leq i \leq u$ and $s_1, \dots, s_i \in S_p$, let

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

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

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

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

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

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

Inductive step:

$$\begin{aligned}
& \mathbb{E} \left[\exp \left(t \cdot F(j, s_1, \dots, s_{u-k-1}) \right) \right] = \\
& \mathbb{E} \left[\exp \left(t \cdot \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \right] = \\
& \mathbb{E} \left[\exp \left(t \cdot \sum_{s_{u-k} \in S_p} \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \right] = \\
& \mathbb{E} \left[\prod_{s_{u-k} \in S_p} \exp \left(t \cdot \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \right].
\end{aligned}$$

Define the random variables

$$X_{s_{u-k}} = \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k}.$$

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

$$\begin{aligned}
& \mathbb{E} \left[\exp \left(t \cdot F(j, s_1, \dots, s_{u-k-1}) \right) \right] = \\
& \mathbb{E} \left[\prod_{s_{u-k} \in S_p} \exp \left(t \cdot X_{s_{u-k}} \right) \right] = \\
& \prod_{s_{u-k} \in S_p} \mathbb{E} \left[\exp \left(t \cdot X_{s_{u-k}} \right) \right].
\end{aligned} \tag{13}$$

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

$$\begin{aligned}
& \mathbb{E} \left[\exp \left(t \cdot X_{s_{u-k}} \right) \right] = \\
& \mathbb{E} \left[\exp \left(t \cdot X_{s_{u-k}} \right) \middle| E_{s_{u-k}} \right] \cdot \Pr \left[E_{s_{u-k}} \right] + \\
& \mathbb{E} \left[\exp \left(t \cdot X_{s_{u-k}} \right) \middle| \neg E_{s_{u-k}} \right] \cdot \Pr \left[\neg E_{s_{u-k}} \right] = \\
& \mathbb{E} \left[\exp \left(t \cdot X_{s_{u-k}} \right) \middle| E_{s_{u-k}} \right] \cdot \frac{1}{n_p} + \mathbb{E} \left[\exp \left(t \cdot 0 \right) \middle| E_{s_{u-k}} \right] \cdot \left(1 - \frac{1}{n_p} \right) = \\
& \frac{1}{n_p} \cdot \mathbb{E} \left[\exp \left(t \cdot X_{s_{u-k}} \right) \middle| E_{s_{u-k}} \right] + 1 - \frac{1}{n_p}.
\end{aligned} \tag{14}$$

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

$$\begin{aligned}
X_{s_{u-k}} &= \\
&\sum_{\substack{u-k \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} = \\
&\sum_{\substack{u-k+1 \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} + I_{j, s_1, \dots, s_{u-k}, u-k} = \\
&\sum_{\substack{u-k+1 \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k+1} + 1 = \\
&F(j, s_1, \dots, s_{u-k}) + 1
\end{aligned}$$

Using equation 14,

$$\begin{aligned}
&\mathbb{E} \left[\exp(t \cdot X_{s_{u-k}}) \right] = \\
&\frac{1}{n_p} \cdot \mathbb{E} \left[\exp \left(t \cdot (F(j, s_1, \dots, s_{u-k}) + 1) \right) \middle| E_{s_{u-k}} \right] + 1 - \frac{1}{n_p} = \\
&\frac{1}{n_p} \cdot \mathbb{E} \left[\exp(t \cdot F(j, s_1, \dots, s_{u-k})) \middle| E_{s_{u-k}} \right] \cdot e^t + 1 - \frac{1}{n_p} = \\
&\frac{1}{n_p} \cdot \mathbb{E} \left[\exp(t \cdot F(j, s_1, \dots, s_{u-k})) \right] \cdot e^t + 1 - \frac{1}{n_p} = \\
&\frac{1}{n_p} x_k e^t + 1 - \frac{1}{n_p}.
\end{aligned}$$

Combining this with equation 13 we get

$$\begin{aligned}
&\mathbb{E} \left[\exp(t \cdot F(j, s_1, \dots, s_{u-k-1})) \right] = \\
&\prod_{s_{u-k} \in S_p} \mathbb{E} \left[\exp(t \cdot X_{s_{u-k}}) \right] = \\
&\prod_{s_{u-k} \in S_p} \left(\frac{1}{n_p} x_k e^t + 1 - \frac{1}{n_p} \right) = \\
&\left(x_k e^t + 1 - \frac{1}{n_p} \right)^{n_p} = \\
&x_{k+1}
\end{aligned}$$

which concludes the inductive step.

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

E.4 Lemmas for Section 3.2

Lemma 42. *Assume $c \geq 2$, $\lambda > 0$, $d \geq 4cu\lambda$ and $q = \frac{2\lambda}{d}$. Then completeness error is $\leq \frac{2}{c} + e^{-\lambda} - \frac{2}{c} \cdot e^{-\lambda}$.*

Proof. 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 $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+cq^2}$ and let F be the event that the honest prover fails. By Lemma 43, $\Pr[F|E] \leq e^{-\lambda}$. Also by Lemma 4,

$$\Pr[\bar{E}] = \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} > e^{-q+cq^2}\right] \leq \Pr\left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq 1 - q + cq^2\right] \leq \frac{2}{c}.$$

Hence by Lemma 34, $\Pr[F] \leq \frac{2}{c} + e^{-\lambda} - \frac{2}{c} \cdot e^{-\lambda}$. \square

Lemma 43. *Let $\lambda > 0$, $c > 0$, $d \geq 4cu\lambda$, $q = \frac{2\lambda}{d}$, 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 $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+cq^2}$ and let F be the event that the honest prover fails. Then $\Pr[F|E] \leq e^{-\lambda}$.*

Proof. By Lemma 6, $\Pr[F|E] \leq e^{-(q-cuq^2)d}$. This is at most $e^{-\lambda}$ if and only if

$$(q - cuq^2)d \geq \lambda \iff$$

$$\left(\frac{2\lambda}{d} - cu\left(\frac{2\lambda}{d}\right)^2\right)d \geq \lambda \iff$$

$$2 - cu \cdot \frac{4\lambda}{d} \geq 1 \iff$$

$$1 \geq \frac{4cu\lambda}{d} \iff$$

$$d \geq 4cu\lambda$$

which is true by our assumption about d . \square

Lemma 44. *Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i , let $c > 0$, let E be the event that $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+cq^2}$ and let $F(t)$ be the event that there does not exist a valid proof starting with integer t . Then $\Pr[F(t)|E] \leq e^{-q+cuq^2}$.*

Proof. Define random function $f(x) = \frac{1}{n_p} \sum_{i=1}^{n_p} x^{X_i}$. By Lemma 45,

$$\Pr[F(t)|H_0] \leq e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}}\right)^u.$$

Therefore,

$$\begin{aligned} \Pr[F(t)|E] &= \mathbb{E}\left[\Pr[F(t)|H_0, E] \mid E\right] = \mathbb{E}\left[\Pr[F(t)|H_0] \mid E\right] = \\ &= \mathbb{E}\left[e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}}\right)^u \mid f(e^{-q}) \leq e^{-q+cuq^2}\right] \leq \\ &= e^{-q+cuq^2}. \end{aligned}$$

\square

Lemma 45. 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 $F(t, s_1, \dots, s_k)$ be the event that there is no suffix of honest player's signatures that works, meaning there is no s_{k+1}, \dots, s_u such that

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

Then for all t , $0 \leq k \leq u$ and s_1, \dots, s_k ,

$$\Pr[F(t, s_1, \dots, s_k) | H_0] \leq e^{-q} \cdot \left(\frac{f(e^{-q})}{e^{-q}} \right)^{u-k}.$$

Proof. By Lemma 46, $\Pr[F(t, s_1, \dots, s_k) | H_0] = f^{(u-k)}(1 - q)$. Since $f(x)$ is an increasing function, this is at most $f^{(u-k)}(e^{-q})$, and by Lemma 47, it is at most

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

□

Lemma 46. 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 $F(t, s_1, \dots, s_k)$ be the event that there is no suffix of honest player's signatures that works, meaning there is no s_{k+1}, \dots, s_u such that

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

Then for all t , $0 \leq k \leq u$ and s_1, \dots, s_k , $\Pr[F(t, s_1, \dots, s_k) | H_0] = f^{(u-k)}(1 - q)$.

Proof. Notice the following:

- $F(t, s_1, \dots, s_u)$ is true iff $H_2(t, s_1, \dots, s_u) = 0$;
- for all $0 \leq k < u$: $F(t, s_1, \dots, s_k) = \bigwedge_{s_{k+1} \in S_p} ((H_1(t, s_1, \dots, s_k) \neq H_0(s_{k+1})) \vee F(t, s_1, \dots, s_{k+1}))$.

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

$0|H_0] = 1 - q = f^{(0)}(1 - q)$. Inductive step:

$$\begin{aligned}
& \Pr[F(t, s_1, \dots, s_{u-i-1})|H_0] = \\
& \sum_{j=1}^{n_p} \Pr[F(t, s_1, \dots, s_{u-i-1})|H_0, H_1(t, s_1, \dots, s_{u-i-1}) = j] \times \\
& \Pr[H_1(t, s_1, \dots, s_{u-i-1}) = j|H_0] = \\
& \frac{1}{n_p} \sum_{j=1}^{n_p} \Pr[F(t, s_1, \dots, s_{u-i-1})|H_0, H_1(t, s_1, \dots, s_{u-i-1}) = j] = \\
& \frac{1}{n_p} \sum_{j=1}^{n_p} \Pr\left[\bigwedge_{s_{u-i} \in S_p, H_0(s_{u-i})=j} F(t, s_1, \dots, s_{u-i})|H_0, H_1(t, s_1, \dots, s_{u-i-1}) = j\right] [=]
\end{aligned}$$

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

$$[=] \frac{1}{n_p} \sum_{j=1}^{n_p} \Pr\left[\bigwedge_{s_{u-i} \in S_p, H_0(s_{u-i})=j} F(t, s_1, \dots, s_{u-i})|H_0\right] [=]$$

When H_0 is fixed, events $\{F(t, s_1, \dots, s_{u-i}) : s_{u-i} \in S_p, H_0(s_{u-i}) = j\}$ 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,

$$\begin{aligned}
[=] & \frac{1}{n_p} \sum_{j=1}^{n_p} \prod_{s_{u-i} \in S_p, H_0(s_{u-i})=j} \Pr[F(t, s_1, \dots, s_{u-i})|H_0] = \\
& \frac{1}{n_p} \sum_{j=1}^{n_p} \prod_{s_{u-i} \in S_p, H_0(s_{u-i})=j} f^{(i)}(1 - q) = \\
& \frac{1}{n_p} \sum_{j=1}^{n_p} (f^{(i)}(1 - q))^{X_j} = \\
& f^{(i+1)}(1 - q).
\end{aligned}$$

□

Lemma 47. $n \in \mathbb{N}$, $k \in \mathbb{Z}$, $k \geq 0$, 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 $0 < z < 1$, $f^{(k)}(z) \leq z \cdot \left(\frac{f(z)}{z}\right)^k$.

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 $z, f(z), f^{(2)}(z), \dots$ is non-

decreasing and is < 1 . Also, the function $g(x) = \frac{f(x)}{x}$ is non-increasing since

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

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

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

□

E.5 Lemmas for Section 3.2.1

Lemma 48. *Let $u, n_p \in \mathbb{N}$, $\lambda, \alpha, c > 0$,*

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

$X_i = |\{s \in S_p : H_0(s) = i\}|$ 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

$$t = \frac{\alpha}{e^{cu\alpha} \cdot u} \tag{15}$$

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

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

By Lemma 49, $\mathbb{E} [e^{tZ} | H_0] = G_u^d$.

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:

$$\begin{aligned} G_{k+1} &= \frac{1}{n_p} \sum_{i=1}^{n_p} (G_k \cdot e^t)^{X_i} \leq \frac{1}{n_p} \sum_{i=1}^{n_p} e^{(y_k+t)X_i} = \\ &= \frac{1}{n_p} \sum_{i=1}^{n_p} \exp((kte^{ck\alpha} + t)X_i) \leq \\ &= \frac{1}{n_p} \sum_{i=1}^{n_p} \exp((k+1)te^{ck\alpha}X_i) [\leq]. \end{aligned}$$

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

$$\begin{aligned} [\leq] &\left(\frac{1}{n_p} \sum_{i=1}^{n_p} e^{\alpha X_i} \right)^{\frac{(k+1)te^{ck\alpha}}{\alpha}} \leq \\ &= \left(e^{\alpha + c\alpha^2} \right)^{\frac{(k+1)te^{ck\alpha}}{\alpha}} = \\ &= \exp\left(\alpha(1+c\alpha) \frac{(k+1)te^{ck\alpha}}{\alpha} \right) = \\ &= \exp\left((k+1)te^{ck\alpha}(1+c\alpha) \right) \leq \\ &= \exp\left((k+1)te^{ck\alpha}e^{c\alpha} \right) = \\ &= \exp\left((k+1)te^{c(k+1)\alpha} \right) = \\ &= e^{y_{k+1}}. \end{aligned}$$

Hence,

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

By Markov's inequality,

$$\begin{aligned} \Pr[Z \geq \delta du | E] &= \Pr [e^{tZ} \geq e^{\delta t du} | E] \leq \\ &= \frac{\mathbb{E} [e^{tZ} | E]}{e^{\delta t du}} \leq \frac{e^{d\alpha}}{e^{\delta t du}} = \exp(-d(\delta t u - \alpha)). \end{aligned}$$

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

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

Substituting the value of t from equation 15,

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

which is true by the statement of the lemma. \square

Lemma 49. 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

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

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

Proof. For $1 \leq j \leq d$, $1 \leq i \leq u$, $s_1, \dots, s_i \in S_p$ and $1 \leq k \leq i$, let the indicator random variable

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

Then

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

Also for $1 \leq j \leq d$, $0 \leq i \leq u$ and $s_1, \dots, s_i \in S_p$, let

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

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

$$\begin{aligned}
& \mathbb{E}[e^{tZ} | H_0] = \\
& \mathbb{E} \left[\exp \left(t \cdot \sum_{j=1}^d F(j) \right) \middle| H_0 \right] = \\
& \mathbb{E} \left[\prod_{j=1}^d e^{tF(j)} \middle| H_0 \right] = \\
& \prod_{j=1}^d \mathbb{E} \left[e^{tF(j)} \middle| H_0 \right].
\end{aligned} \tag{16}$$

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

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

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

Inductive step:

$$\begin{aligned}
& \mathbb{E} \left[\exp \left(t \cdot F(j, s_1, \dots, s_{u-k-1}) \right) \middle| H_0 \right] = \\
& \mathbb{E} \left[\exp \left(t \cdot \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \middle| H_0 \right] = \\
& \sum_{b=1}^{n_p} \mathbb{E} \left[\exp \left(t \cdot \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \middle| H_1(j, s_1, \dots, s_{u-k-1}) = b, H_0 \right] \times \\
& \Pr \left[H_1(j, s_1, \dots, s_{u-k-1}) = b \middle| H_0 \right] = \\
& \frac{1}{n_p} \sum_{b=1}^{n_p} \mathbb{E} \left[\exp \left(t \cdot \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k}, \dots, s_r \in S_p, \\ H_0(s_{u-k}) = b}} I_{j, s_1, \dots, s_r, u-k} \right) \middle| H_1(j, s_1, \dots, s_{u-k-1}) = b, H_0 \right] = \\
& \frac{1}{n_p} \sum_{b=1}^{n_p} \mathbb{E} \left[\exp \left(t \cdot \sum_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k}) = b}} \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \middle| H_1(j, s_1, \dots, s_{u-k-1}) = b, H_0 \right] = \\
& \frac{1}{n_p} \sum_{b=1}^{n_p} \mathbb{E} \left[\prod_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k}) = b}} \exp \left(t \cdot \sum_{\substack{u-k \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \middle| H_1(j, s_1, \dots, s_{u-k-1}) = b, H_0 \right] = \\
& \frac{1}{n_p} \sum_{b=1}^{n_p} \mathbb{E} \left[\prod_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k}) = b}} \exp \left(t \cdot \left(I_{j, s_1, \dots, s_{u-k}, u-k} + \sum_{\substack{u-k+1 \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k} \right) \right) \middle| \right. \\
& \left. H_1(j, s_1, \dots, s_{u-k-1}) = b, H_0 \right] = \\
& \frac{1}{n_p} \sum_{b=1}^{n_p} \mathbb{E} \left[\prod_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k}) = b}} \exp \left(t \cdot \left(1 + \sum_{\substack{u-k+1 \leq r \leq u, \\ s_{u-k+1}, \dots, s_r \in S_p}} I_{j, s_1, \dots, s_r, u-k+1} \right) \right) \middle| \right. \\
& \left. H_1(j, s_1, \dots, s_{u-k-1}) = b, H_0 \right] = \\
& \frac{1}{n_p} \sum_{b=1}^{n_p} \mathbb{E} \left[\prod_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k}) = b}} \exp \left(t \cdot (1 + F(j, s_1, \dots, s_{u-k})) \right) \middle| H_0 \right] [=]
\end{aligned}$$

Since with fixed H_0 , $F(j, s_1, \dots, s_{u-k})$ for $s_{u-k} \in S_p$ are all independent,

$$\begin{aligned}
[&=] \frac{1}{n_p} \sum_{b=1}^{n_p} \prod_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k})=b}} \mathbb{E} \left[\exp \left(t \cdot (1 + F(j, s_1, \dots, s_{u-k})) \right) \middle| H_0 \right] = \\
&\frac{1}{n_p} \sum_{b=1}^{n_p} \prod_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k})=b}} \left(e^t \cdot \mathbb{E} \left[\exp \left(t \cdot F(j, s_1, \dots, s_{u-k}) \right) \middle| H_0 \right] \right) = \\
&\frac{1}{n_p} \sum_{b=1}^{n_p} \prod_{\substack{s_{u-k} \in S_p, \\ H_0(s_{u-k})=b}} \left(e^t \cdot G_k \right) = \\
&\frac{1}{n_p} \sum_{b=1}^{n_p} \left(G_k \cdot e^t \right)^{X_b} = \\
&G_{k+1}
\end{aligned}$$

which concludes the inductive step.

Therefore by equation 16, $\mathbb{E}[e^{tZ} | H_0] = \prod_{j=1}^d \mathbb{E} \left[e^{tF(j)} \middle| H_0 \right] = \prod_{j=1}^d G_u = G_u^d$. \square

Lemma 50. *Let $\lambda > 0$, $w, n \in \mathbb{N}$, $0 < \alpha \leq \frac{1}{w}$, and*

$$c = \begin{cases} 2 \left(\frac{1}{\alpha} + w \right) \sqrt{\frac{2\lambda}{n}} + 2 & \text{if } n \geq \frac{w^2\lambda}{2} \\ \left(\frac{1}{\alpha} + w \right) \cdot \frac{w\lambda}{n} + 2 \left(1 + \frac{1}{\alpha w} \right) + 2 & \text{otherwise} \end{cases}$$

(note that $2 \left(\frac{1}{\alpha} + w \right) \sqrt{\frac{2\lambda}{n}} + 2 \leq \left(\frac{1}{\alpha} + w \right) \cdot \frac{w\lambda}{n} + 2 \left(1 + \frac{1}{\alpha w} \right) + 2$ for all n). Also let Y_i be Poisson random variables with expectation 1 and let

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

Then

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

Proof. Let $0 < r \leq \frac{1}{\alpha w \cdot (1 + \alpha w)}$. We calculate the following:

$$\begin{aligned} \mathbb{E}[e^{rA_i}] &= \\ &= \sum_{j=0}^{\infty} \mathbb{E}[e^{rA_i} | Y_i = j] \cdot \Pr[Y_i = j] = \\ &= \sum_{j=0}^w \exp(re^{\alpha j}) \cdot \frac{1}{ej!} + \sum_{j=w+1}^{\infty} \frac{1}{ej!} \leq \\ &= e^{r-1} \sum_{j=0}^w \frac{\exp\left(r(e^{\alpha j} - 1)\right)}{j!} + \sum_{j=w+1}^{\infty} \frac{1}{ej!} [\leq] \end{aligned}$$

The next two steps use the fact that when $x \leq 1$, $e^x \leq 1 + x + x^2$. $r(e^{\alpha j} - 1) \leq r(e^{\alpha w} - 1) \leq r(1 + \alpha w + (\alpha w)^2 - 1) = r \cdot \alpha w \cdot (1 + \alpha w) \leq 1$ by the assumption about r . Thus,

$$\begin{aligned} & [\leq] e^{r-1} \sum_{j=0}^w \frac{1 + r(e^{\alpha j} - 1) + r^2(e^{\alpha j} - 1)^2}{j!} + \sum_{j=w+1}^{\infty} \frac{1}{ej!} \leq \\ & e^{r-1} \sum_{j=0}^w \frac{1 + r(\alpha j + \alpha^2 j^2) + r^2(\alpha j + \alpha^2 j^2)^2}{j!} + \sum_{j=w+1}^{\infty} \frac{1}{ej!} \leq \\ & e^{r-1} \sum_{j=0}^w \frac{1 + r(\alpha j + \alpha^2 j^2) + r^2((1 + \alpha w)\alpha j)^2}{j!} + \sum_{j=w+1}^{\infty} \frac{1}{ej!} = \\ & e^{r-1} \sum_{j=0}^w \frac{1 + r\alpha j + (r + (1 + \alpha w)^2 r^2)\alpha^2 j^2}{j!} + \sum_{j=w+1}^{\infty} \frac{1}{ej!} \leq \\ & e^{r-1} \sum_{j=0}^{\infty} \frac{1 + r\alpha j + (r + (1 + \alpha w)^2 r^2)\alpha^2 j^2}{j!} = \\ & e^{r-1} \left(\sum_{j=0}^{\infty} \frac{1}{j!} + r\alpha \sum_{j=0}^{\infty} \frac{j}{j!} + (r + (1 + \alpha w)^2 r^2)\alpha^2 \sum_{j=0}^{\infty} \frac{j^2}{j!} \right) [=] \end{aligned}$$

By Lemma 33,

$$\begin{aligned}
[=]e^{r-1} &\left(e + r\alpha \cdot e + (r + (1 + \alpha w)^2 r^2) \alpha^2 \cdot 2e \right) = \\
&e^r \left(1 + r\alpha + 2(r + (1 + \alpha w)^2 r^2) \alpha^2 \right) \leq \\
&\exp(r) \cdot \exp \left(r\alpha + 2(r + (1 + \alpha w)^2 r^2) \alpha^2 \right) = \\
&\exp \left(r + r\alpha + 2(r + (1 + \alpha w)^2 r^2) \alpha^2 \right) = \\
&\exp \left(r \left(1 + \alpha + 2(1 + (1 + \alpha w)^2 r) \alpha^2 \right) \right).
\end{aligned}$$

We are now ready to bound $\frac{1}{n} \sum_{i=1}^n A_i$. Assume $s > 0$ and $\frac{s}{n} \leq \frac{1}{\alpha w \cdot (1 + \alpha w)}$, and define $c_1 = 2(1 + (1 + \alpha w)^2 \cdot \frac{s}{n})$. By Markov's inequality,

$$\begin{aligned}
&\Pr \left[\frac{1}{n} \sum_{i=1}^n A_i \geq 1 + \alpha + c\alpha^2 \right] = \\
&\Pr \left[\exp \left(\frac{s}{n} \sum_{i=1}^n A_i \right) \geq \exp \left(s(1 + \alpha + c\alpha^2) \right) \right] \leq \\
&\frac{\mathbb{E} \left[\exp \left(\frac{s}{n} \sum_{i=1}^n A_i \right) \right]}{\exp \left(s(1 + \alpha + c\alpha^2) \right)} = \frac{\mathbb{E} \left[\prod_{i=1}^n \exp \left(\frac{s}{n} A_i \right) \right]}{\exp \left(s(1 + \alpha + c\alpha^2) \right)} = \frac{\prod_{i=1}^n \mathbb{E} \left[\exp \left(\frac{s}{n} A_i \right) \right]}{\exp \left(s(1 + \alpha + c\alpha^2) \right)} = \\
&\frac{\prod_{i=1}^n \exp \left(\frac{s}{n} \cdot (1 + \alpha + c_1 \alpha^2) \right)}{\exp \left(s(1 + \alpha + c\alpha^2) \right)} = \frac{\exp \left(s(1 + \alpha + c_1 \alpha^2) \right)}{\exp \left(s(1 + \alpha + c\alpha^2) \right)} = \\
&\exp \left(- (c - c_1) s \alpha^2 \right).
\end{aligned}$$

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

$$(c - c_1) s \alpha^2 \geq \lambda \iff c \geq \frac{\lambda}{s \alpha^2} + c_1; \iff c \geq \frac{\lambda}{s \alpha^2} + 2(1 + \alpha w)^2 \cdot \frac{s}{n} + 2;$$

thus we set $c = \frac{\lambda}{s \alpha^2} + 2(1 + \alpha w)^2 \cdot \frac{s}{n} + 2$. Differentiating with respect to s we find that the minimum is achieved when $s = \frac{1}{(1 + \alpha w) \alpha} \cdot \sqrt{\frac{\lambda n}{2}}$. Then the requirement that $\frac{s}{n} \leq \frac{1}{\alpha w \cdot (1 + \alpha w)}$ is satisfied if and only if $n \geq \frac{w^2 \lambda}{2}$.

Therefore, if $n \geq \frac{w^2 \lambda}{2}$, we set

$$s = \frac{1}{(1 + \alpha w) \alpha} \cdot \sqrt{\frac{\lambda n}{2}}; \quad c = 2 \left(\frac{1}{\alpha} + w \right) \sqrt{\frac{2 \lambda}{n}} + 2.$$

Else, we set

$$s = \frac{n}{\alpha w \cdot (1 + \alpha w)}; \quad c = \left(\frac{1}{\alpha} + w \right) \cdot \frac{w \lambda}{n} + 2 \left(1 + \frac{1}{\alpha w} \right) + 2.$$

□

Lemma 51. Let $0 < \alpha \leq 1$, $w, n \in \mathbb{N}$, $x > 0$, Y_i be Poisson random variables with expectation 1, and define

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

Then

$$\Pr \left[\frac{1}{n} \sum_{i=1}^n B_i \geq x \right] \leq \frac{(w+2) \cdot e^{\alpha(w+1)}}{e \cdot (w+2 - e^\alpha) \cdot (w+1)! \cdot x}.$$

Proof. First we bound the following:

$$\begin{aligned} \mathbb{E}[B_i] &= \\ &= \sum_{j=0}^{\infty} \mathbb{E}[B_i | Y_i = j] \cdot \Pr[Y_i = j] = \\ &= \sum_{j=w+1}^{\infty} e^{\alpha j} \cdot \Pr[Y_i = j] = \\ &= \sum_{j=w+1}^{\infty} \frac{e^{\alpha j}}{e j!} = \\ &= \sum_{j=0}^{\infty} \frac{e^{\alpha(w+1+j)}}{e \cdot (w+1+j)!} \leq \\ &= \frac{e^{\alpha(w+1)}}{e \cdot (w+1)!} \sum_{j=0}^{\infty} \frac{e^{\alpha j}}{(w+2)^j} = \\ &= \frac{e^{\alpha(w+1)}}{e \cdot (w+1)!} \sum_{j=0}^{\infty} \left(\frac{e^\alpha}{w+2} \right)^j \stackrel{[=]}{=} \end{aligned}$$

Since $\alpha \leq 1$ and $w \geq 1$, $e^\alpha \leq w+2$; then

$$\begin{aligned} \stackrel{[=]}{=} & \frac{e^{\alpha(w+1)}}{e \cdot (w+1)!} \cdot \frac{1}{1 - \frac{e^\alpha}{w+2}} = \\ & \frac{(w+2) \cdot e^{\alpha(w+1)}}{e \cdot (w+2 - e^\alpha) \cdot (w+1)!} \end{aligned}$$

Then by Markov's inequality,

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

□

E.6 Lemmas for Section 3.2.2

Lemma 52. *Assume*

$$c_0 \geq 2; \quad c_1 \geq 1; \quad \lambda > 0; \quad d \geq 4c_0u\lambda; \quad q = \frac{2\lambda}{d}; \quad B = \frac{2c_1(u+1)d}{\lambda}.$$

*Then the (DFS) algorithm visits less than B vertices **and** finds a valid proof with probability at least*

$$1 - \frac{2}{c_0} - \frac{1}{c_1} - e^{-\lambda} + \frac{2}{c_0} \cdot \left(\frac{1}{c_1} + e^{-\lambda} \right).$$

Proof. Let $X_i = |\{s \in S_p : H_0(s) = i\}|$ be the number of balls in bin i and let E be the event that $\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \leq e^{-q+c_0q^2}$. By Lemma 4,

$$\Pr \left[\bar{E} \right] = \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} > e^{-q+c_0q^2} \right] \leq \Pr \left[\frac{1}{n_p} \sum_{i=1}^{n_p} e^{-qX_i} \geq 1 - q + c_0q^2 \right] \leq \frac{2}{c_0}.$$

Let F be the event that the honest prover fails. By Lemma 43, $\Pr[F|E] \leq e^{-\lambda}$. Finally, let V be the number of visited vertices. By Lemma 7 with $c := c_0$,

$$\mathbb{E} [V|E] \leq \frac{2(u+1)}{q - c_0uq^2} \leq \frac{2(u+1)}{\frac{q}{2}} = \frac{2(u+1)d}{\lambda},$$

and by Markov's inequality, $\Pr [V \geq B|E] \leq \frac{1}{c_1}$.

Hence, $\Pr [F \vee (V \geq B)|E] \leq e^{-\lambda} + \frac{1}{c_1}$, and by Lemma 34,

$$\Pr [F \vee (V \geq B)] \leq \frac{2}{c_0} + \frac{1}{c_1} + e^{-\lambda} - \frac{2}{c_0} \cdot \left(\frac{1}{c_1} + e^{-\lambda} \right).$$

□

E.7 Lemmas for Section 4.2

Lemma 53. *Assume*

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

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

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

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

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

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

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

$$\begin{aligned}
& \frac{1}{2} \ln k + nD\left(\frac{k-1}{n} \parallel p\right) = \\
& \frac{1}{2} \ln k + nD\left((1-\delta)p - \frac{1}{n} \parallel p\right) = \\
& \frac{1}{2} \ln k + n\left(\left((1-\delta)p - \frac{1}{n}\right) \ln \frac{(1-\delta)p - \frac{1}{n}}{p} + \left(1 - (1-\delta)p + \frac{1}{n}\right) \ln \frac{1 - (1-\delta)p + \frac{1}{n}}{1-p}\right) \leq \\
& \frac{1}{2} \ln k + n\left((1-\delta)p \cdot \ln(1-\delta) + \left(1 - (1-\delta)p + \frac{1}{n}\right) \ln \frac{1 - (1-\delta)p + \frac{1}{n}}{1-p}\right) = \\
& \frac{1}{2} \ln k + n\left((1-\delta)p \cdot \ln(1-\delta) + \left(1-p + \delta p + \frac{1}{n}\right) \ln\left(1 + \frac{\delta p + \frac{1}{n}}{1-p}\right)\right) \leq \\
& \frac{1}{2} \ln k + n\left((1-\delta)p \cdot \ln(1-\delta) + \left(1-p + \delta p + \frac{1}{n}\right) \cdot \frac{\delta p + \frac{1}{n}}{1-p}\right) = \\
& \frac{1}{2} \ln k + n\left((1-\delta)p \cdot \ln(1-\delta) + \left(1 + \frac{\delta p + \frac{1}{n}}{1-p}\right) \cdot \left(\delta p + \frac{1}{n}\right)\right) = \\
& \frac{1}{2} \ln k + n\left((1-\delta)p \cdot \ln(1-\delta) + \left(1 + \frac{p}{1-p}\delta + \frac{1}{(1-p)n}\right) \cdot \left(\delta p + \frac{1}{n}\right)\right) [\leq]
\end{aligned}$$

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

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

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

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

which is true by the assumption about μ . □