Random Variable Commitments for Any Sampleable Distribution, and a Certified Laplace Mechanism

Fredrik Meisingseth[1 [0009−0003−7316−9341]], Christian Rechberger[1 [0000−0003−1280−6020]], and Fabian Schmid[1 [0009−0005−7783−0859]]

Graz University of Technology, Austria
{fredrik.meisingseth,christian.rechberger,fabian.schmid}@tugraz.at

Abstract. At CRYPTO'24, Bell et al. propose a definition of *random variable commitment schemes (RVCS)* and show that it leads to a notion of *certified* differential privacy (DP). We prove that there exists RVCS's for all efficiently sampleable distributions, under the discrete logarithm assumption. This follows from three lemmata that additionally enable simple modular design of new RVCS's; First, we show that the properties of RVCS's are closed under polynomial sequential composition. Secondly, we show that homomorphically evaluating a function f on the output of an RVCS for distribution Z leads to an RVCS for distribution $f(Z)$. Thirdly, we show that applying a 'Commit-and-Prove'-style argument of knowledge for a function f onto the output of an RVCS for distribution Z results in an RVCS for distribution $f(Z)$. Further, we observe that Bell et al. use Σ-protocols and a knowledge soundness property seemingly significantly stronger than those typically used when studying Σ-protocols. To align the RVCS definition with the literature on Σ-protocols, we relax it slightly and show that known constructions fulfill this adapted definition. We propose another orthogonal relaxation which allows for sampling algorithms that abort with non-zero probability. We demonstrate the usefulness of the lemmata and definitional adaptations by constructing the first RVCS's for arbitrarily biased coins and a discrete Laplace distribution, leading to the first certified DP protocol for a discrete Laplace mechanism. Practicality is shown via an open-source Rust implementation.

1 Introduction

The central ('curator') model of *differential privacy (DP)* [34] involves two parties; the trusted *curator*, who holds a secret database and wishes to publish some function evaluated on the database, and an untrusted *analyst*, who may want to learn undue information about the database by analyzing this function evaluation. In order for the curator to be able to publish the function evaluation whilst avoiding that a possibly malicious analyst is able to infer unwanted information, the curator introduces some kind of noise to the function evaluation and thus (when done properly) achieves DP. One may naturally want to remove some of

the need for trusting the curator and, in particular, it might be desirable for the curator to be able to prove that it has indeed provided the supposed privacy and accuracy guarantees [16,12]. The ability to publicly[1] certify that a specific DP mechanism (or one with similar DP and utility properties) has been followed is especially important if the curator has vested interest in what the function appears to have evaluated to and at the same time has a duty to transparently guarantee some level of privacy protection. One example of such a curator could be a company publishing statistics about its customers (whose privacy-protection might be demanded by law) but at the same time these statistics may indicate the performance of the company.

In a paper at CRYPTO'24 [12], henceforth referred to as *BGKW*, Bell, Goldwasser, Kim and Watson propose a definition of *Certified DP (CertDP)* and also a blueprint for designing protocols that fulfill the definition. This blueprint forms a stack of three primitives/definitions; *Random Variable Commitment Schemes (RVCS)*, *Certified Probabilistic Mechanisms (CertPM)* and *Certified DP (CertDP)*.[2] Informally, the definitions may be understood as follows:[3]

- **RVCS:** A *dealer* and a *player* interact. The result is a commitment, which the dealer can open, to some sampled value and the player is convinced that the sample is from a specific distribution Z. An efficient cheating dealer can make the distribution of the sample deviate from Z by any statistical distance, but this is detected with probability roughly that of the deviation. Similarly, a cheating player can cause an arbitrary deviation, however only by choosing to abort independently of the sampled value.
- **CertPM:** A *prover* and a *verifier* interact. The result is an evaluation of a probabilistic mechanism $\mathcal{M}$ on input D from the prover. The main differences to an RVCS are that the output distribution is dependent on the secret input, and that the output is in the plain rather than a commitment. Assurances about the output distribution are essentially as in RVCS. Furthermore, the verifier learns no information about the prover's input or randomness, except for what can be learned from the output.
- **CertDP:** A CertDP protocol is between the curator and the analyst. The analyst should receive the output of a DP mechanism with given privacy parameters and accuracy guarantees for a query function in some class of queries. In essence, a CertDP protocol is a CertPM protocol where the mechanism is DP and accurate. Security-wise, no cheating analyst should be able to cause the protocol to compromise the DP guarantees. An efficient cheating

[1] Public verifiability is not demanded by the definitions treated in this paper, but is fulfilled by all constructions and might be a crucial property in practice. This would be important for governmental agencies (like the US Census) wishing to prove to the public (who cannot all take part in the protocol) that DP has been used in a correct way.

[2] We abbreviate certified DP by 'CertDP' rather than 'CDP' as in BGKW [12] because 'CDP' is commonly used for both *Computational DP* [46,11] and *Concentrated DP* [19]. We use 'CertPM' instead of 'CPM' for consistency.

[3] In a DP setting, the first party is run by the curator and the second by the analyst.

curator can however compromise both the accuracy and privacy guarantees but this is detected by the honest analyst with probability proportional to the loss in privacy or accuracy parameters.

The blueprint for constructing certified additive-noise DP mechanisms consists of two main results:

- **CertPM from RVCS:** Given an additively homomorphic commitment scheme CS that allows committing to a database D and homomorphically evaluating a function class F on D and an RVCS for the distribution Z, there exists a CertPM protocol for the additive-noise mechanism $\mathcal{M}_f(D) := f(D) + Z$, for any $f \in F$.
- **CertDP from CertPM:** Any CertPM protocol for an accurate (ε, δ)-DP mechanism $\mathcal{M}$ is an accurate (ε, δ)-CertDP protocol.

In BGKW [12], two RVCS's are presented – one for fair coins and one for a binomial distribution. We call these CoinCS and BinCS, respectively. In BinCS, fair coins from CoinCS are simply added up homomorphically. Having an RVCS for the binomial distribution yields a CertDP protocol via the binomial mechanism [33]. Both constructions rely solely on additively homomorphic commitments, instantiated by Pedersen commitments [47], and witness-indistinguishable arguments of knowledge (WI-AoKs), instantiated by Σ-protocols [29,27]. The protocols are therefore highly efficient. A major open question, prior to this work, about RVCS was whether there exists an RVCS for any distribution that can be efficiently sampled. For CertDP specifically, there were no known RVCS's for the noise distributions mostly used in practice (say, versions of Laplace and Gaussian noise [14,52,40]) and it was unknown if such can be constructed using standard cryptographic techniques. Thus, the main question we treat is:

Does there exist, under standard assumptions, RVCS's for any polynomial-time sampleable distribution?

We resolve this question in the affirmative, given a minor adaptation of the RVCS definition. Concretely, under the discrete logarithm assumption, there exists an RVCS for any distribution over $\mathbb{Z}_q$ which can be sampled by an arithmetic circuit taking polynomially many uniform bits as input.

1.1 Contributions

Adjusting the RVCS definition to follow from special-soundness. We observe that the knowledge soundness property demanded of AoKs in BGKW is significantly stronger than those typically used when studying Σ-protocols and similar AoKs. In particular, it requires a *strict* PPT black-box extractor which succeeds either with negligible or overwhelming probability. It is, to the best of our knowledge, unknown if Σ-protocols can indeed fulfill such a version of knowledge soundness, as BGKW gives no proof of this and the references therein [43,29] treat weaker notions of knowledge soundness.

We circumvent this issue by instead using a weaker knowledge soundness definition [41], which is known to follow from special-soundness (a property Σ-protocols per definition satisfy [27,29]). To accommodate for this, we relax the RVCS definition to require an extractor with expected, rather than strict, polynomial runtime. We make such an adaptation in Section 4, prove that known constructions satisfy it and that it implies CertPM and CertDP in the same way as the original definition. From now on, when we say 'RVCS', we mean this adapted definition, unless otherwise specified.

Adjusting definitions to allow honest aborts. Many algorithms for sampling have a non-zero abort probability, due to using some version of rejection sampling [23,51,35]. This becomes problematic since the CertDP definition explicitly does not allow honest aborts. We therefore propose additional slight relaxations of the RVCS, CertPM and CertDP definitions in Section 6. For clarity, we call these *statistical* RVCS/CertPM/CertDP.[4] Again, these relaxed definitions allow building a statistical CertDP protocol from a statistical RVCS. When distinguishing the definitions, we call the non-statistical ones *perfect*.

Modularity Lemmata. In Section 5, we prove three technical lemmata about the RVCS primitive, the first one about composition and the other two about post-processing. The lemmata are proven with respect to our adapted definition in Section 4, but apply (with minor changes in the proofs) also to the original definition and the variation from Section 6.

Lemma 1 (Informal version of Lemma 5). *If polynomially many RVCS's for distributions $Z_1, \ldots, Z_k$ are run sequentially, then this composed protocol forms an RVCS for the joint distribution $Z_1 \times \ldots \times Z_k$.*

Lemma 2 (Informal version of Lemma 6). *If the output of an RVCS for distribution Z is post-processed solely by the player homomorphically computing some function f on the commitment from the base RVCS, then this composed protocol is an RVCS for the distribution $f(Z)$.*

Lemma 3 (Informal version of Lemma 7). *If one uses a 'commit-and-prove'-style WI-AoK that some function f has been applied to the sample from an RVCS for distribution Z then one gets an RVCS for distribution $f(Z)$.*

From these lemmata follows Corollary 1, which resolves the question of the existence of RVCS's for arbitrary (efficiently sampleable) distributions.

[4] As with standard commitment schemes, RVCS has two security properties, each of which can be perfect, statistical or computational. Here, statistical refers to security against cheating players, as opposed to the perfect security of the original definition.

RVCS's for biased coins and Laplace distribution. The lemmata also give us a straight-forward way to construct an RVCS from any known sampling algorithm which can be represented as a circuit with binary inputs. To showcase this, we use of the lemmata and the adapted definitions above to construct (Section 7) an RVCS for arbitrarily biased coins, BerCS, and a statistical RVCS for a discrete Laplace distribution, DLapCS. These are the first RVCS's for their respective distributions and rely only on additively homomorphic commitments and WI-AoKs with special-soundness. BerCS builds on the RVCS for fair coins, CoinCS, and a WI-AoK for product relations between commitments. BerCS can approximate a $Ber(p)$ distribution for any $p \in (0, 1)$ up to exponentially small statistical distance in linear time. DLapCS builds on BerCS and is based on a protocol for the FDL_1 distribution from [35], which when used in an additive-noise DP mechanism has essentially the same utility-privacy trade-off as the standard Laplace mechanism [34]. We can instantiate DLapCS for any $\varepsilon > 0$ and parameterize the protocol as to approximate FDL_1 up to negligible statistical distance in $O(\kappa \log(\kappa))$ time, with κ being the security parameter.

We provide an open-sourced Rust implementation[5] of these new RVCS's, as well as CoinCS and BinCS. Benchmarks of the implementation shows that the constructions are practical, with it taking around one second to run DLapCS. A detailed breakdown of costs is given in Appendix E.

2 Technical Overview

2.1 On the Notion of Knowledge Soundness

We start by considering the original definition of RVCS [12]. We denote the output of a protocol by a tuple, with subscripts denoting party-specific outputs. The result (the only public output), C^{out}, of an RVCS is given by the player at the end of the protocol. At some point during the interaction, the dealer is said to *propose an output*, denoted C^Q, supposedly being a commitment to a sample from the intended distribution. At the end of the protocol, the player either accepts the proposed output and outputs C^Q, or rejects it, outputting $\perp$ (denoting abort). For a detailed explanation of the notation we use, see Section 3.

Definition 1 (BGKW [12]). *A* random variable commitment scheme *(RVCS) for distribution Z consists of a commitment scheme $\mathsf{CS} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$, a dealer DE and a player PL, which fulfill the following when $pp \leftarrow \mathsf{Setup}(1^\kappa)$:*

1. **Correctness:** *For $(C^{out}, (r^Q, z^Q)_{\mathsf{DE}}) \leftarrow \langle \mathsf{DE}, \mathsf{PL} \rangle(pp)$ and $z \leftarrow Z$:*
$$(C^{out}, r^Q) \sim \mathsf{Commit}(pp, z).$$
2. **Cheating-player distributional soundness:** *For every (potentially unbounded) adversary $\tilde{\mathsf{PL}}$, let $(C^{out}, (r^Q, z^Q)_{\mathsf{DE}}) \leftarrow \langle \mathsf{DE}, \tilde{\mathsf{PL}} \rangle(pp)$ and*
$$z^{out}_{C \neq \perp} \leftarrow \mathsf{Verify}(pp, C^{out}, r^Q, z^Q | C^{out} \neq \perp).$$
Then $z^{out}_{C \neq \perp} \sim Z$.

[5] https://github.com/Fable95/certified-laplace

3. **Cheating-dealer distributional soundness:** *There exists a PPT algorithm* $\mathsf{RevealOpening}$ *which is given the view of the dealer and rewindable black-box access to the dealer such that the following holds. For any PPT* $\tilde{\mathsf{DE}}$*, let* $(C^{out}, \mathsf{VIEW}_{\tilde{\mathsf{DE}}}) \leftarrow \langle \tilde{\mathsf{DE}}, \mathsf{PL} \rangle(pp)$ *and* $(r^Q, z^Q) \leftarrow \mathsf{RevealOpening}(\mathsf{VIEW}_{\tilde{\mathsf{DE}}})$*. Let* $z^{out} \leftarrow \mathsf{Verify}(pp, C^{out}, r^Q, z^Q)$ *and* $\gamma_\perp = \mathbb{P}(C^{out} = \perp)$*. Then:*

$$z^{out} \approx^s_{\gamma_\perp + \mathsf{negl}} Z.$$

Correctness says that the result of an honest execution has the same distribution as a commitment to a sample from the target distribution Z, and that the dealer holds the corresponding opening information r^Q. Cheating-player distributional soundness says, essentially, that no cheating player can change the value to which the dealer can open the final commitment in any way except for aborting the protocol, and unless they abort the sample will have the correct distribution. Cheating-dealer distributional soundness essentially requires the existence of an efficient extractor ($\mathsf{RevealOpening}$) which from any efficient cheating dealer can extract an opening to the final commitment. The result of this opening will have a distribution close to the target distribution, in the sense that their statistical distance will be at most roughly the probability that the cheating dealer causes the verifier to abort.

The construction of an RVCS for fair coins, CoinCS, mimics the classical *coin-toss in the well* [17] protocol and is instructive for our discussion[6]:

1. DE commits to a random bit and proves using an AoK that it has committed to a binary value.
2. If PL accepts this AoK, then they send a random bit (in the plain) to DE.
3. Both parties homomorphically compute the XOR between the commitment sent by DE and the bit sent by PL.

The **proof strategy** for showing that CoinCS is an RVCS goes roughly as follows. Correctness follows directly if CS is additively homomorphic and the AoK is perfectly complete. Cheating-player distributional soundness also follows easily from that CS is perfectly hiding and the AoK is perfectly witness indistinguishable, since the view of PL will be independent of the value that DE committed to in the beginning. Cheating-dealer distributional soundness is a bit more complicated to show, but should intuitively follow from the binding property of CS and the knowledge soundness of the AoK, since the extractor of the AoK should be able to extract a unique valid opening to the initial commitment whenever the cheating dealer convinces the verifier. The randomness of the verifier then assures that the sample underlying the final commitment will have the supposed distribution.

Our first contribution starts with the observation that in BGKW, a particularly strong notion of knowledge soundness is used.[7]

[6] We recall CoinCS more formally in Section 4.

[7] No quantifiers for the negligible function(s) are given in BGKW. We leniently interpret this as meaning that when the accept probability is not negligible, then there exists some negligible function such that the extraction probability is overwhelming.

Definition 2 (BGKW [12]). *A proof system* $(\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *for relation* $\mathcal{R}$ *satisfies* knowledge soundness *if there exists a PPT knowledge extractor* K *such that for any PPT cheating prover* $\tilde{\mathsf{P}}$ *with* $\mathbb{P}\big(1 \leftarrow \langle \tilde{\mathsf{P}}, \mathsf{V} \rangle(pp, x)\big) > \mathsf{negl}(\kappa)$*,* K *outputs, upon rewindable black-box access*[8] *to* $\tilde{\mathsf{P}}$ *and input* x*, a witness* w *such that* $\mathbb{P}((x, w) \in \mathcal{R}) = 1 - \mathsf{negl}(\kappa)$.

There are two crucial aspects of the definition; First, the extractor runs in strict rather than expected polynomial time, regardless of the probability that the verifier accepts. Second, the extractor has an overwhelming rather than only non-negligible success probability whenever the verifier accepts with non-negligible probability.[9]

The issue with relying on Definition 2 is that the AoKs used in BGKW, Σ-protocols, and their modern successors are not studied with respect to it [3,30,4]. In fact, the references stated in BGKW ([29,43]) only prove weaker knowledge soundness properties, either special-soundness or knowledge soundness where the extractor has a bounded *expected* runtime. We have since not been able to find any other work showing that Σ-protocols fulfill Definition 2. Therefore, we consider it an open problem whether or not Σ-protocols, either in general or the specific ones used in BGKW, fulfill Definition 2.

There is evidence (though, to the best of our knowledge, no proof) that Σ-protocols do not generally imply a strict PPT black-box extractor with overwhelming success probability (whenever the cheating dealer succeeds with non-negligible probability), and therefore cannot fulfill Definition 2. Firstly, there is long line of work studying the types of knowledge soundness one can get from (various versions of) special-soundness [27,18,39,6,41], none of which give an extractor strong enough to fulfill Definition 2. Secondly, Definition 2 bears large similarity to strong knowledge soundness as in [8] and it is known that, under a plausible complexity assumption, protocols fulfilling (several versions of that definition) cannot be constant-round, unless the relation proven is trivial.

We get around this issue by instead relying on a definition of knowledge soundness known to be satisfied by Σ-protocols, i.e. one that follows from special-soundness [27]. The most standard versions of knowledge soundness [13,38] seem not to suffice directly, since there either the knowledge extractor K only succeeds with probability roughly equal to that of the verifier accepting the argument (which would be too low) or the extractor does not necessarily have (expected) polynomial runtime in the security parameter. Rather, we will use a definition [41] demanding that *if* the extractor K is given an accepting transcript, then it will successfully extract with overwhelming probability, but nothing is required of K if it is given a rejecting transcript.[10] Since this definition requires an *expected* polynomial-time extractor, rather than a strict polynomial-time one

[8] During rewinding, K can run $\tilde{\mathsf{P}}$ with the random string of its choice.

[9] We distinguish between 'non-negligible', meaning simply 'not negligible', and 'noticeable', meaning of the form $1/poly(\kappa)$. There are non-negligible functions which are not noticeable [38].

[10] This only implies that K will successfully extract with probability roughly that of the cheating prover convincing the verifier, when one does not condition on K being

(as Definition 2), we must adapt the RVCS definition to allow RevealOpening to be expected polynomial time. We do this in Section 4, and show that the proof strategy sketched above indeed works for the definitions we use.

Aligning the RVCS definition with a knowledge soundness property known to follow from a generalized special-soundness property has the additional benefit of enabling the future use of more efficient AoKs, like compressed Σ-protocols [4].

2.2 Modularity Lemmata

When we have re-confirmed the existence of an RVCS for fair coins, we want to use that to build RVCS's for more complicated distributions. It is noted in BGKW that CoinCS can be composed in parallel to get an RVCS for many fair coins. It is similarly noted that the resulting commitments can be homomorphically added up, and that this gives an RVCS for a binomial distribution. We formalize and generalize these remarks to properties of the RVCS definition, regardless of any specific construction.

For composition, we treat sequential composition (Lemma 5) of some polynomial number of (potentially different[11]) RVCS's. Correctness follows directly from the correctness of the individual RVCS's. Both cheating-player and cheating-dealer distributional soundness follows from that the cheating party can simulate all of the sub-protocols except for one, say the ith RVCS, and will thus get a view which is identically distributed to their view before entering the ith RVCS in the composed protocol. This emulation takes polynomial time (as honest parties are PPT) and we therefore get that if the composed protocol is not an RVCS for the joint target distribution, then at least one of the individual sub-protocols is not an RVCS, which is a contradiction.

For homomorphic operations on the output of an RVCS (Lemma 6), all properties are straight-forward to show. Correctness follows directly from the homomorphic property of CS and cheating-player distributional soundness follows directly from the fact that there is no interaction during the homomorphic computations, so they give the player no opportunity to influence what values the player can open the result to. Cheating-dealer distributional soundness follows from the fact that a statistical distance cannot increase from a function being applied to the two compared distributions.

With these two properties and additively homomorphic commitments, we can easily construct an RVCS for any distribution that can be expressed as a linear function of fair coins, by composing and homomorphically post-processing CoinCS.[12] However, sampling algorithms relevant to cryptography typically re-

given an accepting transcript. In this sense, the definition that we use corresponds to the standard definition [13].

[11] This generality is the reason that we consider sequential and not parallel composition.

[12] If one has a commitment scheme which is both additively and multiplicatively homomorphic then this gives a way to construct an RVCS for any distribution computable by a polynomial-sized circuit. There are proposals for such schemes [24] but no efficient implementations, so constructing RVCS's for general distributions in this way is currently unlikely to be practical.

quire transforming the fair coins in a non-linear way. To enable this, we observe that there are AoKs for proving various relations between committed values, such as one value being the product of two other values. These AoKs follow the 'Commit-and-Prove' paradigm [28,30,49], where the prover commits to the inputs and output of some circuit and then prove that the commitment of the output is indeed a commitment to the result of the circuit applied to the values committed to as inputs. Now if the commitments of the inputs are from an RVCS of a distribution Z, then the protocol where they are fed into a commit-and-prove AoK for some function f should be an RVCS for the distribution $f(Z)$.

In the commit-and-prove (CnP) post-processing lemma (Lemma 7) we prove that this is indeed the case. Again, correctness is straight-forward to show and follows from the correctness of the initial RVCS and the perfect completeness of the AoK. Similarly, cheating-player distributional soundness follows from that the view of the the player in the initial RVCS is independent of the sampled value, and that the AoK is perfectly witness indistinguishable. Proving cheating-dealer distributional soundness is a bit more involved, since the knowledge soundness definition we use only guarantees extraction with high probability when the extractor is given an accepting transcript. Further, there are two extractors involved in the construction, one for extracting the sample of the initial RVCS and one for extracting the final sample, whereas we need to prove that there exists a single extractor for the whole protocol. We let the final extractor $\mathsf{RevealOpening}$ simply run the extractor $\mathsf{RevealOpening}_{base}$ of the initial RVCS, getting some result z_{base} out. Then $\mathsf{RevealOpening}$ will output $f(z_{base})$ if there were no aborts, and $\perp$ otherwise. This guarantees, very roughly, that the output of $\mathsf{RevealOpening}$ will have the right distribution, but it does not tie this sample to the final commitment of the full protocol. For this, we need the extractor K of the CnP AoK, which guarantees correct openings of both the commitments from the initial RVCS and the final commitment to the result, as well as that they have the correct relation. This means that there exists two efficient extractors for the commitments from the initial RVCS, so by binding, the openings must be to the same message with overwhelming probability. Thus the output of $\mathsf{RevealOpening}$ must also be a correct opening of the final commitment.

Now finally, since commit-and-prove WI-AoKs exist (as Σ-protocols) for arithmetic circuits with binary inputs [28], using Pedersen commitments, we can construct an RVCS for any polynomial-time sampleable distribution over the message space $\mathbb{Z}_q$ of the commitment scheme, under the discrete logarithm assumption. Particularly, one simply needs to generate the uniformly random seed by using CoinCS, and then apply a suitable CnP AoK.

2.3 Relaxing RVCS to Allow Honest Aborts

With the result that an RVCS for any (efficiently sampleable) distribution can be built by using CoinCS and homomorphic and/or commit-and-prove post-processing, one may hope to construct a specific RVCS for some distribution of interest by simply expressing a known sampling algorithm as a circuit and then applying the correspond CnP AoK. One issue that arises here is that many

of the sampling algorithms for, say, discrete Laplace or Gaussian distributions rely on rejection sampling [23,35,51]. Thus, unless allowed to run indefinitely, they will terminate without result with some non-zero probability. Luckily, such algorithms can oftentimes be made to have a negligible abort probability, by repeating them a fixed number of times.

The RVCS definition (both the original and the one in Section 4) cannot be fulfilled for distributions where the abort symbol $\perp$ is part of the sample space. This is because the cheating-player distributional soundness property, which demands that if the output commitment C^{out} is not $\perp$ then it will open to a sample from precisely the target distribution Z. But if Z contains $\perp$ in its sample space, then conditioning on $C^{out} \neq \perp$ will lead to a statistical distance equal to the probability that samples from Z are $\perp$, i.e. the honest abort probability. This same issue is present for certifiable DP (Definition 16), where it is explicitly stated that the target mechanism never outputs $\perp$. We therefore propose (Section 6) additional adaptations of the RVCS, CertPM and CertDP definitions, called statistical RVCS/CertPM/CertDP, so that they may be easily satisfied for distributions and mechanisms that output $\perp$ with negligible probability also in honest executions. For statistical RVCS, the change we make is the one sketched above, i.e. demanding statistical indistinguishability rather than perfect indistingushability in the cheating-player distributional soundness property. The change in CertPM is analogous. For CertDP, we simply remove the requirement that the target DP mechanism never aborts.

2.4 New Random Variable Commitment Schemes

Having proven the modularity lemmata, given adapted definitions, and proven that they can be achieved, we go on to showcase the utility of these by constructing an RVCS for biased coins and a statistical RVCS for a discrete Laplace distribution (Section 7). As the goal is to show that one can easily turn existing sampling algorithms into an RVCS, we use sampling algorithms from the literature, rather than constructing new ones tailored to an RVCS setting. For biased coins, we give a construction that generates many fair coins using CoinCS and then post-process these by using homomorphisms and CnP AoKs for multiplicative relations. For a discrete Laplace distribution, we implement a sampling algorithm from [35] in which, essentially, many biased coins are added up and then the sample is rejected if the result falls outside of a specific range. This sampling algorithm is not only relatively simple to present, but the property that it (with small probability) rejects samples (outputting $\perp$) also gives us a chance to demonstrate usage of the statistical RVCS definition. As discrete Laplace distributions give DP when used as noise, we also get an sCertDP Laplace mechanism.

3 Preliminaries and Notation

The security parameter is κ and negl is an unspecified negligible function. A probability is *overwhelming* if it is $1 - \mathsf{negl}$. PPT means probabilistic polyno-

mial time and *EPT* means probabilistic expected polynomial time. When differentiating between PPT and EPT, we write *strict* PPT and *expected* PPT, respectively. $\mathbf{1}_S$ is the indicator function for some set S. When instantiating our protocols, we assume that the mechanism range R is $\mathbb{Z}_q \cup \{\perp\}$ with $\mathbb{Z}_q = \{x \in \mathbb{Z} : \lceil -q/2 \rceil \leq x \leq \lfloor q/2 \rfloor\}$ for some large prime q and $\perp$ denoting abort. For $n \in \mathbb{N}$, $[n] := \{1, 2, \ldots, n\}$. We denote distributions by capital letters and samples by the corresponding lower case letter. For instance, $z \leftarrow Z$ says that z is a sample from distribution Z and z is a random variable with distribution Z. For a set S, $z \leftarrow S$ denotes uniform sampling from S. For a distribution Z, $Z_{\neq \perp}$ denotes Z conditioned on not being $\perp$. For distributions X and Y, $X \sim Y$ denotes that they are identical. We use $SD(X, Y)$ to denote the statistical (total variation) distance between X and Y and $X \approx^s_\alpha Y$ means that the statistical distance is upper bounded by α. The Bernoulli distribution $Ber(p)$ with $p \in [0, 1]$ is the distribution on $\{0, 1\}$ with probability mass function (pmf) $f_{Ber}(z) := p^z(1-p)^{1-z}$. We later rely on the three following basic properties of statistical distances.

Proposition 1. *For any probability distributions X, Y, Z on a sample space R:*

1. *(Triangle inequality) We have $SD(X, Y) \leq SD(X, Z) + SD(Z, Y)$.*
2. *For any function $f : R \to R$ we have $SD(X, Y) \geq SD(f(X), f(Y))$.*
3. *For joint distributions $X := X_1 \times \ldots \times X_k$ and $Y := Y_1 \times \ldots \times Y_k$, we have $SD(X, Y) \leq \sum_{i \in [k]} SD(X_i, Y_i)$.*

As our focus is on the RVCS primitive, we defer additional background about DP and mechanism accuracy to Appendix A.

3.1 Commitment Schemes

We use homomorphic commitments and for a function f on messages, we denote the corresponding operation on commitments and opening information by $\hat{f}$ and $\check{f}$, respectively.

Definition 3. *A* commitment scheme CS *is a triple* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ *of PPT algorithms.* Setup *maps* 1^κ *to a set of public parameters* pp. Commit *maps a value* m *in a message space* χ *to a* commitment C_m *in a commitment space* $\mathcal{C}$ *and* opening information r_m *in a randomness space* Γ. Verify *maps public parameters, a commitment, a opening information and a value* m *to either* m *or* $\perp$. *The scheme fulfills the following properties.*

1. **Correctness:** *For $pp \leftarrow \mathsf{Setup}(1^\kappa)$ and every $m \in \chi$, we have*
$$\mathbb{P}\Big(\mathsf{Verify}(pp, C_m, r_m, m) = m | (C_m, r_m) \leftarrow \mathsf{Commit}(pp, m)\Big) = 1.$$
2. **(Perfect) Hiding:** *For every stateful algorithm $\mathcal{A}$, we have:*
$$\mathbb{P}\left(b' = b \middle| \begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\kappa), b \leftarrow \{0,1\} \\ (m_0, m_1) \leftarrow \mathcal{A}(1^\kappa, pp), (C, r) \leftarrow \mathsf{Commit}(pp, m_b) \\ b' \leftarrow \mathcal{A}(C) \end{array}\right) = 1/2.$$
That is, the distribution of C is independent of the message.

3. **(Computational) Binding:** *There exists a negligible function* negl *such that for any EPT* $\mathcal{A}$*, we have:*

$$\mathbb{P}\left(\begin{matrix} m \neq m' \wedge \mathsf{Verify}(pp, C, m, r) = m \\ \wedge \mathsf{Verify}(pp, C, m', r') = m' \end{matrix} \middle| \begin{matrix} pp \leftarrow \mathsf{Setup}(1^\kappa) \\ (C, m, r, m', r') \leftarrow \mathcal{A}(pp) \end{matrix}\right) \leq \mathsf{negl}(\kappa).$$

When committing to a vector $\mathbf{m} = (m_1, \ldots, m_k)$ of messages, $\mathsf{Commit}(\mathbf{m})$ denotes the concatenation of commitments to each element and we have $C_{\mathbf{m}} = (C_{m_1}, \ldots, C_{m_k})$ and $r_{\mathbf{m}} = (r_{m_1}, \ldots, r_{m_k})$.

Definition 4. *For* $k \in \mathbb{N}$ *and a function* $f : \chi^k \to \chi$*, a commitment scheme* CS *is* f-homomorphic *if there exists deterministic operations* $\hat{f} : \mathcal{C}^k \to \mathcal{C}$ *and* $\check{f} : \Gamma^k \to \Gamma$ *that fulfill:*

1. **Homomorphic correctness:** *For every* $\mathbf{m} \in \chi^k$*, we have*

$$\mathbb{P}\Bigg(\mathsf{Verify}\big(pp, \hat{f}(C_{\mathbf{m}}), \check{f}(r_{\mathbf{m}}), f(\mathbf{m})\big) = f(\mathbf{m}) \\ \Bigg| pp \leftarrow \mathsf{Setup}(1^\kappa), (C_{\mathbf{m}}, r_{\mathbf{m}}) \leftarrow \mathsf{Commit}(pp, \mathbf{m})\Bigg) = 1.$$

2. **(Perfect) Homomorphic hiding:** *For every stateful algorithm* $\mathcal{A}$*, we have:*

$$\mathbb{P}\left(b' = b \middle| \begin{matrix} pp \leftarrow \mathsf{Setup}(1^\kappa), b \leftarrow \{0,1\}, (\mathbf{m}^0, \mathbf{m}^1) \leftarrow \mathcal{A}(1^\kappa, pp) \\ (C_{\mathbf{m}^b}, r_{\mathbf{m}^b}) \leftarrow \mathsf{Commit}(pp, \mathbf{m}^b) \\ b' \leftarrow \mathcal{A}\big(\hat{f}(C), C_{\mathbf{m}^b}, \check{f}(r_{\mathbf{m}^b})\big) \end{matrix}\right) = 1/2.$$

A commitment scheme is *additive* if it is f-homomorphic for all f in a family F which includes all linear functions from χ^* to χ.

Our constructions rely on the discrete logarithm assumption (DLog). Setup is an algorithm mapping 1^κ to a description of a finite cyclic group G of prime order p, where $\log_2(p) \approx \kappa$, a generator g and a random group element h.

Definition 5. *The* discrete logarithm assumption (DLog) *with respect to* Setup *is that for every EPT* $\mathcal{A}$*, there exists a negligible function* negl *such that*

$$\mathbb{P}\big(h = g^x : x \leftarrow \mathcal{A}(1^\kappa, G, g, h), (G, g, h) \leftarrow \mathsf{Setup}(1^\kappa)\big) \leq \mathsf{negl}(\kappa)$$

Remark 1. We define binding (and DLog) against EPT adversaries, instead of PPT adversaries as in BGKW. This is common [31,39,21] and unproblematic since all falsifiable assumptionsfor strict PPT adversaries are equivalent to their formulation with expected PPT adversaries [42].

3.2 Arguments of Knowledge

We base the formalizations here on [41], with mostly cosmetic changes, and refer there for further details. For sets Par, X, W, a *parameter-dependent relation* $\mathcal{R}$ is a relation over $Par \times X \times W$. We call $\mathcal{R}$ an *NP-relation* if there exists a

polynomial-time algorithm which given $pp \in Par, x \in X, w \in W$ decides whether or not (pp, x, w) is in $\mathcal{R}$. Throughout, we only consider NP-relations and often let relations be implicitly parameter-dependent, writing rather $\mathcal{R}$ as over $X \times W$. We also never specify $X \times W$ as they are clear from context. It is common, for convenience, to study and design arguments of knowledge by altering the relation and instead analysing a corresponding proof of knowledge. The relation $\mathcal{R}$ is then transformed to some relaxed relation of the form $\mathcal{R}_{ext} := \{(x, w) \in \mathcal{R}$ *or a computationally hard problem has been solved*$\}$. Then by showing a proof of knowledge for $\mathcal{R}_{ext}$ one directly gets an argument of knowledge for $\mathcal{R}$, since the hard problem will only be solved with negligible probability by an efficient adversary [7,41]. We formalize relaxed relations as follows.

Definition 6. *For an NP-relation $\mathcal{R}$, we say that $\mathcal{R}_{ext}$ is a* relaxed relation *of $\mathcal{R}$ if $\mathcal{R}_{ext}$ is an NP-relation and of the form $\mathcal{R}_{ext} := \{(x, w) : (x, w) \in \mathcal{R} \vee (x, w) \in \mathcal{R}_{alt}\}$. We call $\mathcal{R}_{alt}$ the* extension *in $\mathcal{R}_{ext}$.*

Note that we allow $\mathcal{R} = \emptyset$ and thus $\mathcal{R}_{ext} = \mathcal{R}$.

Definition 7. *Let $\mathcal{R}$ be an NP-relation. A* proof system *for $\mathcal{R}$ is a triple $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ where Setup is a PPT algorithm and (P, V) are interactive PPT algorithms. The syntax of the system is as follows, where all algorithms take 1^κ as implicit parameter:*

- Setup *maps 1^κ to public parameters pp.*
- $\mathsf{P}(pp, x, w)$ *with $(x, w) \in \mathcal{R}$ interacts with V and gives no output.*
- $\mathsf{V}(pp, x)$ *interacts with P and gives an output $b \in \{0, 1\}$.*

We write $tr \leftarrow Tr\big(\langle \mathsf{P}(u), \mathsf{V}(u')\rangle(v)\big)$ for the transcript of the interaction between P and V where u, u' are private input and v is common input. We write $b \leftarrow \langle \mathsf{P}(u), \mathsf{V}(u')\rangle(v)$ for the output of the same interaction.

The system satisfies (perfect) completeness*: for every $(pp, x, w) \in \mathcal{R}$,*

$$\mathbb{P}\big(1 \leftarrow \langle \mathsf{P}(w), \mathsf{V}\rangle(pp, x)\big) = 1.$$

When V outputs 1, we say that they *accept* and when they output 0, they *reject*. Similarly, a transcript is either *accepting*, denoted $tr \in Acc(pp, x)$, or *rejecting*.

Definition 8. *For $\mu \in \mathbb{N}$, a $(2\mu + 1)$-round proof system is called* public-coin *if each of V's messages $c_1, \ldots, c_\mu$ are drawn uniformly at random from some sets $Chal_1, \ldots, Chal_\mu$, respectively. We call the messages from the verifier* challenges*.*

All the proof systems considered in this work are public-coin.

Definition 9 ([41]). *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a public-coin proof system for NP-relation $\mathcal{R}$ with relaxed relation $\mathcal{R}_{ext}$. Let K be EPT with rewindable black-box oracle access and it has implicit input 1^κ and explicit input pp, x, tr. Let $\mathcal{A}$ be a probabilistic algorithm and $\tilde{\mathsf{P}}$ be deterministic. Let*

$$Real_{\mathcal{A},\tilde{\mathsf{P}}}(\kappa) := \mathbb{P}\left(tr \in Acc(pp,x) \middle| \begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\kappa) \\ (x, aux) \leftarrow \mathcal{A}(pp) \\ tr \leftarrow Tr\big(\langle \tilde{\mathsf{P}}(aux), \mathsf{V}\rangle(pp,x)\big) \end{array} \right),$$

$$Ideal_{\mathcal{A},\tilde{\mathsf{P}}}(\kappa) := \mathbb{P}\left(\begin{array}{c} tr \in Acc(pp,x) \\ \wedge \\ (pp,x,w) \in \mathcal{R}_{ext} \end{array} \middle| \begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\kappa), (x, aux) \leftarrow \mathcal{A}(pp) \\ tr \leftarrow Tr\big(\langle \tilde{\mathsf{P}}(aux), \mathsf{V}\rangle(pp,x)\big) \\ w \leftarrow \mathsf{K}(pp,x,tr) \end{array} \right).$$

Let $Adv_{\mathcal{A},\tilde{\mathsf{P}}}(\kappa) := Real_{\mathcal{A},\tilde{\mathsf{P}}}(\kappa) - Ideal_{\mathcal{A},\tilde{\mathsf{P}}}(\kappa)$. *We say that* Π *is* statistically knowledge sound *with* knowledge error η *if there exists a* K *as above such that for all* $(\mathcal{A}, \tilde{\mathsf{P}})$, $Adv_{\mathcal{A},\tilde{\mathsf{P}}}(\kappa) \leq \eta(\kappa)$. Π *is* computationally knowledge sound *with knowledge error* η *if there exists a* K *as above and a negligible function* negl *such that for every pair* $(\mathcal{A}, \tilde{\mathsf{P}})$ *of PPT algorithms,* $Adv_{\mathcal{A},\tilde{\mathsf{P}}}(\kappa) \leq \eta(\kappa) + \mathsf{negl}(\kappa)$.

Remark 2. When η is negligible, the knowledge soundness definition above implies that for any given adversary $(\tilde{\mathsf{P}}, \mathcal{A})$, either the transcript is accepting with only negligible probability, or K succeeds with extracting a valid witness with overwhelming probability *when given an accepting transcript.*

Definition 10 ([36]). *A proof system* $(\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *for* $\mathcal{R}$ *is (perfectly)* witness indistinguishable (WI) *if for all* $(x,w),(x,w') \in \mathcal{R}$ *and every* $\tilde{\mathsf{V}}$, *the transcripts of* $\langle \mathsf{P}(w), \tilde{\mathsf{V}}\rangle(pp,x)$ *and* $\langle \mathsf{P}(w'), \tilde{\mathsf{V}}\rangle(pp,x)$ *are identically distributed.*

Definition 11. *A public-coin proof system* Π *is a* proof of knowledge (PoK) *for relaxed relation* $\mathcal{R}_{ext}$ *if it is complete for the corresponding relation* $\mathcal{R}$ *and knowledge sound for* $\mathcal{R}_{ext}$ *with negligible knowledge error* η. *If the knowledge soundness is computational, we call* Π *an* argument of knowledge (AoK). *If the system additionally is WI, then we call it a WI-PoK (or WI-AoK).*

Definition 12. *We say that an NP-relation* $\mathcal{R}$ *is* hard *(with respect to* Setup*) if there exists a negligible function* negl *such that for every EPT adversary* $\mathcal{A}$*:*

$$\mathbb{P}\left((pp,x,w) \in \mathcal{R} \middle| \begin{array}{c} pp \leftarrow \mathsf{Setup}(1^\kappa) \\ (x,w) \leftarrow \mathcal{A}(pp) \end{array} \right) \leq \mathsf{negl}(\kappa).$$

The following lemma is, as far as we know, folklore. We prove it in Appendix B.1 for completeness.

Lemma 4. *Let* $\mathcal{R}_{ext}$ *be a relaxed relation of* $\mathcal{R}$ *with its extension being* $\mathcal{R}_{alt}$. *If* Π *is a PoK for* $\mathcal{R}_{ext}$ *and* $\mathcal{R}_{alt}$ *is hard then* Π *is an AoK for* $\mathcal{R}$.

Definition 13 ([41,3]). *For* $\mu \in \mathbb{N}, \mathbf{n} \in \mathbb{N}^\mu$, *let* Π *be a public-coin proof system for relation* $\mathcal{R}$ *in which* μ *challenges are sent and thus transcripts of* Π *have the form* $(pp, x, a_0, c_1, a_1, \ldots, c_\mu, a_\mu)$, *where* c_i *are challenges and* a_i *are messages from the prover. A* $\mathbf{n}$-tree of transcripts, *denoted* τ, *is a set of* $\prod_{i=1}^{\mu} n_i$ *transcripts with the following associated tree structure: The root is* (pp, x, a_0), *each node in layer* i *is labeled by a prover-message* a_i *and each edge between layers* $i-1$ *and* i *is labeled by a challenge* c_i. *A tree* τ *of transcripts is called* valid *if for all* $i \in [\mu]$, *all edges between layers* $i-1$ *and* i *have distinct labels.*

Definition 14 ([27,41]). *Let Π be a public-coin $(2\mu+1)$-round proof system for $\mathcal{R}$ with challenges drawn from sets $Chal_1, \ldots, Chal_\mu$, respectively. For $\mathbf{n} \in \mathbb{N}^\mu$, we say that Π is* $\mathbf{n}$-special-sound *for relaxed relation $\mathcal{R}_{ext}$ of $\mathcal{R}$ if there exists a PPT algorithm* $\mathsf{TreeExtract}$ *which when given a valid $\mathbf{n}$-tree of transcripts for statement x returns a witness w such that $(x, w) \in \mathcal{R}_{ext}$.*

A *Σ-protocol* is a 3-round public-coin 2-special-sound WI proof system [27]. If Π for relation $\mathcal{R}$ is $\mathbf{n}$-special-sound and there exists an EPT algorithm which from a rewindable transcript oracle of Π returns a valid $\mathbf{n}$-tree of transcripts with probability negligibly close to the probability that the transcript oracle returns an accepting transcript, then Π is a PoK for $\mathcal{R}$ [41]. If Π is a public-coin proof system with each challenge drawn from a super-polynomially large set and $\prod_{i=1}^{\mu} n_i$ is polynomial, then such a tree-finding algorithm exists [18,6]. Therefore, special-soundness (for such parameter regimes) implies statistical knowledge soundness as defined above. In this work, we consider relations of two types, both with respect to a commitment scheme CS. With χ being the message space of CS, some $k \in \mathbb{N}$ and a predicate $\phi : \chi^k \to \{0, 1\}$, we have

$$\mathcal{R}_{\phi}^{\mathsf{CS}} := \left\{ \begin{array}{c} \big(pp, (C_1, \ldots, C_k), (m_1, \ldots, m_k, r_1, \ldots, r_k)\big) : \\ \phi(m_1, \ldots, m_k) = 1 \wedge_{i \in [k]} \mathsf{Verify}(pp, C_i, m_i, r_i) = m_i \end{array} \right\}.$$

We relax this relation by the following relation:

$$\mathcal{R}_{bind}^{\mathsf{CS}} := \left\{ \begin{array}{c} \big(pp, (C_1, \ldots, C_k), (m, r, m', r')\big) : \\ \exists i \in [k] s.t. \mathsf{Verify}(pp, C_i, m, r) = m \wedge \mathsf{Verify}(pp, C_i, m', r') = m' \\ \wedge m \neq m' \end{array} \right\}.$$

Under the assumption that CS is computationally binding, $\mathcal{R}_{bind}^{\mathsf{CS}}$ is clearly hard. Thus a public-coin PoK for $\mathcal{R}_{ext}^{\mathsf{CS}} := \{\mathcal{R}_{\phi}^{\mathsf{CS}} \vee \mathcal{R}_{bind}^{\mathsf{CS}}\}$ is a public-coin AoK for $\mathcal{R}_{\phi}^{\mathsf{CS}}$. Pedersen commitments [47] are computationally binding under DLog. PoKs for $\mathcal{R}_{ext}^{\mathsf{CS}}$ with CS being Pedersen commitments and ϕ being that each message is in a given range, or that the third message is the product of the first two, exist in the form of Σ-protocols [27,29,28]. Thus there exists WI-AoKs for those relations $\mathcal{R}_{\phi}^{\mathsf{CS}}$, under DLog. Throughout, we assume that the AoKs and commitment schemes used have the same Setup procedure, that Setup is run before the protocols start and that all subsequent algorithms take pp as input. When CS and the AoKs are instantiated with Pedersen commitments, Setup is as in DLog.

3.3 Certifiable Private Mechanisms and DP

We recall the definitions of CertPM and CertDP from BGKW [12].

Definition 15. *Given a mechanism ensemble $\mathcal{M}_F = \{\mathcal{M}_f : \mathcal{D} \to R\}_{f \in F}$ for a function class F, a* certified probabilistic mechanism *(CertPM) for $\mathcal{M}_F$ is a triple* $(\mathsf{Setup}, \mathsf{P} = (\mathsf{P}^{com}, \mathsf{P}^{open}), \mathsf{V} = (\mathsf{V}^{com}, \mathsf{V}^{open}))$ *where* Setup *is a setup procedure,* P *is a prover and* V *is a verifier such that the following holds for $pp \leftarrow \mathsf{Setup}(1^\kappa)$ and all $f \in F, D \in \mathcal{D}$:*

1. **Correctness:** *The output of* $\langle \mathsf{P}(D), \mathsf{V}\rangle(pp, f)$ *is identically distributed to* $\mathcal{M}_f(D)$.
2. **Cheating-verifier soundness:** *For every cheating verifier* $\tilde{\mathsf{V}}$ *and every pair* $D, D' \in \mathcal{D}$ *and some* $y \in R$,

$$\mathsf{VIEW}^y_{\tilde{\mathsf{V}}}(D) \sim \mathsf{VIEW}^y_{\tilde{\mathsf{V}}}(D').$$

   Here, $\mathsf{VIEW}^y_{\tilde{\mathsf{V}}}(D)$ *denotes the view of* $\tilde{\mathsf{V}}$ *in* $\langle \mathsf{P}(D), \tilde{\mathsf{V}}\rangle(pp, f)$ *conditioned on that* $Q = y$*, with* Q *being the proposed output from* P*. Further, with* $Q_{\neq\perp}$ *denoting* Q *conditioned on* $\tilde{\mathsf{V}}$ *not aborting then* $Q_{\neq\perp} \sim \mathcal{M}_f(D)$.
3. **Cheating-prover soundness:** *There exists a negligible function* negl *such that for any PPT cheating prover* $\tilde{\mathsf{P}}$*, letting* $\gamma_\perp := \mathbb{P}(\langle \tilde{\mathsf{P}}(D), \mathsf{V}\rangle(pp, f) = \perp)$*, the output of* $\langle \tilde{\mathsf{P}}(D), \mathsf{V}\rangle(pp, f)$ *has statistical distance at most* $\gamma_\perp + \mathsf{negl}(\kappa)$ *to* $\mathcal{M}_f(D)$.

Definition 16. *A triple* $(\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *is a* certified (α, β)-accurate (ε, δ)-DP *($(\alpha, \beta, \varepsilon, \delta)$-CertDP) protocol for a function class* F *if there exists a negligible function* negl *such that for all* $D \in \mathcal{D}, f \in F$ *when* $pp \leftarrow \mathsf{Setup}(1^\kappa)$*, the following holds.*

1. **Correctness:** $\mathcal{M}_f(\cdot) \leftarrow \langle \mathsf{P}(\cdot), \mathsf{V}\rangle(pp, f)$ *is* $(\alpha, \beta + \mathsf{negl}(\kappa))$*-accurate,* $(\varepsilon, \delta + \mathsf{negl}(\kappa))$*-DP and* $\mathbb{P}(\mathcal{M}_f(D) = \perp) = 0$.
2. **Honest-curator DP:** *For any cheating verifier* $\tilde{\mathsf{V}}$*,* $\mathcal{M}'_f(\cdot) \leftarrow \langle \mathsf{P}(\cdot), \tilde{\mathsf{V}}\rangle(pp, f)$ *is* $(\varepsilon, \delta + \mathsf{negl})$*-DP.*
3. **Dishonest-curator DP:** *Let for any cheating prover* $\tilde{\mathsf{P}}$ *and any* $D' \in \mathcal{D}$*,*

$$\mathcal{M}_f(D') \leftarrow \begin{cases} \langle \tilde{\mathsf{P}}(D'), \mathsf{V}\rangle(pp, f), \text{ if } D' = D, \\ \langle \mathsf{P}(D'), \mathsf{V}\rangle(pp, f), \text{ otherwise.} \end{cases}$$

   If $\tilde{\mathsf{P}}$ *is PPT then for any constant* $\gamma \geq 0$*, if* $\mathcal{M}_f$ *fails to be either* $(\alpha, \beta + \gamma + \mathsf{negl}(\kappa))$*-accurate or* $(\varepsilon, \delta + \gamma + \mathsf{negl}(\kappa))$*-DP, then* $\mathbb{P}(\mathcal{M}_f(D') = \perp) \geq c \cdot \gamma - \mathsf{negl}(\kappa)$ *for some constant* $c \in (0, 1]$.

4 Relaxing the RVCS Definition to Align with Special-Soundness

We now propose an adaptation of the RVCS definition and prove that it is fulfilled by existing constructions, whenever the AoK fulfills Definition 9.

Definition 17 (RVCS, adapted). *A* random variable commitment scheme *(RVCS) for distribution* Z *is defined as in Definition 1 but with the cheating-dealer distributional soundness property replaced by:*

There exists a negligible function negl *and an EPT* $\mathsf{RevealOpening}$ *algorithm, with rewindable black-box access to the cheating dealer* $\tilde{\mathsf{DE}}$*, such that for every PPT* $\tilde{\mathsf{DE}}$*, the following holds. Let* $C^{out} \leftarrow \langle \tilde{\mathsf{DE}}, \mathsf{PL}\rangle(pp)$*,* $tr \leftarrow Tr\big(\langle \tilde{\mathsf{DE}}, \mathsf{PL}\rangle(pp)\big)$ *and let* $(r^Q, z^Q) \leftarrow \mathsf{RevealOpening}(pp, tr)$*. Let* $z^{out} \leftarrow \mathsf{Verify}(pp, C^{out}, r^Q, z^Q)$ *and* $\gamma_\perp = \mathbb{P}(C^{out} = \perp)$*. Then* $z^{out} \approx^s_{\gamma_\perp + \mathsf{negl}(\kappa)} Z$.

Remark 3. The changes we make are that RevealOpening is EPT and is given the transcript rather than the view of the cheating dealer.

4.1 The RVCS for Fair Coins

We now show that the RVCS construction in BGKW for fair coins, CoinCS, fulfills our adapted RVCS definition if the AoK satisfies the formalization in Section 3.2. Let WI-AoK$_{bin}$ be for relation $\mathcal{R}_{\phi}^{\mathsf{CS}}$ with $\phi(m) := \mathbf{1}_{\{0,1\}}(m)$. $\oplus$ denotes XOR, so $\hat{\oplus}$ (and $\check{\oplus}$) is the homomorphic XOR [12]. We recall the CoinCS construction.

Definition 18. *Let* $\mathsf{CS} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ *be an additive commitment scheme on* $\mathbb{Z}_q$*. We define* $\mathsf{CoinCS} := (\mathsf{CS}, \mathsf{DE}, \mathsf{PL})$ *as:*

1. DE *samples* $b_{\mathsf{DE}} \leftarrow Ber(1/2)$*, sets* $(C_{b_{\mathsf{DE}}}, r_{b_{\mathsf{DE}}}) \leftarrow \mathsf{Commit}(b_{\mathsf{DE}})$ *and sends* $C_{b_{\mathsf{DE}}}$ *to* PL.
2. DE *and* PL *together execute* WI-AoK$_{bin}$ *on* $C_{b_{\mathsf{DE}}}$*. If* PL *reject, they abort.*
3. PL *samples* $b_{\mathsf{PL}} \leftarrow Ber(1/2)$ *and sends it to* DE.
4. DE *computes* $C_{b_{\mathsf{DE}}+b_{\mathsf{PL}}} \leftarrow b_{\mathsf{PL}} \hat{\oplus} C_{b_{\mathsf{DE}}}$ *and sends it to* PL.
5. PL *computes* $C'_{b_{\mathsf{DE}}+b_{\mathsf{PL}}} \leftarrow b_{\mathsf{PL}} \hat{\oplus} C_{b_{\mathsf{DE}}}$ *and accepts if* $C'_{b_{\mathsf{DE}}+b_{\mathsf{PL}}} = C_{b_{\mathsf{DE}}+b_{\mathsf{PL}}}$.

Theorem 1. *If* WI-AoK$_{bin}$ *is perfectly complete, perfectly WI and computationally knowledge sound, then* CoinCS *is an RVCS (Definition 17) for* $Ber(1/2)$.

Proof. The only property affected by our adaptation is cheating-dealer distributional soundness, so we show only that.[13]

Cheating-dealer distributional soundness: We construct RevealOpening as follows: It is given a transcript tr of the interaction and if PL aborted before or during the AoK, then RevealOpening aborts and we define $b' = \perp$. Otherwise RevealOpening runs the knowledge extractor K (as given by Definition 9) on the part of tr corresponding to the AoK. K extracts a valid witness $b_{\tilde{\mathsf{DE}}}, r_{\tilde{\mathsf{DE}}}$ with overwhelming probability (since the AoK transcript is accepting). Then define $b' \leftarrow b_{\tilde{\mathsf{DE}}} \oplus b_{\mathsf{PL}}$.[14] Note that b_{PL} is explicitly part of tr and that PL will always send some b_{PL} if it accepts the AoK. Now, if $C^{out} = \perp$ (i.e. if $\tilde{\mathsf{DE}}$ caused an abort after receiving b_{PL}) then RevealOpening aborts. Otherwise RevealOpening outputs b' and $r' \leftarrow r_{\tilde{\mathsf{DE}}} \check{\oplus} b_{\mathsf{PL}}$.

Let (b^Q, r^Q) denote the output of $\mathsf{RevealOpening}(pp, tr)$, which is either (b', r') or $(\perp, \perp)$. We have $z^{out} \leftarrow \mathsf{Verify}(pp, C^{out}, r^Q, b^Q)$. We want to prove that $SD(z^{out}, Z) \leq \gamma_{\perp} + \mathsf{negl}(\kappa)$, with Z being a fair coin toss and negl being some negligible function. First we note that if $\gamma_{\perp}$ is overwhelming then it holds directly, so we turn to the other case. We will show this by using the triangle inequality for statistical distances (Proposition 1), as follows:

$$SD(z^{out}, Z) \leq SD(z^{out}, b') + SD(b', Z).$$

[13] For the other two properties, see BGKW [12]. We note that the three properties in the RVCS definition are proven independently of each other and each of the subproofs use different properties of the AoK, with knowledge soundness being used only to prove cheating-dealer distributional soundness.

[14] Set $\perp \oplus b = \perp$ for any b. Thus, b' is $\perp$ with probability equal to that of PL aborting before it has sent b_{PL} plus some negligible value.

For $SD(b', Z)$, if $b' \neq \perp$ then b' is a uniformly random bit with overwhelming probability due to the homomorphic property, since b_{PL} is uniform and K succeeds with overwhelming probability when given an accepting AoK transcript. Therefore $SD(b', Z) = \gamma_{\perp}^{early} + \mathsf{negl}'(\kappa)$, for some negl', where $\gamma_{\perp}^{early}$ denotes the probability that PL aborts before or during the AoK.

For $SD(z^{out}, b')$, there are two possible sources for distance; when $b' \neq \perp = C^{out}$ and when neither b' or C^{out} are $\perp$ but $z^{out} \neq b'$. The first case is precisely the probability that PL aborts but only after having sent b_{PL}. Let us call this probability $\gamma_{\perp}^{late}$ and note that, clearly, $\gamma_{\perp} = \gamma_{\perp}^{early} + \gamma_{\perp}^{late}$. In the second case, we know that $C^{out} = C_{\tilde{\mathsf{DE}+\mathsf{PL}}}$, since otherwise PL would have aborted. Now since K succeeds in its extraction with overwhelming probability, the homomorphism and binding of CS implies that with overwhelming probability, r' opens C^{out} to b'. Thus the statistical distance between b' and z^{out} in the case that no aborts occur is negligible. Therefore $SD(z^{out}, b') = \gamma_{\perp}^{late} + \mathsf{negl}''(\kappa)$, and finally

$$SD(z^{out}, Z) \leq \gamma_{\perp}^{early} + \mathsf{negl}'(\kappa) + \gamma_{\perp}^{late} + \mathsf{negl}''(\kappa) = \gamma_{\perp} + \mathsf{negl}(\kappa).$$

□

There exists Σ-protocols for binary opening of Pedersen commitments under DLog [5,28]. Therefore, CoinCS can be instantiated with Σ-protocols and Pedersen commitments to fulfill Definition 17.

4.2 CertPM from Adapted RVCS Definition

We now show that our version of RVCS implies CertPM by use of the same construction as the original RVCS definition.

Definition 19 (CertPM construction, BGKW [12]). *Let* CS *be an additive* f*-homomorphic commitment scheme for all* f *in a family* F*. Let* RVCS *be an RVCS for distribution* Z*. Define* $\big(\mathsf{Setup}, (\mathsf{P}^{com}(D), \mathsf{P}^{open}), (\mathsf{V}^{com}, \mathsf{V}^{open})\big)$ *as follows.*

1. ***Commitment phase:*** $\mathsf{P}^{com}(D)$ *computes* $(C_D, r_D) \leftarrow \mathsf{Commit}(pp, D)$ *and sends* C_D *to* V^{com}*.*
2. ***Randomness phase:*** P^{com} *and* V^{com} *together run* RVCS*, from which* P^{com} *gets* (z, r_z) *and both get* C_z*.*
3. ***Querying phase:*** $\mathsf{P}^{open}(D, r_D, r_z)$ *computes* $r_Q \leftarrow \check{f}(r_D) \check{+} r_z$ *and sends* $Q \leftarrow f(D) + z$ *and* r_Q *to* V^{open}*.* $\mathsf{V}^{open}(C_D, C_z)$ *computes* $C^Q \leftarrow \hat{f}(C_D) \hat{+} C_z$ *and outputs* $\mathsf{Verify}(pp, C^Q, r_Q, Q)$*.*

Below we assume, as in BGKW, that the commitment to the database D is honestly generated, i.e. that even dishonest provers can open C_D to some well-defined D with probability 1. For a discussion of how to handle dishonest database commitments, see Appendix A in BGKW.

Theorem 2. *The CertPM construction (Definition 19) instantiated with an RVCS (Definition 17) for distribution* Z *is a CertPM for ensemble* $\{\mathcal{M}_f(D) = f(D) + Z\}_{f \in F}$*.*

Proof. Cheating-dealer distributional soundness affects only the cheating-prover soundness of the CertPM construction.

Cheating-prover soundness: Let $\mathsf{RevealOpening}(pp, tr)$ output (r^Q, z^Q) and set $z^{out} \leftarrow \mathsf{Verify}(pp, C_z, r^Q, z^Q)$. By cheating-player distributional soundness, we have that $SD(z^{out}, Z) \leq \gamma_\perp^{RVCS} + \mathsf{negl}'(\kappa)$, where $\gamma_\perp^{RVCS} = \mathbb{P}(C_z = \perp)$. We must show that $SD(Q^{out}, \mathcal{M}_f(D)) \leq \gamma_\perp + \mathsf{negl}$, for some negl, where $Q^{out} \leftarrow \langle \tilde{\mathsf{P}}(D, r_D), \mathsf{V} \rangle (C_D, f)$ and $\gamma_\perp = \mathbb{P}(Q^{out} = \perp)$. We have two cases. If $\gamma_\perp$ is overwhelming, the bound on $SD(Q^{out}, \mathcal{M}_f(D))$ is directly satisfied, since a statistical distance cannot be larger than 1.

If $\gamma_\perp$ is not overwhelming, then this in particular means that V accepts the RVCS with non-negligible probability, which in turn implies that $z^{out} = z^Q$ with non-negligible probability. Further, by the assumption of an honest database commitment, a valid opening of C_D to D can be efficiently extracted from $\tilde{\mathsf{P}}$ with probability 1, and thus C^Q can be opened to only $f(D) + z^Q$, except with negligible probability, by homomorphism and binding.

Therefore, with overwhelming probability, Q^{out} will be either $f(D) + z^{out}$ or $\perp$. Let $\gamma_\perp^{other}$ denote the probability that V aborts after the RVCS and note that by assumption, $\tilde{\mathsf{P}}$ will never cause an abort before the randomness phase. Clearly $\gamma_\perp = \gamma_\perp^{RVCS} + \gamma_\perp^{other}$ and[15] $SD(Q^{out}, f(D) + z^{out}) = \gamma_\perp^{other} + \mathsf{negl}''(\kappa)$. Therefore, by the triangle inequality (Proposition 1),

$$\begin{aligned} &SD(Q^{out}, \mathcal{M}_f(D)) \leq SD(Q^{out}, f(D) + z^{out}) + SD(f(D) + z^{out}, \mathcal{M}_f(D)) \\ &\leq \gamma_\perp^{other} + \mathsf{negl}''(\kappa) + \gamma_\perp^{RVCS} + \mathsf{negl}'(\kappa) = \gamma_\perp + \mathsf{negl}(\kappa). \end{aligned}$$

□

5 Modularity Lemmata

We present our lemmata about Definition 17. They hold also for the original RVCS definition, when using knowledge soundness as in Definition 2.

5.1 Sequential Composition

In Lemma 5 below, we restrict ourselves to a polynomial number of compositions. The reason for this is that the proof technique requires the cheating party to emulate some of the subprotocols internally and then uses this to attack one specific subprotocol. Since the cheating dealer is PPT, we cannot have more than polynomially many subprotocols. As before, C^{out} denotes the output of the protocol and in the case of this composed protocol, for an honest player we either have $C^{out} = (C_1^Q, \ldots, C_k^Q)$ with C_i^Q being the proposed commitment from the dealer in RVCS_i or we have $C^{out} = \perp$. The honest player in the composed

[15] If $z^{out} = \perp$ then necessarily $Q^{out} = f(D) + z^{out} = \perp$. So the distance between Q^{out} and $f(D) + z^{out}$ is $\mathbb{P}(z^{out} \neq \perp \wedge Q^{out} = \perp)$, which is $\gamma_\perp^{other}$, plus $\mathbb{P}(\perp \neq f(D) + z^{out} \neq Q^{out} \neq \perp)$, which is negligible.

protocol will output $C^{out} = \perp$ if $C_i^{out} = \perp$ for any $i \in [k]$. Otherwise PL outputs $C^{out} = (C_1^Q, \ldots, C_k^Q)$ and, similarly, we have $\mathbf{r}^Q = (r_1^Q, \ldots, r_k^Q)$. Thus in the composed protocol, we have $\mathbb{P}(C^{out} = \perp) = \sum_{i \in [k]} \mathbb{P}(C_i^{out} = \perp)$.

Lemma 5. *Let $\mathsf{RVCS}_1, \ldots \mathsf{RVCS}_k$ be polynomially many RVCS's for distributions $Z_1, \ldots, Z_k$, respectively. The protocol where $\mathsf{RVCS}_1, \ldots \mathsf{RVCS}_k$ are run sequentially is an RVCS for the joint probability distribution $Z_1 \times \ldots \times Z_k$.*

Proof. We prove the three properties separately.

Correctness: This follows directly from the correctness of the individual RVCS's. That is, for each $i \in [k]$, we have $(C_i^Q, r_i^Q) \sim \mathsf{Commit}(z_i)$ with $z_i \leftarrow Z_i$ and thus $((C_1^Q, \ldots, C_k^Q), \mathbf{r}^Q) \sim \mathsf{Commit}(pp, \mathbf{z})$ with $\mathbf{z} \leftarrow Z_1 \times \ldots \times Z_k$.

Cheating-player distributional soundness: Assume towards a contradiction that a cheating player $\tilde{\mathsf{PL}}$ can break the cheating-player distributional soundness of the composed protocol and let $\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<i}} := (\mathsf{VIEW}_{\tilde{\mathsf{PL}}_1}, \ldots \mathsf{VIEW}_{\tilde{\mathsf{PL}}_{i-1}})$ be the view of $\tilde{\mathsf{PL}}$ up until the i'th protocol. That is, we assume that $\tilde{\mathsf{PL}} = (\tilde{\mathsf{PL}}_1, \tilde{\mathsf{PL}}_2(\mathsf{VIEW}_{\tilde{\mathsf{PL}}_1}), \ldots, \tilde{\mathsf{PL}}_k(\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<k}}))$ for the composed protocol is such that $\mathbf{z}^{out} := \mathbf{z}^{out}_{C \neq \perp}$ is not distributed according to $Z_1 \times \ldots \times Z_k$. Since z_1^{out} not having distribution Z_1 directly contradicts the cheating-player distributional soundness of RVCS_1, we let $i > 1$ be the first index such that z_i^{out} is not distributed according to Z_i.

Consider the following cheating player strategy $\tilde{\mathsf{PL}}^*$ for RVCS_i: $\tilde{\mathsf{PL}}^*$ internally runs a copy of $(\mathsf{DE}_1, \tilde{\mathsf{PL}}_1), \ldots, (\mathsf{DE}_{i-1}, \tilde{\mathsf{PL}}_{i-1}(\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<i-1}}))$ (running both parties[16]) and gets a simulated view $\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<i}}$. Then $\tilde{\mathsf{PL}}^*$ runs the interaction between DE_i and $\tilde{\mathsf{PL}}_i(\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<i}})$. The simulated view $\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<i}}$ from the internal executions is identically distributed to the view that $\tilde{\mathsf{PL}}$ gets from interacting with $\mathsf{DE}_1, \ldots, \mathsf{DE}_{i-1}$ in the (real) composed protocol. Therefore, the assumption that $\tilde{\mathsf{PL}}$ breaks cheating-player distributional soundness of the composed protocol implies that $\tilde{\mathsf{PL}}^*$ breaks the same property of RVCS_i, which is a contradiction.

Cheating-prover distributional soundness: Assume towards a contradiction that $\tilde{\mathsf{DE}} = (\tilde{\mathsf{DE}}_1, \ldots, \tilde{\mathsf{DE}}_k(\mathsf{VIEW}_{\tilde{\mathsf{DE}}_{<k}}))$ is a PPT cheating dealer of the composed protocol that breaks cheating-dealer distributional soundness. That is, $\tilde{\mathsf{DE}}$ is such that all EPT $\mathsf{RevealOpening}$ return $\mathbf{r}^Q, \mathbf{z}^Q$ such that $\mathbf{z}^{out} \leftarrow \mathsf{Verify}(C^{out}, \mathbf{r}^Q, \mathbf{z}^Q)$ has $SD(\mathbf{z}^{out}, Z_1 \times \ldots \times Z_k)$ non-negligibly above $\mathbb{P}(C^{out} = \perp)$. That is, $\mathbb{P}(C^{out} = \perp) \leq SD(\mathbf{z}^{out}, Z_1 \times \ldots \times Z_k) + \nu(\kappa)$ for some non-negligible function ν. Let $\gamma_\perp^1, \ldots, \gamma_\perp^k$ denote the probabilities that the abort occurs in $\mathsf{RVCS}_1, \ldots, \mathsf{RVCS}_k$, respectively (conditional on the abort not occurring earlier). By Proposition 1,

$$\gamma_\perp = \sum_{i \in [k]} \gamma_\perp^i \leq \nu(\kappa) + SD(\mathbf{z}^{out}, Z_1 \times \ldots \times Z_k) \leq \nu(\kappa) + \sum_{i \in [k]} SD(z_i, Z_i).$$

So for at least one protocol RVCS_i we have $\gamma_\perp^i \leq SD(z_i^{out}, Z_i) + \nu'(\kappa)$ with ν' non-negligible (since k is polynomial). Let RVCS_i be the first RVCS for which

[16] Note that the description of DE is public and DE has no private input.

this holds. Define $\tilde{\mathsf{DE}}^*$ for RVCS_i by that it runs a copy of $(\tilde{\mathsf{DE}}_1, \mathsf{PL}_1), \ldots, (\tilde{\mathsf{DE}}_{i-1}(\mathsf{VIEW}_{\tilde{\mathsf{DE}}_{<i-1}}), \mathsf{PL}_{i-1})$ internally and gets a simulated view $\mathsf{VIEW}_{\tilde{\mathsf{DE}}_{<i}}$. Then in a real execution of RVCS_i, $\tilde{\mathsf{DE}}^*$ acts like $\tilde{\mathsf{DE}}_i(\mathsf{VIEW}_{\tilde{\mathsf{DE}}_{<i}})$. The view of $\tilde{\mathsf{DE}}^*$ in the internal execution of $\mathsf{RVCS}_1, \ldots, \mathsf{RVCS}_{i-1}$ is distributed identically to the view of $\tilde{\mathsf{DE}}$ at this point in the composed protocol. Thus every EPT $\mathsf{RevealOpening}$ procedure given the transcript of the interaction with PL_i returns r^Q, z^Q such that $z_i^{out} \leftarrow \mathsf{Verify}(C^{out}, r^Q, z^Q)$ has $SD(z_i^{out}, Z_i)$ non-negligibly larger than $\gamma_\perp^i$. This contradicts the cheating-prover distributional soundness of RVCS_i. □

5.2 Preservation under Homomorphic Post-processing

We now show that one can easily transform any RVCS for a distribution Z to a distribution $f(Z)$ for any function f if CS is f-homomorphic. The player simply needs to homomorphically apply the function f to the commitment it gets from the RVCS for Z. We call this *homomorphic post-processing*.

Definition 20. *Let* CS *be an* f*-homomorphic commitment scheme for some* f *and let* RVCS_{base} *be an RVCS for a distribution* Z*. We say that* $\mathsf{RVCS}_{post} := (\mathsf{CS}, \mathsf{DE}_{post}, \mathsf{PL}_{post})$ *is a* homomorphic post-processing *of* RVCS_{base} *with respect to* f *if* $\mathsf{DE}_{post}, \mathsf{PL}_{post}$ *proceed as below.*

1. DE_{post} *and* PL_{post} *together run* RVCS_{base}*, both receive* C_{base}^{out} *and* DE_{post} *also gets* r_{base} *and* z_{base}*. If* $C_{base}^{out} = \perp$ *then* PL_{post} *aborts.*
2. DE_{post} *computes* $r_{post} \leftarrow \check{f}(r_{base})$*,* $C_{post}^{out} \leftarrow \hat{f}(C_{base}^{out})$*, and* $z_{post} \leftarrow f(z_{base})$*.* PL_{post} *computes* $C_{post}^{out} \leftarrow \hat{f}(C_{base}^{out})$ *and outputs it.*

Lemma 6. *The homomorphic post-processing of any RVCS for distribution* Z *with respect to any* f *is an RVCS for the distribution* $f(Z)$*.*

In the proof below (and that of the next lemma), we rely on the observation [12] that cheating-player distributional soundness follows if one can show that the view of the cheating player is independent of the sampled value and that the player can only influence the sample by aborting.

Proof. We prove the three properties separately.

Correctness: In an honest execution, none of the parties abort and thus $r_{z_{base}}$ indeed opens C_{base}^{out} to z_{base} and the correctness of RVCS_{base} gives that z_{base} is distributed according to Z. Due to the homomorphic property of CS, we have $\mathsf{Verify}\big(pp, \hat{f}(C_{z_{base}}), \check{f}(r_{base}), f(z_{base})\big) = f(z_{base})$, and $f(z_{base})$ clearly has distribution $f(Z)$.

Cheating-player distributional soundness: Denote the cheating player by $\tilde{\mathsf{PL}}_{post}$ and its actions in step 1 by $\tilde{\mathsf{PL}}_{base}$. Since RVCS_{base} is an RVCS for Z, unless if $\tilde{\mathsf{PL}}_{base}$ causes an abort, z_{base} is distributed according to Z, C_{base}^{out} is a correct commitment to z_{base} and r_{base} is the corresponding opening information. Due to the cheating-player distributional soundness of RVCS_{base}, the distribution

of the view of $\tilde{\mathsf{PL}}_{base}$ is independent of the value of z_{base} and since in step 2, no new messages are sent between the parties, the distribution of $\tilde{\mathsf{PL}}_{post}$'s view is independent of z_{post}. Also, the lack of interaction implies that the only way for $\tilde{\mathsf{PL}}_{post}$ to change the output distribution is to abort selectively. Thus, that $\tilde{\mathsf{PL}}_{post}$'s choice whether or not to abort must be independent of z_{post} implies cheating-player distributional soundness.

Cheating-prover distributional soundness: Denote the cheating dealer by $\tilde{\mathsf{DE}}_{post}$ and its actions in step 1 by $\tilde{\mathsf{DE}}_{base}$. Let $\gamma_{\perp}^{base}$ be $\mathbb{P}(C_{base}^{out} = \perp)$. We now extend the $\mathsf{RevealOpening}$ procedure for RVCS_{base}. We have given that there exists an EPT $\mathsf{RevealOpening}_{base}$ algorithm which on rewindable black-box access to the $\tilde{\mathsf{DE}}_{base}$ in step 1 returns opening information r_{base}^{Q} and z_{base}^{Q} such that $z_{base}^{out} \leftarrow \mathsf{Verify}(pp, C_{base}^{out}, r_{base}^{Q}, z_{base}^{Q})$ has statistical distance at most $\gamma_{\perp}^{base} + \mathsf{negl}(\kappa)$ to Z. Now we can set $\mathsf{RevealOpening}_{post}$ to run $\mathsf{RevealOpening}_{base}$ on the transcript of RVCS_{base} and output $\check{f}(r_{base}^{Q})$ and $z_{post}^{out} \leftarrow f(z_{base}^{Q})$. Since no messages are exchanged between the parties in step 2, the honest player will never abort during step 2 so the total abort probability $\gamma_{\perp}$ is equal to $\gamma_{\perp}^{base}$. Thus $C_{post}^{out} = \hat{f}(C_{base}^{out})$ and therefore, we get (via Proposition 1)

$$z_{post}^{out} := \mathsf{Verify}(pp, C_{post}^{out}, \check{f}(r_{base}^{Q}), f(z_{base}^{Q})) = f(z_{base}^{out}) \approx_{\gamma_{\perp}+\mathsf{negl}(\kappa)}^{s} f(Z).$$

□

5.3 Preservation under Commit-and-Prove Post-processing

We show that if one feeds RVCS outputs (for distributions $Z_1, \ldots, Z_k$) into a "Commit-and-Prove"-style [4,49] WI-AoK for some function f then the resulting protocol is an RVCS for the distribution $f(Z_1 \times \ldots \times Z_k)$. For a function $f : \chi^k \to \chi$, let $\mathcal{R}_f^{\mathsf{CS}}$ be $\mathcal{R}_\phi^{\mathsf{CS}}$ with $\phi((\mathbf{z}, y)) = \mathbf{1}_{\{\mathbf{m} \in \chi^{k+1} : m_{k+1} = f(m_1, \ldots, m_k)\}}(\mathbf{z}, y)$.

Definition 21. *Let* $\mathsf{RVCS}_1, \ldots, \mathsf{RVCS}_k$ *be RVCS's for distributions* $Z_1, \ldots, Z_k$ *over* χ*, respectively,* f *be a function from* χ^k *to* χ*, and let* $\mathsf{WI\text{-}AoK}_f$ *be a WI-AoK for* $\mathcal{R}_f^{\mathsf{CS}}$*. The* CnP post-processing *of* $\mathsf{RVCS}_1, \ldots, \mathsf{RVCS}_k$ *with respect to* f *proceeds as follows:*

1. DE *and* PL *together run* $\mathsf{RVCS}_1, \ldots, \mathsf{RVCS}_k$ *sequentially. The outputs are* $C_1^{out}, \ldots, C_k^{out}$ *and* DE *holds the corresponding samples* $\mathbf{z} = (z_1, \ldots, z_k)$ *and opening information* $\mathbf{r} = (r_1, \ldots, r_k)$*.*
2. DE *computes* $y = f(\mathbf{z})$*, commits to* y *in* C_{post}^{out} *and sends* C_{post}^{out} *to* PL*.*
3. DE *and* PL *run* $\mathsf{WI\text{-}AoK}_f$ *on* $(C_1^{out}, \ldots, C_k^{out}, C_{post}^{out})$ *and if* PL *accepts then it outputs* C_{post}^{out}*, else* $\perp$*.*

Lemma 7. *Let* $\mathsf{RVCS}_1, \ldots, \mathsf{RVCS}_k$ *be polynomially many RVCS's for distributions* $Z_1, \ldots, Z_k$ *over* χ*, respectively. Let* $f : \chi^k \to \chi$ *be a function which is computable in polynomial time, and let* $\mathsf{WI\text{-}AoK}_f$ *be a WI-AoK for* $\mathcal{R}_f^{\mathsf{CS}}$*. Then the CnP post-processing of* $\mathsf{RVCS}_1, \ldots, \mathsf{RVCS}_k$ *with respect to* f *is an RVCS for* $f(Z_1 \times \ldots \times Z_k)$*.*

Proof. **Correctness:** In an honest execution, due to Lemma 5, the sample $\mathbf{z}$ has distribution $Z_1 \times \ldots \times Z_k$ and DE can open C^{out}_{post} to $y = f(\mathbf{z})$. Since $\mathsf{WI\text{-}AoK}_f$ is complete, PL always accepts in an honest execution.

Cheating-player distributional soundness: $\tilde{\mathsf{PL}}$ only sends messages during the executions of the RVCS's and during the AoK. Due to the cheating-player distributional soundness of the RVCS's, unless $\tilde{\mathsf{PL}}$ causes an abort, $\mathbf{z}$ will have distribution $Z_1 \times \ldots \times Z_k$. The interaction during the AoK occurs after DE has proposed an output commitment and thus no message from $\tilde{\mathsf{PL}}$ during the AoK can change the values which DE can open C^{out}_{post} to. Therefore, $\tilde{\mathsf{PL}}$ may only influence the distribution of those openings by aborting. The distribution of the view of $\tilde{\mathsf{PL}}$ is independent of y. Independence in step 1 is guaranteed by the cheating-player distributional soundness of the RVCS's, independence in the second step follows from the perfect hiding of CS and independence in the final step follows from $\mathsf{WI\text{-}AoK}_f$ being perfectly WI. Thus $\tilde{\mathsf{PL}}$ must make the choice of whether to abort independently of y so if $\tilde{\mathsf{PL}}$ does not abort then z^{out} is distributed according to $f(Z_1 \times \ldots \times Z_k)$.

Cheating-prover distributional soundness: By Lemma 5, step 1 is an RVCS for $Z_1 \times \ldots \times Z_k$. We construct $\mathsf{RevealOpening}$ as follows. It gets the transcript tr and if PL aborted in step 1 (i.e. if any of $C^{out}_1, \ldots, C^{out}_k$ is $\perp$) then $\mathsf{RevealOpening}$ aborts and we define $y = r_{post} = \perp$. Otherwise it runs the reveal procedure $\mathsf{RevealOpening}_{base}$ of step 1 (on the corresponding part of tr) and gets $\mathbf{z}$ and $\mathbf{r}$. Define $y = f(\mathbf{z})$ and $r_{post} = \check{f}(\mathbf{r})$. Now if PL aborted in the post-processing, then $\mathsf{RevealOpening}$ aborts and outputs $(\perp, \perp)$. Otherwise $\mathsf{RevealOpening}$ outputs (y, r_{post}).

Let (y^*, r^*_{post}) be the output of $\mathsf{RevealOpening}$ (which is either (y, r_{post}) or $(\perp, \perp)$). Set $z^{out} \leftarrow \mathsf{Verify}(pp, C^{out}_{post}, r^*_{post}, y^*)$. If PL aborts with overwhelming probability then $SD(z^{out}, Z) \leq 1 = \gamma_\perp + \mathsf{negl}$ so we assume that PL accepts with non-negligible probability. We use the triangle inequality (Proposition 1) as

$$SD(z^{out}, Z) \leq SD(z^{out}, y) + SD(y, Z).$$

Let $\gamma^{base}_\perp$ denote the probability of abort in step 1 and $\gamma^{post}_\perp$ denote the probability of abort in during the post-processing (conditional on no abort in step 1). Clearly, $\gamma_\perp = \gamma^{base}_\perp + \gamma^{post}_\perp$. Since step 1 is an RVCS for $Z_1 \times \ldots \times Z_k$, we get $SD(\mathbf{z}, Z_1 \times \ldots \times Z_k) \leq \gamma^{base}_\perp + \mathsf{negl}'(\kappa)$. Thus Proposition 1 implies that $SD(y, Z) = SD(f(\mathbf{z}), f(Z_1 \times \ldots \times Z_k)) \leq \gamma^{base}_\perp + \mathsf{negl}'(\kappa)$.

If $y = \perp$ (PL aborted in step 1) then necessarily $C^{out}_{post} = y^* = \perp$, so we turn to the other case. There are three possibilities:

1. $z^{out} = y^* = \perp \neq y$, i.e. PL rejects the AoK.
2. $z^{out} = y^* = y \neq \perp$, i.e. PL accepts the AoK (giving $y^* = y$) and r^*_{post} is valid opening information for C^{out}_{post}.
3. $z^{out} = \perp \neq y^* = y$, i.e. PL accepts the AoK but r^*_{post} is not valid opening information for C^{out}_{post}.

The probability of the first is exactly $\gamma^{post}_\perp$. As the probability that PL accepts the AoK is non-negligible, the knowledge extractor K of the AoK succeeds with non-negligible probability to extract $\big((\mathbf{z}', y'), (\mathbf{r}', r'_{post})\big)$ for $(C^{out}_1, \ldots, C^{out}_k, C^{out}_{post})$

such that $y' = f(\mathbf{z}')$. So we now have two efficient ways of finding openings of $(C_1^{out}, \ldots, C_k^{out})$ (one by $\mathsf{RevealOpening}_{base}$ and one by K), both of which work with non-negligible probability. Therefore the binding property of CS implies that with overwhelming probability, $\mathbf{z}' = \mathbf{z}$, and thus $y = y'$ (since CS is homomorphic). Therefore if the AoK is accepted then with overwhelming probability, r_{post} opens C_{post}^{out} to y, so $SD(z^{out}, y) = \gamma_\perp^{post} + \mathsf{negl}''(\kappa)$, which gives

$$SD(z^{out}, Z) \leq \gamma_\perp^{post} + \mathsf{negl}''(\kappa) + \gamma_\perp^{base} + \mathsf{negl}'(\kappa) = \gamma_\perp + \mathsf{negl}(\kappa).$$

□

The lemmata together with CoinCS imply the following corollary, as there are CnP WI-AoK's (as Σ-protocols) for binary openings and any arithmetic circuit over $\mathbb{F}_q$, using Pedersen commitments [28].

Corollary 1. *Under the discrete logarithm assumption, there exists an RVCS for any distribution over $\mathbb{Z}_q$ which can be sampled by an arithmetic circuit taking polynomially many uniform bits as input.*

Let U_k denote the uniform distribution of k-bit strings.

Proof. Under DLog, there exists CnP WI-AoK's (in the form of Σ-protocols) for binary openings and any arithmetic circuit over $\mathbb{F}_q$, using Pedersen commitments [28]. Similarly, CoinCS is an RVCS for fair coins, under DLog, when instantiated with Σ-protocols and Pedersen commitments. Let k be polynomial and let $f : \mathbb{Z}_q^k \to \mathbb{Z}_q$ be a function computable by an arithmetic circuit over $\mathbb{F}_q$. Build an RVCS as follows: First run CoinCS k times in sequence. By Lemma 5, this is an RVCS for U_k. Then perform the CnP post-processing of the output from the first step, with respect to the function f. By Lemma 7, this full protocol is an RVCS for the distribution $f(U_k)$, as desired. □

6 Relaxing the Definitions to Allow Honest Aborts

To allow for sampling algorithms that fail/abort with negligible probability, we now propose weakened definitions of RVCS, CertPM and CertDP. We show that these follow from each other in the same way as for the original versions.

Definition 22. *A* statistical random variable commitment scheme *(sRVCS) for distribution Z is defined as in Definition 17 but with the cheating-player distributional soundness property replaced by: There exists a negligible function* negl *such that for every cheating player* $\tilde{\mathsf{PL}}$*, letting* $(C^{out}, (r^Q, z^Q)_{\mathsf{DE}}) \leftarrow \langle \mathsf{DE}, \tilde{\mathsf{PL}} \rangle(pp)$ *and* $z_{C\neq\perp}^{out} \leftarrow \mathsf{Verify}(pp, C^{out}, r^Q, z^Q | C^{out} \neq \perp, z \leftarrow Z)$*, we have* $z_{C\neq\perp}^{out} \approx_{\mathsf{negl}(\kappa)}^{s} Z$.

Remark 4. The modularity lemmata (Section 5) all hold for statistical RVCS, with minor adjustments in the proofs. In particular, the proofs only need to (straight-forwardly) account for negligible distances between $z_{C\neq\perp}^{out}$ and Z. We therefore defer the proofs to Appendix C.

Definition 23. *Given a mechanism ensemble* $\mathcal{M}_F := \{\mathcal{M}_f : \mathcal{D} \to R\}_{f \in F}$ *for function class* F*, a* statistical certified probabilistic mechanism *(sCertPM) for* $\mathcal{M}_F$ *is defined as in Definition 15 but with the cheating-verifier soundness property replaced by: There exists a negligible function* negl *such that for every cheating verifier* $\tilde{\mathsf{V}}$ *and every pair* $D, D' \in \mathcal{D}$ *and some* $y \in R$, $\mathsf{VIEW}^y_{\tilde{\mathsf{V}}}(D) \sim \mathsf{VIEW}^y_{\tilde{\mathsf{V}}}(D')$*. Here,* $\mathsf{VIEW}^y_{\tilde{\mathsf{V}}}(D)$ *denotes the view of* $\tilde{\mathsf{V}}$ *in* $\langle \mathsf{P}(D), \tilde{\mathsf{V}} \rangle(pp, f)$ *conditioned on that* $Q = y$*, with* Q *being the proposed output from* P*. Furthermore, with* $Q_{\neq \perp}$ *denoting* Q *conditioned on* $\tilde{\mathsf{V}}$ *not aborting, we have* $Q_{\neq \perp} \approx^s_{\mathsf{negl}(\kappa)} \mathcal{M}_f(D)$*.*

Theorem 3. *Let* $\mathcal{M}_F$ *be an additive-noise mechanism ensemble with noise distribution* Z *and let* CS *be an additive* f*-homomorphic commitment scheme for all* $f \in F$*. Then the CertPM construction (Definition 19) instantiated with an sRVCS is an sCertPM for* $\mathcal{M}_F$*.*

Theorem 3 follows by minor changes to the proofs of the corresponding theorems for perfect CertPM (BGKW and Theorem 2) so we defer it to Appendix B.2.

More precisely, the correctness and cheating-prover soundness properties are unaffected by the RVCS being weakened and the cheating-verifier property follows by the same arguments as in the perfect case, but with the negligible term added to the statistical distance between $Q_{\neq \perp}$ and $\mathcal{M}_f(D)$.

Definition 24. *Let* $F \subseteq \{f : \mathcal{D} \to R \setminus \{\perp\}\}$ *be a function family. A* statistical certified (α, β)-accurate (ε, δ)-DP *(*$(\alpha, \beta, \varepsilon, \delta)$*-sCertDP) protocol for* F *is an interactive protocol* (P, V) *such that for all* $D \in \mathcal{D}, f \in F$*, it fulfills honest-curator DP and dishonest-curator DP as in Definition 16 and* correctness*: There exists a negligible function* negl *such that for* $\mathcal{M}_f(D) \leftarrow \langle \mathsf{P}(D), \mathsf{V} \rangle(pp, f)$*,* $\mathcal{M}_f$ *is* $\big(\alpha, \beta + \mathsf{negl}(\kappa)\big)$*-accurate and* $\big(\varepsilon, \delta + \mathsf{negl}(\kappa)\big)$*-DP.*

The weakening lies entirely in that we allow the honest protocol to abort with non-zero probability. Just as for CertDP, any sCertPM for a DP mechanism is an sCertDP protocol. Again, we defer the proof of this to Appendix B.3 due to similarity to BGKW's corresponding proof.

Theorem 4. *For any constants* $\alpha, \varepsilon > 0$ *and* $\beta, \delta \in [0, 1]$*, let* $\mathcal{M}_F$ *be an* (α, β)*-accurate* (ε, δ)*-DP mechanism ensemble for function class* F*. Then any sCertPM for* $\mathcal{M}_F$ *is a* $(\alpha, \beta, \varepsilon, \delta)$*-sCertDP protocol.*

7 New Random Variable Commitment Schemes

We now present our new RVCS constructions. For the remaining, $R = \mathbb{Z}_q \cup \{\perp\}$ and CS has message space $\chi = \mathbb{Z}_q$. Whenever an AoK is run, the player will abort if they reject the AoK.

7.1 RVCS for Biased Coins

By using Lemma 7, we can multiply two commitments to fair coins and get one commitment to a coin with distribution $Ber(1/4)$. Then Lemma 6 allows us to turn this into a commitment of a coin with distribution $Ber(3/4)$, by negating it (taking a commitment of 1 minus the commitment to our biased coin). By iteratively multiplying the tentative result by a fair coin and negating in the right iterations, we can generalize to a coin with distribution $Ber(p)$ with $p = n/2^k$ for $k \in \mathbb{N}, n < 2^k$. Below is an RVCS constructed this way, in recursive form. $\mathsf{WI\text{-}AoK}_{mult}$ is for relation $\mathcal{R}_\phi^{\mathsf{CS}}$ with $\phi(b_1, b_2, b_3) = 1_{\{(m_1,m_2,m_3)\in\mathbb{Z}_q^3 : m_3 = m_1 \cdot m_2\}}(b_1, b_2, b_2)$.

Definition 25 (Protocol $\mathsf{BerCS}(n,k)$). *For integers $k \geq 1, n \in [1, 2^k)$, we define the protocol $\mathsf{BerCS}(n,k) = (\mathsf{CS}, \mathsf{DE}, \mathsf{PL})(n,k)$ as follows:*

1. *If $n = 2^{k-1}$:* DE *and* PL *together run* CoinCS *and both get the result $C_{b_{res}}$.*
   Else if $n < 2^{k-1}$:
   - DE *and* PL *together run* CoinCS *and* $\mathsf{BerCS}(n, k-1)$*, and get C_1, C_2.* DE *gets the samples b_1, b_2.*
   - DE *computes $b_3 \leftarrow b_1 \cdot b_2$, commits to it in C_3 and sends is to* PL.
   - DE *and* PL *run* $\mathsf{WI\text{-}AoK}_{mult}$ *on C_1, C_2, C_3. Both parties set $C_{res} \leftarrow C_3$.*

   Else ($2^{k-1} < n < 2^k$):
   - DE *and* PL *together run* CoinCS *and* $\mathsf{BerCS}(n, k-1)$*, and get C_1, C_2.* DE *gets the samples b_1, b_2.*
   - DE *computes $b_3 \leftarrow b_1 \cdot b_2$, commits to it in C_3 and sends is to* PL.
   - DE *and* PL *run* $\mathsf{WI\text{-}AoK}_{mult}$ *on C_1, C_2, C_3.*
   - DE *computes $(C_{one}, r_{one}) \leftarrow \mathsf{Commit}(1)$ and sends (C_{one}, r_{one}) to* PL.
   - DE *sets $b_{res} \leftarrow 1 - b_3$ and both parties compute $C_{res} \leftarrow C_{one} \hat{-} C_{b_3}$.*
2. PL *returns C_{res}.*

Theorem 5. *For any $k \geq 1, n \in [1, 2^k)$, $\mathsf{BerCS}(n,k)$ is an RVCS for $Ber(\frac{n}{2^k})$.*

Proof. The proof follows from recursively lowering k and the modularity lemmata. That is, the BerCS protocol consists essentially solely homomorphic or CnP post-processing of the results of recursive calls to itself and CoinCS, which all iteratively lower the parameter k until the base case $k = n = 1$ is reached (or $n = 2^{k-1}$), in which case BerCS = CoinCS. We prove all the RVCS properties simultaneously. Let $k \geq 2$ be arbitrary and n be any integer in $[1, 2^k)$. Assume for now that $\mathsf{BerCS}(n', k-1)$, for any $n' \in [1, 2^{k-1})$, is an RVCS for $Ber(n'/2^{k-1})$.

If $n = 2^{k-1}$ then BerCS is simply a call to CoinCS so therefore all properties except for correctness follows directly from CoinCS being an RVCS. Correctness follows from noting that $Ber(2^{k-1}/2^k) = Ber(1/2)$.

If $1 \leq n < 2^{k-1}$ then BerCS is precisely the multiplicative CnP post-processing of CoinCS and $\mathsf{BerCS}(n, k-1)$. As per the assumption that $\mathsf{BerCS}(n, k-1)$ is an RVCS, Lemma 7 implies that $\mathsf{BerCS}(n,k)$ is an RVCS for $Ber(n/2^k)$, since $b_1 \sim Ber(1/2)$ and $b_2 \sim Ber(n/2^{k-1})$ and their product b_3 has distribution $Ber(1/2 \cdot n/2^{k-1}) = Ber(n/2^k)$.

If $2^{k-1} < n < 2^k$, the parties start by performing the multiplicative CnP post-processing of CoinCS and $\mathsf{BerCS}(2^k - n, k-1)$, so just as in the bulletpoint

above, the steps taken until DE sends the commitment of 1 constitute an RVCS for $Ber((2^k - n)/2^k)$. Now, by the homomorphic property of CS, DE indeed holds opening information so that they can open C_{res} to $1 - b_3$, which has the distribution $Ber(1 - (2^k - n)/2^k) = Ber(n/2^k)$.

So the theorem for the case k follows from the assumption that BerCS for the case $k - 1$ is an RVCS (for any valid n'). Now we need only note that the assumption being true follows from that CoinCS is an RVCS for $Ber(1/2)$ since the case $k = 1$ (which necessitates $n = 1$) satisfies $n = 2^{k-1}$ and thus $\mathsf{BerCS}(1, 1)$ is simply a call to CoinCS. □

7.2 Statistical RVCS for Truncated Discrete Laplace Distribution

We now present a sRVCS based on the algorithm for the FDL_1 distribution (henceforth called FDL) in [35].

Definition 26 ([35]). *For $p \in (0, 1), N \in \mathbb{N}$, define the* truncated geometric distribution *$tGeo(p, N)$ by the pmf $f_{tGeo}(z) \propto (1 - p)p^z$ for $z \in \{0, \dots N - 1\}$. For $p \in (0, 1), N, M \in \mathbb{N}$ with $M < N$, $N < q/2$ and $X_1, X_2 \sim tGeo(p, N)$, let the distribution X on $\mathbb{Z}_q \cup \{\perp\}$ be $X_1 - X_2$ if $|X_1 - X_2| \leq M$ and $\perp$ otherwise. Define $FDL(p, N, M)$ as $X_{\neq\perp}$.*

Proposition 2 ([35]). *Let $f : \mathcal{D} \to \mathbb{Z}_q$ be a function with l_1-sensitivity Δ and $|f(D)| \leq L$ for all $D \in \mathcal{D}$ and a constant L. Let $N+L < q/2$, $p^{-\Delta}\frac{1-p^{2(N-M)}}{1-p^{2(N-M-\Delta)}} \leq e^{\varepsilon}$ and $\frac{p^{M-\Delta}}{(1-p)(1-p^{2N})} \leq \delta$. Then the FDL mechanism $\mathcal{M}_{FDL}(\cdot) := f(\cdot) + z$ with $z \sim FDL(p, N, M)$ is (ε, δ)-DP. Further, $\mathcal{M}_{FDL}$ is (α, β)-accurate with $\alpha = \frac{\Delta}{\varepsilon} \log\left(\frac{2}{\beta(1-p)(1-p^{2N})}\right)$.*

The accuracy of the FDL mechanism is essentially the same as for the Laplace mechanism. The algorithm for FDL [35] builds on 'bit-wise' sampling [33], where sampling $tGeo(p, N = 2^c)$ is reduced to computing the sum $\sum_{i=0}^{c-1} b_i 2^i$ with $b_i \sim Ber(\eta_i)$ for $\eta_i := 1/\left(1 + p^{-2^i}\right)$. As $\mathsf{BerCS}(n, k)$ samples from $Ber(n/2^k)$ we can, by suitable choice of n and k, approximate each $Ber(\eta_i)$ to a statistical distance at most 2^{-k}. By setting, say, $k \propto \kappa$, we can approximate FDL to a negligible statistical distance. We approximate FDL by the distribution FDL'.

Definition 27 ([35]). *For $p \in (0, 1), c, k, M \in \mathbb{N}, N = 2^c < q/2, M < N$, define $\left\{\eta_i := 1/\left(1 + p^{-2^i}\right)\right\}_{i\in[c]}$, $\{n_i := \lceil \eta_i \cdot 2^k \rfloor\}_{i\in[c]}$ and let $X_1 := \sum_{i=0}^{c-1} 2^i b_i$, $X_2 := \sum_{i=0}^{c-1} 2^i b'_i$ with i.i.d. $b_i, b'_i \sim Ber(n_i/2^k)$. Define*

$$FDL'(p, N, M, k) := \begin{cases} X_1 - X_2, \text{ if } |X_1 - X_2| \leq M \\ \perp, \text{ otherwise.} \end{cases} \tag{1}$$

Proposition 3 ([35]). *For parameters as above, we have $\mathbb{P}(FDL'(p, N, M, k) = \perp) \leq 2p^{M+1}/((1+p)(1-p^N)^2)$ and the statistical distance between $FDL(p, N, M)$ and $FDL'(p, N, M, k)$ is at most $c \cdot 2^{-k+1} + 2p^{M+1}/((1 + p)(1 - p^N)^2)$.*

Below we present our sRVCS protocol for FDL'. $\mathsf{WI\text{-}AoK}_M$ denotes a range-proof, i.e. for relation $\mathcal{R}_\phi^{\mathsf{CS}}$ with $\phi(z) = \mathbf{1}_{\{x \in \mathbb{Z}_q : -M \leq x \leq M\}}(z)$.

Definition 28 (Protocol DLapCS). *For parameters $p \in (0,1)$, $N, M, k \in \mathbb{N}_+$ with $N = 2^c < q/2, M < N$ for $c \in \mathbb{N}_+$, let $\eta_i := 1/\left(1 + p^{-2^i}\right)$ and $n_i \leftarrow \lceil \eta_i \cdot 2^k \rfloor$ for $i \in [c]$. We define the protocol $\mathsf{DLapCS}(p, N, M, k) = (\mathsf{CS}, \mathsf{DE}, \mathsf{PL})(p, N, M, k)$ as follows.*

1. DE *and* PL *sequentially run* $\mathsf{BerCS}(n_i, k)$ *twice, giving C_i, C_i', respectively, for $i \in [c]$.* DE *gets samples b_i, b_i' and opening information r_i, r_i', for $i \in [c]$.*
2. DE *computes $x \leftarrow \sum_{i=0}^{c-1} 2^i b_i$, $x' \leftarrow \sum_{i=0}^{c-1} 2^i b_i', r_x \leftarrow \check{\sum}_{i=0}^{c-1} 2^i r_i$ and $r_{x'} \leftarrow \check{\sum}_{i=0}^{c-1} 2^i r_i'$. Both parties compute $C_x \leftarrow \hat{\sum}_{i=0}^{c-1} 2^i C_i$, $C_x' \leftarrow \hat{\sum}_{i=0}^{c-1} 2^i C_i'$.*
3. *Both parties compute $C_e = C_x \hat{-} C_x'$.* DE *computes $r_e = r_x \check{-} r_x'$.*
4. DE *computes $e = x - x'$ and if $|e| > M$ then* DE *aborts, else the parties together execute* $\mathsf{WI\text{-}AoK}_M$ *on C_e.*
5. PL *outputs C_e.*

As we require honest parties to be PPT, k, c and M must be polynomial in κ.

Theorem 6. *For parameters $p, M, N = 2^c, k$ such that $\mathbb{P}(FDL'(p, N, M, k) = \perp)$ is negligible and c, k are polynomial, the protocol $\mathsf{DLapCS}(p, N, M, k)$ is an sRVCS for the $FDL'(p, N, M, k)$ distribution.*

Proof. Step 1 consists of polynomially many RVCS's run sequentially so by Lemma 5, step 1 is an RVCS for $Ber(\frac{n_0}{2^k}) \times \ldots \times Ber(\frac{n_{c-1}}{2^k})$. By Lemma 6, the first three steps of DLapCS is an RVCS for $X_1 - X_2$ as in Definition 27. Thus the rest of the proof only needs to show that step 4 does not contradict any of the properties of an sRVCS.

Correctness: If $|e| > M$ then DE sends $C^Q = \perp$ and PL sets $C^{out} = \perp$. Else if $|e| \leq M$ then PL will always accept the AoK (due to completeness) and set $C^{out} = C^Q = C_e$. Then correctness follows from the first three steps being an RVCS for $X_1 - X_2$.

Cheating-player distributional soundness: As $\mathsf{WI\text{-}AoK}_M$ is WI, the view of any cheating player $\tilde{\mathsf{PL}}$ is distributed independently of the underlying value of C_e. Thus $\tilde{\mathsf{PL}}$ can only choose to either output $C^Q = C_e$ or $\perp$ and this choice must be made independently of the sampled value. $\mathbb{P}(C^{out} = \perp)$ is the sum of the probability that the honest dealer aborts due to $|e| > M$ and the probability that $\tilde{\mathsf{PL}}$ causes an abort. As both of these cases are ignored when conditioning z^{out} on $C^{out} \neq \perp$, the statistical distance between $z^{out}_{C \neq \perp}$ and $FDL'(p, N, M, k)$ is precisely $\mathbb{P}(FDL'(p, N, M, k) = \perp)$, which is negligible.

Cheating-dealer distributional soundness: Let $\gamma_\perp^{start}$ be the probability of abort in steps 1-3 and $\gamma_\perp^{end}$ be the probability of abort during $\mathsf{WI\text{-}AoK}_M$. Note that $\gamma_\perp = \gamma_\perp^{start} + \gamma_\perp^{end}$. Since steps 1-3 constitute an RVCS for $X_1 - X_2$, there exists an EPT procedure $\mathsf{RevealOpening}_{start}$ which given the transcript of steps 1-3 outputs $(r^Q_{start}, z^Q_{start})$ such that $z^{start} \leftarrow \mathsf{Verify}(pp, C_e, r_{start}, z^Q_{start})$ has statistical distance at most $\gamma_\perp^{start} + \mathsf{negl}(\kappa)$ to $X_1 - X_2$ and thus (by Proposition 3) also to $FDL'(p, N, M, k)$, for some negligible function negl.

Let RevealOpening be as follows. It gets a transcript and if there was no abort in steps 1-3, then it runs $\mathsf{RevealOpening}_{start}$ and we define e' as the extracted sample z^Q_{start}. Otherwise we define $e' = \perp$. Now if PL accepted in step 4, RevealOpening outputs $(r^Q, z^Q) = (r^Q_{start}, e')$ and otherwise $(r^Q, z^Q) = (\perp, \perp)$. This means that $z^{out} \leftarrow \mathsf{Verify}(pp, C_e, r^Q, z^Q)$ is either e' or $\perp$.

If the transcript as a whole is rejecting with overwhelming probability, then cheating-dealer distributional soundness directly holds, as statistical distances cannot be larger than 1, so assume that PL accepts with non-negligible probability. Then, in particular, PL accepts in step 4 with non-negligible probability so there exists an EPT extractor K of $\mathsf{WI\text{-}AoK}_M$ which extracts some $e'' \in [M, M]$ (with opening information r''_e) with non-negligible probability. So we now have two different EPT extractors for C_e, one by RevealOpening and one by K, that succeed with non-negligible probability so with overwhelming probability $e' = e''$, by binding. So if RevealOpening manages to extract a valid opening (to e'), then $e' \in [M, M]$ with overwhelming probability.

By the specification of RevealOpening, $z^{out} = e'$ whenever PL does not abort during step 4, so $SD(z^{out}, e') \leq \gamma^{end}_{\perp} + \mathsf{negl}'(\kappa)$, for some negligible negl'. Thus we have, by the triangle inequality (Proposition 1), that

$$SD(z^{out}, FDL'(p, N, M, k)) \leq SD(z^{out}, e') + SD(e', FDL'(p, N, M, k))$$
$$\leq \gamma^{end}_{\perp} + \mathsf{negl}'(\kappa) + \gamma^{start}_{\perp} + \mathsf{negl}(\kappa) = \gamma_{\perp} + \mathsf{negl}''(\kappa).$$

□

By having an sRVCS for a discrete Laplace distribution, we directly get a statistical certified discrete Laplace mechanism via Theorems 3 and 4. As our focus is on the theoretical properties of the RVCS definition(s), we defer the detailed treatment of this mechanism to Appendix D. Similarly, we defer the efficiency analysis, both analytic and experimental, of our RVCS's to Appendix E. For use in a (w)CertDP mechanism, if one requires a total δ of roughly $2^{-\kappa}$ and a constant ε, and we have $k, M \propto \kappa$ and $c \propto \log_2(\kappa)$, then BinCS has constant round complexity and computation/communication complexity linear in κ, whereas DLapCS has all metrics proportional to $\kappa \log_2(\kappa)$. Our experiments, run on a common desktop, show for $\varepsilon = 1, \delta = 2^{-70}$ that DLapCS has a total runtime of around 1.1 seconds per party. As a comparison, each individual run of CoinCS takes around 0.3 ms per party and BinCS requires generating roughly 400 coins so, as the cost of BinCS is dominated by generating the coins, BinCS has a total runtime of around 120 ms per party, making it roughly 10 times faster than DLapCS. We note that the Laplace and binomial mechanisms have fundamentally different privacy-utility trade-offs [33,23] and therefore these benchmarks indicate that DLapCS plausibly is a practical alternative to BinCS, in use-cases that favor a Laplace mechanism.

8 Open Directions

As RVCS is a new primitive, there are many tempting directions for future work. We have established that RVCS's exist for any efficiently sampleable distribution,

under the discrete logarithm assumption, and given a generic modular way of constructing them. A natural next step is to design more efficient RVCS's, for instance, by utilizing more efficient AoKs and/or by algorithmic improvements in the sampling strategy. For instance, our framework allows replacing all of the Σ-protocols by any suitable $(n_1, \ldots, n_\mu)$-special-sound protocol, which could give large improvements in performance, especially when used in CnP post-processing for somewhat large circuits. Further, one might be able to improve the design strategy of running an RVCS for fair coins and then applying CnP post-processing, for instance by removing the need for these two separate stages.

As, generically seen, an RVCS is a publicly verifiable two-party protocol for sampling, it would be interesting to relate what is known about RVCS to the well-developed existing theory of multiparty sampling [25,22,1] and publicly verifiable multiparty computation [10,9]. Given recent progress on practical multiparty noise sampling [45,37,51] and publicly verifiable multiparty computation [48], it would be fruitful to find specific applications where both such protocols (for sampling) and RVCS are suitable alternatives and to compare their merits.

Acknowledgments. We thank Shibam Mukherjee for many helpful discussions during the early stages of this work. We thank anonymous reviewers for feedback that significantly improved the disposition and clarity of this work.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abram, D., Fehr, S., Obremski, M., Scholl, P.: On the impossibility of actively secure distributed samplers. In: TCC. LNCS, vol. 16271, pp. 547–581. Springer (2025). https://doi.org/10.1007/978-3-032-12290-2_18
2. Agarwal, N., Suresh, A.T., Yu, F.X., Kumar, S., McMahan, B.: cpsgd: Communication-efficient and differentially-private distributed SGD. In: NeurIPS. pp. 7575–7586 (2018), https://proceedings.neurips.cc/paper/2018/hash/21ce689121e39821d07d04faab328370-Abstract.html
3. Attema, T.: Compressed Sigma-protocol Theory. Ph.D. thesis, Leiden University (2023), https://hdl.handle.net/1887/3619596
4. Attema, T., Cramer, R.: Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In: CRYPTO'20. LNCS, vol. 12172, pp. 513–543. Springer (2020). https://doi.org/10.1007/978-3-030-56877-1_18
5. Attema, T., Cramer, R., Fehr, S.: Compressing proofs of k-out-of-n partial knowledge. In: CRYPTO'21. pp. 65–91. Springer (2021)
6. Attema, T., Cramer, R., Kohl, L.: A compressed Σ-protocol theory for lattices. In: CRYPTO'21. LNCS, vol. 12826, pp. 549–579. Springer (2021). https://doi.org/10.1007/978-3-030-84245-1_19
7. Attema, T., Fehr, S., Resch, N.: Generalized special-sound interactive proofs and their knowledge soundness. In: TCC'23. LNCS, vol. 14371, pp. 424–454. Springer (2023). https://doi.org/10.1007/978-3-031-48621-0_15

8. Barak, B., Lindell, Y., Vadhan, S.P.: Lower bounds for non-black-box zero knowledge. J. Comput. Syst. Sci. **72**(2), 321–391 (2006). https://doi.org/10.1016/J.JCSS.2005.06.010
9. Baum, C., Damgård, I., Orlandi, C.: Publicly auditable secure multi-party computation. In: SCN'14. LNCS, vol. 8642, pp. 175–196. Springer (2014). https://doi.org/10.1007/978-3-319-10879-7_11
10. Baum, C., David, B., Dowsley, R.: (public) verifiability for composable protocols without adaptivity or zero-knowledge. In: Provable and Practical Security. Springer (2022)
11. Beimel, A., Nissim, K., Omri, E.: Distributed private data analysis: Simultaneously solving how and what. In: CRYPTO'08. LNCS, vol. 5157, pp. 451–468. Springer (2008). https://doi.org/10.1007/978-3-540-85174-5_25
12. Bell, Z.R., Goldwasser, S., Kim, M.P., Watson, J.: Certifying private probabilistic mechanisms. In: CRYPTO'24. LNCS, vol. 14925, pp. 348–386. Springer (2024). https://doi.org/10.1007/978-3-031-68391-6_11
13. Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: CRYPTO' 92. pp. 390–420. Springer Berlin Heidelberg, Berlin, Heidelberg (1993)
14. Berghel, S., Bohannon, P., Desfontaines, D., Estes, C., Haney, S., Hartman, L., Hay, M., Machanavajjhala, A., Magerlein, T., Miklau, G., Pai, A., Sexton, W., Shrestha, R.: Tumult analytics: a robust, easy-to-use, scalable, and expressive framework for differential privacy (2022), https://arxiv.org/abs/2212.04133
15. Bernstein, D.J.: Curve25519: New diffie-hellman speed records. In: PKC. LNCS, vol. 3958, pp. 207–228. Springer (2006). https://doi.org/10.1007/11745853_14
16. Biswas, A., Cormode, G.: Interactive proofs for differentially private counting. In: CCS'23. p. 1919–1933. CCS '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3576915.3616681
17. Blum, M.: Coin flipping by telephone a protocol for solving impossible problems. SIGACT News **15**(1), 23–27 (1983). https://doi.org/10.1145/1008908.1008911
18. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: EUROCRYPT'16. LNCS, vol. 9666, pp. 327–357. Springer (2016). https://doi.org/10.1007/978-3-662-49896-5_12
19. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: TCC'16. LNCS, vol. 9985, pp. 635–658 (2016). https://doi.org/10.1007/978-3-662-53641-4_24
20. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: IEEE S&P'18. pp. 315–334. IEEE Computer Society (2018). https://doi.org/10.1109/SP.2018.00020
21. Bünz, B., Chiesa, A., Lin, W., Mishra, P., Spooner, N.: Proof-carrying data without succinct arguments. In: CRYPTO. LNCS, vol. 12825, pp. 681–710. Springer (2021). https://doi.org/10.1007/978-3-030-84242-0_24
22. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: EUROCRYPT'03. LNCS, vol. 2656, pp. 68–86. Springer (2003). https://doi.org/10.1007/3-540-39200-9_5
23. Canonne, C.L., Kamath, G., Steinke, T.: The discrete gaussian for differential privacy. In: NeurIPS'20 (2020), https://proceedings.neurips.cc/paper/2020/hash/b53b3a3d6ab90ce0268229151c9bde11-Abstract.html
24. Cascudo, I., Damgård, I., David, B., Döttling, N., Dowsley, R., Giacomelli, I.: Efficient UC commitment extension with homomorphism for free (and appli-

cations). In: ASIACRYPT'19. LNCS, vol. 11922, pp. 606–635. Springer (2019). https://doi.org/10.1007/978-3-030-34621-8_22
25. Cleve, R.: Limits on the security of coin flips when half the processors are faulty (extended abstract). In: STOC'86. pp. 364–369. ACM (1986). https://doi.org/10.1145/12130.12168
26. arkworks contributors: `arkworks` zksnark ecosystem (2022), https://arkworks.rs
27. Cramer, R.: Modular Design of Secure yet Practical Cryptographic Protocols. Ph.D. thesis, University of Amsterdam (Jan 1997), https://ir.cwi.nl/pub/21438
28. Cramer, R., Damgård, I.: Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In: CRYPTO'98. LNCS, vol. 1462, pp. 424–441. Springer (1998). https://doi.org/10.1007/BFB0055745
29. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: CRYPTO'94. LNCS, vol. 839, pp. 174–187. Springer (1994). https://doi.org/10.1007/3-540-48658-5_19
30. Damgård, I.: On sigma protocols (2010), https://www.cs.au.dk/~ivan/Sigma.pdf, lecture notes
31. Dao, Q., Grubbs, P.: Spartan and bulletproofs are simulation-extractable (for free!). In: EUROCRYPT'23. LNCS, vol. 14005, pp. 531–562. Springer (2023). https://doi.org/10.1007/978-3-031-30617-4_18
32. Dwork, C.: Differential privacy. In: ICALP'06. p. 1–12. ICALP'06, Springer-Verlag, Berlin, Heidelberg (2006), https://doi.org/10.1007/11787006_1
33. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: EUROCRYPT'06. LNCS, vol. 4004, pp. 486–503. Springer (2006). https://doi.org/10.1007/11761679_29
34. Dwork, C., McSherry, F., Nissim, K., Smith, A.D.: Calibrating noise to sensitivity in private data analysis. In: TCC'06. LNCS, vol. 3876, pp. 265–284. Springer (2006). https://doi.org/10.1007/11681878_14
35. Eriguchi, R., Ichikawa, A., Kunihiro, N., Nuida, K.: Efficient noise generation protocols for differentially private multiparty computation. IEEE TDSC **20**(6), 4486–4501 (2023). https://doi.org/10.1109/TDSC.2022.3227568
36. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: STOC'90. p. 416–426. STOC '90, Association for Computing Machinery, New York, NY, USA (1990). https://doi.org/10.1145/100216.100272
37. Franzese, O., Fang, C., Garg, R., Wang, X., Jha, S., Papernot, N., Dziedzic, A.: Secure noise sampling for differentially private collaborative learning. In: CCS'25. pp. 4649–4663. ACM (2025). https://doi.org/10.1145/3719027.3744834
38. Goldreich, O.: Foundations of Cryptography: Volume 1. Cambridge University Press, USA (2006)
39. Jaeger, J., Tessaro, S.: Expected-time cryptography: Generic techniques and applications to concrete soundness. In: TCC'20. LNCS, vol. 12552, pp. 414–443. Springer (2020). https://doi.org/10.1007/978-3-030-64381-2_15
40. Johnson, N.M., Near, J.P., Hellerstein, J.M., Song, D.: Chorus: a programming framework for building scalable differential privacy mechanisms. In: EuroS&P'20. pp. 535–551. IEEE (2020). https://doi.org/10.1109/EUROSP48549.2020.00041
41. Klooß, M.: On Efficient Zero-Knowledge Arguments. Ph.D. thesis, Karlsruhe Institute of Technology, Germany (2023), https://d-nb.info/1282148052
42. Lipmaa, H., Parisella, R., Siim, J.: Constant-size zk-snarks in ROM from falsifiable assumptions. In: EUROCRYPT'24. LNCS, vol. 14656, pp. 34–64. Springer (2024). https://doi.org/10.1007/978-3-031-58751-1_2
43. Maurer, U.: Zero-knowledge proofs of knowledge for group homomorphisms. DCC **77**(2-3), 663–676 (2015). https://doi.org/10.1007/S10623-015-0103-5

44. Maurer, U.M.: Unifying zero-knowledge proofs of knowledge. In: AFRICACRYPT'09. LNCS, vol. 5580, pp. 272–286. Springer (2009). https://doi.org/10.1007/978-3-642-02384-2_17
45. Meisingseth, F., Rechberger, C., Schmid, F.: Accelerating multiparty noise generation using lookups. Cryptology ePrint Archive, Paper 2025/805 (2025), https://eprint.iacr.org/2025/805, to appear at CCS'26
46. Mironov, I., Pandey, O., Reingold, O., Vadhan, S.P.: Computational differential privacy. In: CRYPTO' 09. LNCS, vol. 5677, pp. 126–142. Springer (2009). https://doi.org/10.1007/978-3-642-03356-8_8
47. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: CRYPTO'91. LNCS, vol. 576, pp. 129–140. Springer (1991). https://doi.org/10.1007/3-540-46766-1_9
48. Rivinius, M.: MPC with publicly identifiable abort from pseudorandomness and homomorphic encryption. In: EUROCRYPT'25. LNCS, vol. 15605, pp. 270–300. Springer (2025). https://doi.org/10.1007/978-3-031-91092-0_10
49. Thaler, J.: Proofs, arguments, and zero-knowledge. Found. Trends Priv. Secur. **4**(2-4), 117–660 (2022). https://doi.org/10.1561/3300000030
50. Warner, S.L.: Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association **60**(309), 63–69 (1965). https://doi.org/10.1080/01621459.1965.10480775
51. Wei, C., Yu, R., Fan, Y., Chen, W., Wang, T.: Securely sampling discrete gaussian noise for multi-party differential privacy. In: CCS '23. ACM (2023), https://doi.org/10.1145/3576915.3616641
52. Wilson, R.J., Zhang, C.Y., Lam, W., Desfontaines, D., Simmons-Marengo, D., Gipson, B.: Differentially private SQL with bounded user contribution. PoPETS'20 **2020**(2), 230–250 (2020). https://doi.org/10.2478/POPETS-2020-0025

A Additional Preliminaries

A.1 Differential Privacy

We recall the standard definition of (ε, δ)-DP [34,32]. A *mechanism* is a probabilistic algorithm that maps a database D to an output in some range R. A database is an element in the domain $\mathcal{D} = \chi^*$ for some data domain χ.

Definition 29. *Let $\varepsilon \geq 0, \delta \in [0,1]$. A mechanism $\mathcal{M} : \mathcal{D} \to R$ is* (ε, δ)-differentially private (DP) *if for all pairs (D, D') of adjacent databases (i.e. that differ in at most one element) in $\mathcal{D}$ and all subsets S of R,*

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(D') \in S) + \delta, \tag{2}$$

where the probability is over $\mathcal{M}$'s internal coin tosses.

We focus on *additive-noise mechanisms* of the form $\mathcal{M}_f(\cdot) := f(\cdot) + z$ where f is a function from $\mathcal{D}$ to R and z is a random variable on R with distribution Z. For a function family F, $\mathcal{M}_F$ is the mechanism ensemble $\{\mathcal{M}_f\}_{f \in F}$. We use (α, β)-accuracy to measure utility.

Definition 30 ((α,β)-accuracy). *For a function $f : \mathcal{D} \to R \setminus \{\perp\}$ and a mechanism $\mathcal{M} : \mathcal{D} \to R$, let $|\perp - y|$ be infinite for any $y \in R$. We say that $\mathcal{M}$ is (α,β)-*accurate for f *if for all $D \in \mathcal{D}$, $\mathbb{P}(|\mathcal{M}(D) - f(D)| > \alpha) \leq \beta$. For a function family F, a mechanism ensemble $\mathcal{M}_F = \{\mathcal{M}_f\}_{f\in F}$ is (α,β)-accurate for F if for all $f \in F$, $\mathcal{M}_f$ is (α,β)-accurate for f.*

The concentration of the noise must be calibrated to the *sensitivity* of the function, i.e. how much the function evaluation can differ between adjacent databases.

Definition 31. *The l_1-*sensitivity *of a function $f : \mathcal{D} \to R \subseteq \mathbb{R}^n$, for some $n \in \mathbb{N}$, is*

$$\Delta_1(f) := \max_{D,D'\in\mathcal{D} \text{ adjacent}} ||f(D) - f(D')||_1,$$

where $||\cdot||_1$ is the l_1-norm.

A.2 RVCS for Binomial Distribution

The binomial distribution $Bin(N,p)$ with $p \in [0,1], N \in \mathbb{N}$ is the distribution on $\mathbb{N} \cup \{0\}$ with pmf $f_{Bin}(z) := \binom{N}{z} p^z (1-p)^{N-z}$.

Definition 32 ([12]). *Let $\mathsf{CS} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ be an additive commitment scheme on $\mathbb{Z}_q$. We define the protocol $\mathsf{BinCS} := (\mathsf{CS}, \mathsf{DE}, \mathsf{PL})$ for $Bin(N, 1/2)$ as:*

1. DE *and* PL *run* CoinCS *N times in parallel, resulting in* DE *holding N fair coins $b_1, \ldots, b_N$ and opening information $r_1, \ldots, r_N$ and both parties holding $C_1, \ldots, C_N$.*
2. DE *computes $z_{Bin} \leftarrow \sum_{i=1}^{N} b_i$ and $r_{Bin} \leftarrow \check{\sum}_{i=1}^{N} r_i$. Both parties compute $C_{Bin} \leftarrow \hat{\sum}_{i=1}^{N} C_i$.*

B Deferred Proofs

We now give the proofs omitted in the main body.

B.1 Proof of Lemma 4

Proof. Let $(\mathcal{A}_{\Pi}, \tilde{\mathsf{P}})$ be a PPT adversary against Π, K be the EPT extractor and $\rho(\kappa)$ be $Real_{\mathcal{A}_{\Pi},\tilde{\mathsf{P}}}(\kappa)$. Let $\mathcal{A}$ against $\mathcal{R}_{alt}$ be as follows: It gets pp and runs $(x, aux) \leftarrow \mathcal{A}_{\Pi}$, $tr \leftarrow Tr(\langle \tilde{\mathsf{P}}(aux), \mathsf{V}\rangle(pp, x))$ and $w \leftarrow \mathsf{K}(x, tr)$. Then $\mathcal{A}$ outputs (pp, x, w). The runtime of $\mathcal{A}$ is polynomial in expectation since $\mathcal{A}_{\Pi}, \tilde{\mathsf{P}}$ and V are PPT and K is EPT. The probability of $(pp, x, w) \in \mathcal{R}_{ext}$ is at least $\rho(\kappa) - \eta(\kappa)$, where η is negligible, since Π is a PoK and $\mathcal{A}$ is exactly an emulation of the ideal experiment. Let p, p_{alt} denote the probability of the output being in $\mathcal{R}$ and $\mathcal{R}_{alt}$, respectively. Then, by the union bound, $\rho(\kappa) - \eta(\kappa) \leq p + p_{alt}$. As $\mathcal{R}_{alt}$ is hard and $\mathcal{A}$ is EPT, we know that $p_{alt} \leq \mathsf{negl}(\kappa)$ for some negligible function negl. Thus $p \geq \rho(\kappa) - \eta(\kappa) - \mathsf{negl}(\kappa)$.

□

B.2 Proof of Theorem 3 (sCertPM from sRVCS)

Proof. The proof is essentially identical to the proof of the corresponding theorem for perfect CertPM and CertDP. In particular, the parts on correctness and cheating-prover soundness are unchanged and that on cheating-verifier soundness differs only in the end where the distance between $Q_{\neq\perp}$ and $\mathcal{M}_f(D)$ is analyzed.

Correctness: The correctness of the sRVCS implies that the verifier accepts C_z with probability 1 (i.e. we have $\mathbb{P}(C_z = \perp) = \mathbb{P}(Z = \perp)$) and that the prover knows a witness such that it can open C_z to a z drawn from Z. The correctness of CS and that it is f-homomorphic for all $f \in F$ implies that the verifier always accepts C_Q and the prover can open this to $f(D) + z$. Thus the sCertPM construction is correct.

Cheating-verifier soundness: The first part of cheating-verifier soundness requires that the distribution of the view of any cheating verifier $\tilde{\mathsf{V}}$ does not change dependent on the database D of the prover, when one conditions on that $\mathcal{M}_f(D) = \mathcal{M}_f(D')$ for some $y \in R$. The first part of $\tilde{\mathsf{V}}$'s view is from the commitment phase and it consists of C_D (and the verifier's internal randomness but we ignore that in the following). The distribution of C_D is independent of D since CS is perfectly hiding. The second part of $\tilde{\mathsf{V}}$'s view is from the randomness generation phase and consists of the cheating player $\tilde{\mathsf{PL}}$'s view in the execution of sRVCS. The view of this player is independent of the database since the honest dealer (played by V) does not condition its messages within the sRVCS protocol on the database. Finally, in the querying phase, the only parts of $\tilde{\mathsf{V}}$'s view that depends on P is the opening information that $\tilde{\mathsf{V}}$ gets in the end and the result of the mechanism. Due to CS having perfectly hiding openings, the distribution of the opening information does not depend on the value it opens to and thus also not on the database in question, and since we condition on $\mathcal{M}_f(D) = \mathcal{M}_f(D')$, the view of $\tilde{\mathsf{V}}$ in the querying phase has distribution independent of the database used. Thus the full view of $\tilde{\mathsf{V}}$ is identically distributed in the two worlds where D and D' are used, respectively.

The second part of cheating-verifier soundness is that $Q_{\neq\perp}$ has at most negligible statistical distance to $\mathcal{M}_f(D)$. From the cheating-player distributional soundness of sRVCS, we have that $z^{out}_{C\neq\perp} \approx^s_{\mathsf{negl}} Z$. Thus, for all $D \in D, f \in F$

$$Q_{\neq\perp}(D) := f(D) + z^{out}_{C\neq\perp} \approx^s_{\mathsf{negl}} f(D) + Z =: \mathcal{M}_f(D).$$

So cheating-verifier soundness is confirmed.

Cheating-prover soundness: Since the proof for cheating-prover soundness does not depend on the cheating-player distributional soundness of the (w)RVCS, it follows precisely as in the perfect case from that CS is binding and the cheating-dealer distributional soundness of sRVCS. In particular, by the binding, the cheating prover can change the distribution of the values that C_Q opens to by at most a negligible statistical distance, unless it chooses to abort. Similarly, the cheating prover can change the distribution of the value it can open C_z to by at most a negligible statistical distance (unless it causes an abort). For more details, see the proof of Theorem 4.4 in BGKW [12]. □

B.3 Proof of Theorem 4 (sCertDP from sCertPM)

Proof. The correctness and dishonest-curator DP properties follows by the same arguments as for perfect CertDP and CertPM. Therefore, we focus on the honest-curator DP property. There we only need to ascertain that the negligible slack in the CertPM cheating-verifier soundness can be incorporated into the negligible δ-term.

Correctness: This follows directly from the correctness of the sCertPM and that the mechanism is (ε, δ)-DP and (α, β)-accurate.

Honest-curator DP: We need that the view of any cheating verifier $\tilde{\mathsf{V}}$ is a DP mechanism, with respect to the database of the curator. For any two databases (in particular, adjacent ones) $D, D' \in \mathcal{D}$, we compare the views $\mathsf{VIEW}_{\tilde{\mathsf{V}}(D)}$ and $\mathsf{VIEW}_{\tilde{\mathsf{V}}(D)}$. The views consist of $\tilde{\mathsf{V}}$'s internal randomness, the output Q of the protocol and the other messages $\tilde{\mathsf{V}}$ receives throughout the protocol. The internal randomness is clearly identically distributed in the two views. From cheating-verifier soundness, we get that if the outputs in the two views are the same, then the whole views are identically distributed, so in particular the probability that the protocol aborts is the same in both worlds. Thus we get

$$Q_{\neq\perp} \approx^s_{\mathsf{negl}} \mathcal{M}_q(D) \underset{DP:(\varepsilon,\delta)}{\approx} \mathcal{M}_q(D') \approx^s_{\mathsf{negl}} Q'_{\neq\perp},$$

with $\underset{DP:(\varepsilon,\delta)}{\approx}$ denoting similarity between distributions as required by the DP definition. That is: $\forall S \subseteq R : \mathbb{P}(\mathcal{M}_q(D) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}_q(D') \in S) + \delta$ and $\forall S \subseteq R : \mathbb{P}(\mathcal{M}_q(D') \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}_q(D) \in S) + \delta$. As we consider constant ε, this implies $Q_{\neq\perp} \underset{DP:(\varepsilon,\delta+\mathsf{negl})}{\approx} Q'_{\neq\perp}$. Thus the view of the adversary is $(\varepsilon, \delta+\mathsf{negl})$-DP.

Dishonest-curator DP: The dishonest-curator DP part follows precisely as in the case for perfect CertPM and CertDP, since the dishonest-curator property is independent from the honest-curator property (and in particular, relaxations made thereof). We refer to the proof of Theorem 3.4 in BGKW [12] for details. □

C Properties of Statistical RVCS

We now state and prove the modularity lemmata for sRVCS. The proofs for each of the theorems are essentially identical to the corresponding proofs for perfect RVCS. More precisely, correctness and cheating-dealer distributional soundness follows precisely as in case for perfect RVCS. This is due to the relaxation from perfect to statistical RVCS being only in the cheating-player distributional soundness. The proof for cheating-player distributional soundness is also almost identical to in the proofs in the perfect case – the only difference is that we need to account for negligible statistical distances trickling through the protocols. We highlight these differences in bold. In the proofs, we say that some random variable (the view or some choice of the cheating player) 'depends at most negligibly'

on the sampled value. By that we mean that for any two values z, z' of the sample, the distribution of the random variable given that the value z was sampled and the random variable given that value z' was sampled have at most negligible statistical distance. That is, the sampled value changes the distribution of the random variable in question by at most a negligible statistical distance.

C.1 Sequential composition for sRVCS

Lemma 8. *Given polynomially many sRVCS's* $\mathsf{sRVCS}_1, \ldots, \mathsf{sRVCS}_k$ *for distributions* $Z_1, \ldots, Z_k$*, respectively, the protocol where* $\mathsf{sRVCS}_1, \ldots, \mathsf{sRVCS}_k$ *are run sequentially is a sRVCS for the joint probability distribution* $Z_1 \times \ldots \times Z_k$*.*

Proof. Correctness and cheating-dealer distributional soundness follows precisely as in the proof of Lemma 5.

Cheating-player distributional soundness: Assume towards a contradiction that a cheating player can break the cheating-player distributional soundness of the composed protocol and let $\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<i}} := (\mathsf{VIEW}_{\tilde{\mathsf{PL}}_1}, \ldots, \mathsf{VIEW}_{\tilde{\mathsf{PL}}_{i-1}})$ be the view of $\tilde{\mathsf{PL}}$ up until the i'th protocol. That is, we assume there is a cheating player strategy $\tilde{\mathsf{PL}} = (\tilde{\mathsf{PL}}_1, \tilde{\mathsf{PL}}_2(\mathsf{VIEW}_{\tilde{\mathsf{PL}}_1}), \ldots, \tilde{\mathsf{PL}}_k(\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<k}}))$ for the composed protocol such that $(z_1^{out}, \ldots, z_k^{out}) := z_{C \neq \perp}^{out}$ has a **non-negligible statistical distance to** $Z_1 \times \ldots \times Z_k$. Since $SD(z_1^{out}, Z_1)$ **being non-negligible** directly contradicts the cheating-player distributional soundness of sRVCS_1, we let $i > 1$ be the index of the first output such that $SD(z_i^{out}, Z_i)$ **is non-negligible**.

Consider the cheating player strategy $\tilde{\mathsf{PL}}^*$ for sRVCS_i as follows; $\tilde{\mathsf{PL}}^*$ runs a copy of $(\mathsf{CS}, \mathsf{DE}_1, \tilde{\mathsf{PL}}_1), \ldots, (\mathsf{CS}, \mathsf{DE}_{i-1}, \tilde{\mathsf{PL}}_{i-1}(\mathsf{VIEW}_{\tilde{\mathsf{PL}}_{<i-1}}))$ internally (running both parties and then $\tilde{\mathsf{PL}}^*$ runs the interaction between DE_i and $\tilde{\mathsf{PL}}_i$. The simulated views of $\tilde{\mathsf{PL}}_1, \ldots, \tilde{\mathsf{PL}}_{i-1}$ in the internal executions are identically distributed to the views that $\tilde{\mathsf{PL}}$ gets from interacting with $\mathsf{DE}_1, \ldots, \mathsf{DE}_{i-1}$ in the composed (real) protocol. Therefore, the assumption that $\tilde{\mathsf{PL}}$ breaks cheating-player distributional soundness of the composed protocol implies that $\tilde{\mathsf{PL}}^*$ breaks the same property of sRVCS_i, which is a contradiction since sRVCS_i is an sRVCS. □

C.2 Homomorphic post-processing for sRVCS

Lemma 9. *The homomorphic post-processing of any sRVCS for distribution* Z *with respect to any* f *is an sRVCS for the distribution* $f(Z)$*.*

Proof. Correctness and cheating-dealer distributional soundness follow precisely as in the proof of Lemma 6.

Cheating-player distributional soundness: Denote the cheating player by $\tilde{\mathsf{PL}}_{post}$ and its actions during the execution of sRVCS_{base} by $\tilde{\mathsf{PL}}_{base}$. Since sRVCS_{base} is an sRVCS for Z, unless if $\tilde{\mathsf{PL}}_{base}$ causes an abort, z_{base} **has at most negligible statistical distance** to Z, C_{base}^{out} is a correct commitment to z_{base} and $r_{z_{base}}$ is the corresponding opening information. Due to the (statistical) cheating-player distributional soundness of sRVCS_{base}, the distribution

of the view of $\tilde{\mathsf{PL}}_{base}$ changes by **at most a negligible statistical distance** depending on the value of z_{base} and since in step 2, no new messages are sent between the parties, the distribution of $\tilde{\mathsf{PL}}_{post}$'s view depends **at most negligibly on** the value of z_{post}. Also, the lack of interaction in step 2 implies that the only way for $\tilde{\mathsf{PL}}_{post}$ to change the output distribution is to abort selectively. Thus, that $\tilde{\mathsf{PL}}_{post}$'s choice whether or not to abort must **depend at most negligibly on** z_{post} implies (statistical) cheating-player distributional soundness. □

C.3 CnP post-processing for sRVCS

Lemma 10. *Let* $\mathsf{sRVCS}_1, \ldots, \mathsf{sRVCS}_k$ *be polynomially many sRVCS's for distributions* $Z_1, \ldots, Z_k$ *over* χ*, respectively. Let* $f : \chi^k \to \chi$ *be a function which is computable in polynomial time, and let* $\mathsf{WI\text{-}AoK}_f$ *be a WI-AoK for* $\mathcal{R}_f^{\mathsf{CS}}$*. Then the CnP post-processing of* $\mathsf{sRVCS}_1, \ldots, \mathsf{sRVCS}_k$ *with respect to* f *is an sRVCS for* $f(Z_1 \times \ldots \times Z_k)$*.*

Proof. Correctness and cheating-dealer distributional soundness follows precisely as in the proof of Lemma 7.

Cheating-player distributional soundness: $\tilde{\mathsf{PL}}$ only sends messages during the executions of the sRVCS's and during the AoK. Due to the cheating-player distributional soundness of the sRVCS's, unless $\tilde{\mathsf{PL}}$ causes an abort, $z_1, \ldots, z_k$ will have distributions **negligibly close to** $Z_1, \ldots Z_k$, respectively. The interaction during the AoK occurs after DE has proposed an output commitment and thus no message from $\tilde{\mathsf{PL}}$ during the AoK can change the values which DE can open C_{post}^{out} to. Therefore, $\tilde{\mathsf{PL}}$ may only influence the distribution of those openings by aborting.

The distribution of the view of $\tilde{\mathsf{PL}}$ is **at most negligibly dependent** of y. Independence in step 1 is guaranteed by the cheating-player distributional soundness of the RVCS's, independence in the second step follows from the perfect hiding of CS and independence in the final step follows from $\mathsf{WI\text{-}AoK}_f$ being perfectly WI. Thus $\tilde{\mathsf{PL}}$ must make the choice of whether to abort at most **negligibly dependent on** y so if $\tilde{\mathsf{PL}}$ does not abort then y **has a distribution with negligible statistical distance to** $f(Z_1 \times \ldots \times Z_k)$. □

D A Certified Discrete Laplace Mechanism

We now use DLapCS to construct an sCertDP protocol. The $FDL'(p, N, M, k)$ distribution has statistical distance at most $c \cdot 2^{-k+1} + 2p^{M+1}/((1+p)(1-p^N)^2)$ from the $FDL(p, N, M)$ distribution, which gives DP when used in an additive-noise mechanism. Therefore FDL' also gives DP with a slightly larger δ-parameter, since any mechanism with statistical distance ν from some (ε, δ)-DP mechanism is $(\varepsilon, \delta + (e^\varepsilon + 1)\nu)$-DP [35]. Similarly, the accuracy is reduced with the β-parameter increasing by ν. As we have ν being negligible and consider constant ε, the decay of δ and β is negligible. Thus we get an sCertDP protocol by instantiating the CertPM construction (Definition 19) with DLapCS. We only

need to take care in the choice of parameters; Firstly, DLapCS is only an sRVCS for $FDL'(p, N, M, k)$ if the honest abort probability is negligible. That is, if $2^{-k+1} + 2p^{M+1}/((1+p)(1-p^N)^2)$ is negligible. This can be guaranteed by setting $k, M \propto \kappa$ as long as p is constant and less than 1. Secondly, for the use of the sequential composition lemma, we need that c is at most polynomial, and $N = 2^c > M$ with $M \propto \kappa$ demands that c is at least logarithmic, so we can set $c \propto \log_2(\kappa)$. Together with the parameter constraints from Proposition 2, we get the theorem below.

Theorem 7. *Let F be a family of functions $f : \mathcal{D} \to \mathbb{Z}_q$ of l_1-sensitivity at most Δ and with $|f(D)| \leq L$ for all $f \in F, D \in \mathcal{D}$ and some constant L. Let $p, N = 2^c, M < N, k, c$ be parameters fulfilling the following constraints:*

1. *$k, M \propto \kappa$, $c \propto \log_2(\kappa)$, $p < 1$ constant,*
2. *$N + L < q/2$,*
3. *$p^{-\Delta} \frac{1-p^{2(N-M)}}{1-p^{2(N-M-\Delta)}} \leq e^{\varepsilon}$,*
4. *$\frac{p^{M-\Delta}}{(1-p)(1-p^{2N})} \leq \delta$.*

Then the FDL' mechanism $\mathcal{M}_f(\cdot) := f(\cdot) + z$ with $z \leftarrow FDL'(p, N, M, k)$ is $(\varepsilon, \delta + \mathsf{negl})$-DP and $(\alpha, \beta + \mathsf{negl})$-accurate with $\alpha = \frac{\Delta}{\varepsilon} \log\left(\frac{2}{\beta(1-p)(1-p^{2N})}\right)$ for any $\beta \in (0, 1)$. Further, the CertPM construction (Definition 19) instantiated with the FDL' mechanism via DLapCS is $(\varepsilon, \delta + \mathsf{negl}, \alpha, \beta + \mathsf{negl})$-sCertDP.

Proof. Constraints 2, 3 and 4 guarantee that the FDL mechanism is (ε, δ)-DP and (α, β)-accurate for F (Proposition 2). Constraint 1 gives that the statistical distance between FDL and FDL' is negligible (Proposition 3). Thus the FDL' mechanism is $(\varepsilon, \delta + \mathsf{negl})$-DP and $(\alpha, \beta + \mathsf{negl})$-accurate. Further, constraint 1 also satisfies the requirements of Theorem 6 and thus DLapCS is an sRVCS for FDL'. Finally, by Theorem 3 and Theorem 4, the sCertPM construction when instantiated with DLapCS is indeed $(\alpha, \beta + \mathsf{negl}, \varepsilon, \delta + \mathsf{negl})$-sCertDP. □

Remark 5. With the parameterization in the theorem, if Δ is constant then δ is negligible, meaning we get certified $(\varepsilon, \mathsf{negl})$-DP. This is, of course, not the same as $(\varepsilon, 0)$-DP but, as far as we are aware, there are no known sampling algorithms (with general ε) which give 'pure' $(\varepsilon, 0)$-DP and run in strict polynomial time. The reason for this is that the choice of ε in a pure DP mechanism typically can make the noise distribution impossible to sample exactly in strict polynomial time. Think, for instance, of randomized response [50] with $Ber(1/3)$ noise [23].

E Efficiency Analysis

The summary of the performance of BerCS is found in Table 1 and the analysis is found in Appendix E.1. For any $p \in (0, 1)$ and $k \in \mathbb{N}$, there exists an n such that $|p - n/2^k| \leq 2^{-k}$ and thus (letting k be linear in κ) BerCS can approximate $Ber(p)$ for any p up to negligible statistical distance in linear time.

Table 1. Number of rounds, commitments, verifications and homomorphic operations needed in BerCS. $\oplus$ and $\otimes$ denote homomorphic additions and scalar multiplications, respectively.

Protocol	BerCS
#**Rounds**	$6k-3$
DE #**Commitments**	$7k-5$
PL #**Verifications**	$6k-4$
DE #$\oplus$&$\otimes$	$3k-1$ & $3k-1$
PL #$\oplus$&$\otimes$	$4k-1$ & $7k-4$

The summary of the performance of DLapCS is found in Table 2 and the analysis is found in Appendix E.2. If one requires a total δ of roughly $2^{-\kappa}$ and a constant ε, and we have $k, M \propto \kappa$ and $c \propto \log_2(\kappa)$, then BinCS has all metrics except round number linear in κ, whereas DLapCS has all metrics proportional to $\kappa \log_2(\kappa)$. For our implementation, we use the classical bit-decomposition approach for range-proofs using Σ-protocols. Whilst that method is less efficient than modern alternatives [4,20], the cost of DLapCS is anyhow dominated by the many runs of BerCS. As a feasibility study, we provide an open-sourced Rust implementation of all RVCS's treated in this paper and the benchmarks are found in Appendix E.3

Table 2. Efficiency comparison to BinCS when the number of fair coins used in BinCS is $8\log(2/\delta)/\varepsilon^2$. The performance analysis for BinCS is from BGKW [12], Table 1. $\oplus$ and $\otimes$ denote homomorphic additions and scalar multiplications, respectively.

Protocol	DLapCS	BinCS
#**Rounds**	$12ck-6c+3$	3
DE #**Comm.**	$14ck-10c+3\log_2(2M)+1$	$24\log(2/\delta)/\varepsilon^2$
PL #**Ver.**	$12ck-8c+2\log_2(2M)+1$	$16\log(2/\delta)/\varepsilon^2$
DE #$\oplus$	$6ck+3\log_2(2M)-2$	$24\log(2/\delta)/\varepsilon^2-1$
DE #$\otimes$	$6ck+3\log_2(2M)-2$	$16\log(2/\delta)/\varepsilon^2$
PL #$\oplus$	$8ck+4\log_2(2M)-2$	$32\log(2/\delta)/\varepsilon^2-1$
PL #$\otimes$	$14ck-4c+4\log_2(2M)-2$	$24\log(2/\delta)/\varepsilon^2$

E.1 Analytic Costs of the RVCS for Biased Coins

BerCS(n,k) consists of at most $k-1$ recursive steps and k invocations of CoinCS. Since the case that $n > 2^{k-1}$ is more expensive than the other cases, we upper bound the cost for arbitrary n, k by assuming that in every recursive step, the most expensive branch is run. The case where $n > 2^{k-1}$ involves, except for the invocation of CoinCS and BerCS (for $k-1$) the following:

- 2 commitments sent from DE to PL and one verification by PL.

– One homomorphic subtraction (achieved by one homomorphic scalar multiplication by -1 and one addition) per party.
– One WI-AoK for product relation between commitments.

So, in total, running $\mathsf{BerCS}(n,k)$ requires at most k invocations of CoinCS, $k-1$ invocations of a WI-AoK for product relations, $2(k-1)$ additional commitments, $k-1$ verifications and, per party, $k-1$ homomorphic scalar multiplications and additions.

CoinCS has 3 rounds so each iteration in BerCS is at most 6 rounds. This gives us (at most) $6(k-1)+3 = 6k-3$ rounds of interaction. In CoinCS, the dealer makes 3 commitments and 2 of each homomorphic operation, and the player makes 2 verifications and 3 of each homomorphic operation. [12]

If one uses Pedersen commitments and does the AoK of product relation as in Protocol 9 in [49], then the dealer makes 3 commitments and the player makes 3 verifications. It also involves 3 homomorphic scalar multiplications by the player.

Thus in BerCS, the total number of commitments by the dealer is at most $2(k-1)+k\cdot 2+(k-1)\cdot 3 = 7k-5$. The total number of verifications by the player is at most $k-1+(k-1)\cdot 3+k\cdot 2 = 6k-4$. DE performs in total (at most) $k-1+k\cdot 2 = 3k-1$ of each homomorphic operation. PL performs at most $k-1+3k = 4k-1$ homomorphic additions and $k-1+3(k-1)+3k = 7k-4$ homomorphic multiplications.

E.2 Analytic Costs of the Discrete Laplace sRVCS

DLapCS consists of $2c$ sequential calls to $\mathsf{BerCS}(\cdot,k)$, one range-proof, $2(c-1)$ homomorphic scalar multiplications per party, and $2(c-1)$ homomorphic additions per party and one homomorphic subtraction. If one does the range-proof using the classical bit-decomposition approach (detailed below) and restricts M to be a power of 2, it incurs 3 rounds of interaction, making $3\log_2(2M)+1$ commitments, $2\log_2(2M)+1$ verification, DE performing $3\log_2(2M)$ homomorphic operations of each type and PL performing $4\log_2(2M)$.

Thus there are at most $2c(6k-3)+3 = 12ck-6c+3$ rounds of interaction in DLapCS. There are at most $2c(7k-5)+3\log_2(2M)+1 = 14ck-10c+3\log_2(2M)+1$ commitments and $2c(6k-4)+2\log_2(2M)+1$ verifications. DE makes at most $2c(3k-1)+2(c-1)+3\log_2(2M) = 6ck+3\log_2(2M)-2$ homomorphic additions and $2c(3k-1)+2(c-1)+3\log_2(2M) = 6ck+3\log_2(2M)-2$ homomorphic scalar multiplications. PL makes at most $2c(4k-1)+2(c-1)+4\log_2(2M) = 8ck+4\log_2(2M)-2$ homomorphic additions and $2c(7k-4)+2(c-1)+4\log_2(2M) = 14ck-4c+4\log_2(2M)-2$ homomorphic scalar multiplications.

Range-proof via Bit-decomposition. The AoK range-proof for showing that C_e opens to a value z such that $-M < z < M$ can be done using the following classic method.

1. Shift C_e to be positive by addition to a public commitment to $M-1$ (i.e. DE sends a commitment C_{M-1} of $M-1$ and opens it and both parties set $C'_e \leftarrow C_e \hat{+} C_{M-1}$). DE needs to prove that it can open C'_e to a value in the range $[0, 2M-1]$.
2. Assume that M is a power of 2. DE commits to each bit $b_1, \dots b_{\log_2(2M)}$ such that $C'_e = \hat{\sum}_{i \in [\log_2(2M)]} 2^{i-1} C_{b_i}$ and $r'_e = \hat{\sum}_{i \in [\log_2(2M)]} 2^{i-1} r_{b_i}$.[17]
3. For each b_i, DE and PL run (in parallel) a WI-AoK proving that b_i is binary. This can be done in the same way as in CoinCS.
4. PL accepts if the WI-AoK is accepting and if $C'_e = \hat{\sum} 2^i C_{b_i}$.

This requires 3 rounds of interaction (the AoKs of binary opening are done in parallel, the bit-commitments are sent with the first round of the AoK and the opened commitment can be computed locally, using public randomness). DE needs to send $1 + 3\log_2(2M)$ commitments (one at the start, one per each bit, and 2 per Σ-protocol.) PL performs $1 + 2\log_2(2M)$ verifications (one at the start and 2 per Σ-protocol). DE performs $3\log_2(2M)$ of each operation (one at the start, $\log_2(2M) - 1$ in the sums and 2 per Σ-protocol). Similarly, PL performs $4\log_2(2M)$ of each homomorphic operation.

The binomial RVCS For the binomial mechanism, it is sufficient to use $N_b = 8\log(2/\delta)/\varepsilon^2$ fair coins, given that ε is small enough [2].[18] Since BinCS consists of only N_b executions of CoinCS (which can be done in parallel) followed by $N_b - 1$ homomorphic additions, only three rounds of interaction are needed. There are $3N_b = 24\log(2/\delta)/\varepsilon^2$ commitments and $2N_b = 16\log(2/\delta)/\varepsilon^2$ verifications. DE does $2N_b + N_b - 1 = 24\log(2/\delta)/\varepsilon^2 - 1$ homomorphic additions and $2N_b = 16\log(2/\delta)/\varepsilon^2$ homomorphic scalar multiplications. PL does $3N_b + N_b - 1 = 32\log(2/\delta)/\varepsilon^2 - 1$ homomorphic additions and $3N_b = 24\log(2/\delta)/\varepsilon^2$ homomorphic scalar multiplications.

E.3 Experimental benchmarks

We implement BerCS and DLapCS, as well as their necessary components (such as CoinCS), and this also gives an implementation of BinCS[19]. As a base, we build on Arkworks [26]. We extend their implementation of Pedersen commitments with that of homomorphic operations and implement the Σ-protocols for CoinCS, $\mathsf{WI\text{-}AoK}_{mult}$, and range proofs. We run our benchmarks on a 12th Gen Intel(R) Core(TM) i7-1260P with 2.1GHz and 32GB RAM. While our implementation can be configured with any supported elliptic curve in Arkworks, we run our

[17] As in additive secret-sharing, we can set each r_i uniform except the last one which is set to so that the sum is correct.

[18] Otherwise $N_b = 64\log(2/\delta)/\varepsilon^2$ suffices [33], so the essence of our comparison is not contingent on using small ε.

[19] All our implementations, as well as raw benchmark outputs, can be found at: https://github.com/Fable95/certified-laplace

benchmarks with Curve25519, commonly used in Elliptic-Curve Diffie-Hellman (ECDH), due to its fast multi-scalar multiplication [15].

We test the implementation on the parameters $\Delta = 1, \varepsilon = 1, k = \kappa = 80, M = 2^6, N = 2^c = 2^7$ and this allows $p = 1/e$. We get the following calculations of δ: $\frac{p^{M-\Delta}}{(1-p)(1-p^{2N})} = \frac{e^{(-M+\Delta)}}{(1-e^{-1})(1-e^{-2N})} < \frac{e^{-63}}{(1-1/e)(1-e^{-256})} < \frac{e^{-63}}{1/e \cdot 1/e} = e^{-61} < 2^{-88} =: \delta$.

Further,

$$\begin{aligned}
&\delta + (e^{\varepsilon} + 1)(c \cdot 2^{-k+1} + 2p^{M+1}/((1+p)(1-p^N)^2)) \\
&\leq 2^{-88} + 4(7 \cdot 2^{-79} + 2e^{-63}/((1+1/e)(1-e^{-128})^2) \\
&\leq 2^{-88} + 2^{-71} \leq 2^{-70}.
\end{aligned}$$

So setting the total δ-parameter in DLapCS to $\delta_{tot} = 2^{-70}$ suffices.

In Table 3, we present the concrete breakdown of runtimes for DLapCS. The numbers for CoinCS represent the total runtime of generating all the fair coins. The numbers in BerCS+compose, are the total runtime of all remaining parts of the BerCS invocations, as well as the subsequent composition of bits and subtraction of composed values. Within BerCS, we include the Σ-protocol for proving all product relations and the recursive algorithm based on the results. Finally, WI-AoK$_{decomp}$ refers to the cost of the range proof. As claimed in the theoretical analysis, the efficiency impact of the range proof is very minor.

Table 3. Breakdown of runtime cost of individual components in the DLapCS implementation. All times are given in ms. Parameters are set as $k = 80, c = 7, M = 2^6$.

Protocol	Dealer	Player
CoinCS	498.5	439.75
BerCS+compose	603.4	595.9
WI-AoK$_{decomp}$	2.48	2.01
Total	1104.4	1037.66

We now give an in-depth overview of the concrete performance of the subprocedures. In Table 4, we show the cost of proving a commitment to hold a binary value. While the runtime is fast at roughly $300\mu s$ per bit, it still has a significant impact on overall runtime as it is used both in BerCS and the range-check. We implement CoinCS as presented in [12].

For product relations, similar to BGKW, we base our implementation on [44]. We deviate in that we move the challenge-related computation from the verifier/player into the final verification step. This helps us break down computational costs better. The results in Table 5 show that this protocol presents the main computational burden for the verifier/player.

Finally, we get to the concrete breakdown of BerCS for $k = 80$ and $n = 1$. These parameters will lead to 80 required bits and 79 product relations. The first rows in Table 6 represent the combined cost of all invocations of CoinCS. Then,

Table 4. Breakdown of runtime cost of CoinCS. Runtime is given in μs.

Protocol	Dealer	Player
Commit	184.3	
Response	0.02	
Verify		248.06
CoinCS	184.3	248.06

Table 5. Breakdown of runtime cost of WI-AoK$_{mult}$. Runtime is given in μs.

Protocol	Dealer	Player
Commit	210.7	
Response	0.08	
Verify		410.2
WI-AoK$_{mult}$	210.8	410.2

the cost of performing the exclusive-or for all committed bits. In commit, we see the combined cost of performing all product commitments and computing the final output commitment of BerCS. Product response and verify product, present the cost of the remainder of the product relation proof. In Compute Homomorphic, we see the cost of evaluating the recursive protocol at the player side, while Verify Final is the cost of verifying the last commitment.

Table 6. Breakdown of runtime cost of BerCS. Runtime is given in ms.

Protocol	Dealer	Player
Bit Commit	27.7	
Bit Response	0.002	
Bit Verify		22.1
Bit Flip		17.2
Commit	53.6	
Product Response	0.007	
Verify Product		35.8
Compute Homomorphic		6.2
Verify Final		0.05
Total	81.3	81.4