# Practical Two-party Computational Differential Privacy with Active Security

Fredrik Meisingseth[1]*, Christian Rechberger[1], and Fabian Schmid[1]

Graz University of Technology, Graz, Austria
firstname.lastname@iaik.tugraz.at

**Abstract.** In this work we revisit the problem of using general-purpose MPC schemes to emulate the trusted dataholder in central differential privacy, to achieve same accuracy but without the need to trust one single dataholder. In particular, we consider the two-party model of having two computational parties (or dataholders) each with their own dataset wishing to compute a canonical differentially private mechanism on their combined data and to do so with active security. We start by remarking that available definitions of computational DP (CDP) for protocols are somewhat ill-suited for such a use-case, due to them using formalisms that either are much weaker than one can typically get for MPC protocols, or they are too strict in the sense that they need significant adjustment in order to be realisable by using common DP and MPC techniques. With this in mind we propose a new version of simulation-based CDP, called SIM$^*$-CDP, specifically geared towards being easy to use for MPC practitioners and more closely capture guarantees granted by using state-of-the-art MPC schemes to compute standard DP mechanism. We demonstrate the merit of the SIM$^*$-CDP definition by showing how to satsify it by use of an available distributed protocol for sampling truncated geometric noise. Further, we use the protocol to compute two-party inner products with computational DP and with similar levels of accuracy as in the central model, being the first to do so. Finally, we provide an open-sourced implementation and benchmark its practical performance.

**Keywords:** Differential privacy, Multiparty computation, UC-security

## 1 Introduction

The study of differential privacy in various distributed settings has given rise to a plethora of new definitions of DP, such as DP in the *local model (LDP)* [47], the *shuffle model* [7, 17] and definitions with a computationally bounded adversary, giving guarantees of *computational DP (CDP)* [28, 5, 58]. Each of the definitions are subject to their own restrictions in the adversarial model and in the accuracy that can be achieved within them. For instance is it well studied that LDP, which is a computationally efficient model with very few trust assumptions,

[*] Parts of this work was performed whilst at Know-Center, Graz, Austria.

must add much more noise than the standard central model of DP [47, 35, 16, 5]. One popular remark is that one can use general-purpose *multiparty computation (MPC)* to *emulate* a trusted central dataholder and thus one may get the accuracy that is possible in the central model of DP without having to trust a central computational party [31, 17]. The troubles in realising this idea, which we can call *generic emulation of the dataholder (GED)*, are firstly that one must accept the, potentially, large computational costs of MPC and secondly that it is not necessarily clear how one should define DP in this new distributed and computational setting. In order to avoid or reduce the computational costs of using MPC, up until now, most of the works in this area have opted for considering passive adversaries [5, 33, 65], only allowing aggregate functions [20, 48] and/or requiring honest majorities [28]. In this work, we focus on the case of two parties, active (static) corruptions, and require efficient protocols[1] for non-aggregate functionalities that achieve the same accuracy as in the central model.

**Existing CDP notions.** In order to design practical protocols for GED, we would want a DP notion that is directly compatible with the security notions of state-of-the-art MPC schemes and that allows the emulated dataholder to compute common DP mechanisms. Many such mechanisms, such as the Laplace [30], geometric [40], Gaussian [31] and discrete Gaussian [14] mechanisms are not computable exactly in strict probabilistic polynomial time (PPT) on a finite computer.[2] This means that, since general-purpose MPC only allow PPT computable functionalities, the used definition needs to allow either that the protocol does not exactly emulate the dataholder (imperfect correctness) or that the emulated dataholder does not exactly compute the DP mechanism, or both. Further, since we consider the case of two parties and active corruptions, for which information-theoretic general-purpose MPC is impossible [19, 41, 36], the only candidates of a suitable DP definition are the CDP notions introduced in [58, 5]. Since we will refer to it recurrently, let us call the paper [58] *MPRV*, after its authors.

Of the multiparty CDP definitions in MPRV, IND-CDP and SIM-CDP (Definition 6 in MPRV, also reiterated in Appendix A) are defined such that they allow inefficient protocols, precisely in order to allow them to compute inefficient (non-PPT) DP mechanisms. However, there is no separation between the protocol (which we will want to be efficient) and the ideal DP mechanism (which we will want to allow being inefficient), which makes using those definitions an unnatural fit for the purpose of GED. Further, their simple structure and security models means that whilst they are convenient to use for analysing specific protocols [48, 62, 33] and for deriving theoretical bounds [56, 43, 45, 39, 8], they do not harness the strong security guarantees oftentimes available for MPC

[1] In particular, we require that the protocols are computable in strict polynomial time in a finite computational model, as suggested in [3].

[2] This is due to the need to represent probabilities that are not multiples of $2^{-l(\kappa)}$ for any polynomial $l$, as noted in, for instance, [14].

schemes and thus potentially result in unnecessarily weak claims for the case of GED. Similar arguments hold with regards to the CDP notion introduced in [5] due to its similarity to IND-CDP.

The third multiparty CDP notion in MPRV, SIM$^+$-CDP, on the other hand uses the ideal/real world paradigm for security, allows the ideal functionality to be inefficient but the protocol not and the CDP definition inherits security properties of a standard notion of security in MPC [41]. Still, SIM$^+$-CDP requires that the emulation of the dataholder has perfect correctness, which implies that the definition *is not satisfiable* for non-PPT DP mechanisms, and one would need to instantiate it with a finite version of them, for instance using the mechanisms introduced in [3]. Whereas this is not necessarily an unsatisfying approach, it does mean in some sense a less direct realisation of GED, since the intuition is still to, say, 'use MPC to run the geometric mechanism' and thus it is appealing to have a definition that can also be fulfilled when the ideal functionality is inefficient. Further, SIM$^+$-CDP is restricted to the functionality of *secure function evaluation (SFE)*, whereas IND-CDP and SIM-CDP are not, and there are other natural cryptographic functionalities for which one might want to apply CDP, such as scenarios where SFE is done whilst allowing differentially private leakage throughout the protocol [55, 44]. All in all, it therefore seems a more appealing approach to define a CDP notion in the ideal/real paradigm with another security notion than the one used in SIM$^+$-CDP. In particular, using the more expressive UC-security framework [13] as security notion, one can simultaneously solve the issue of requiring perfect correctness and open up for more ideal functionalities. Finally, using UC security gives stronger security guarantees with respect to composition of protocols and is it also the security notion used by some of the most popular schemes for MPC, such as [23, 21, 38, 1].

**A new definition.** With this motivation, we propose a new version of SIM$^+$-CDP, which we call SIM$^*$-CDP. We underscore that the merit of our new definition is not that it necessarily allows studying new scenarios altogether or is to be preferred over previous definitions in all cases, indeed there are many relevant cryptographic tasks for which a UC security proof is missing or for which it is not the most desirable framework to use (as is, for instance, argued in [24] for the case of *verifiable distributed aggregate functions (VDAFs)*). Rather the merit is that for settings where UC secure protocols are readily available, then we have a formulation that takes advantage of that to give results that are both stronger and easier to obtain. Further, we also propose a generalised definition of simulation-based CDP via the ideal/real paradigm, which we call SIM$^\circ$-CDP of which both SIM$^+$-CDP and SIM$^*$-CDP can be seen as instantiations. To demonstrate the advantages of SIM$^*$-CDP over previous definitions, we give a generic protocol for satisfying SIM$^*$-CDP for the ideal functionality computing the truncated geometric mechanism in SFE. Further, we implement the protocol, use it to compute differentially private inner-products and benchmark the implementation, hence showing its practical performance. The treatment of (non-binary) integer inner-products might be of independent interest, perhaps primarily due to our considerations relating to the fact that the function sensitivity of the inner-product is dependent on the data of the corrupted party, thus creating a need for input validation.

**Contributions:**

- We identify aspects of existing CDP definitions that make them cumbersome to work with in the context of generic emulation of a central trusted data-holder that computes an inefficient DP mechanism. With these difficulties in mind we present a new version of SIM$^+$-CDP, which we call SIM$^*$-CDP$\dot{\text{W}}$e also propose a generalised version of SIM$^+$-CDP and SIM$^*$-CDP, which we call SIM$^\circ$-CDP (Section 3).
- We demonstrate the usability of the SIM$^*$-CDP definition by showing how it can be achieved for the truncated geometric mechanism by proving that a slightly adapted version of the efficient MPC protocol by [33] for sampling geometric noise satisfies our definition (Sections 4 and 5).
- We use the protocol to compute differentially private two-party inner-products with security against active adversaries, to the best of our knowledge being the first to do so with accuracy equal to that in the central model, and provide an open-sourced implementation[3]. Ours is the first implementation of the noise generation protocol of [33]. We provide benchmarks of the implementation and thereby show that it is efficient in practice (Section 6).

**Related works.** The first work that aims to emulate a central trusted party for DP by use of MPC is *Our data, ourselves* [28], where they propose a protocol for computing sums with security against active adversaries corrupting less than a third of the parties. As a part of this protocol, they propose a method for distributed noise generation. Following [28], other works have also proposed noise sampling protocols for DP in an MPC setting [2, 15, 33, 66] and perhaps the work most related to ours is EIKN [33]. EIKN gives an efficient MPC protocol for sampling from an approximate truncated geometric distribution, which we use in this work. Their results however only hold for passive corruptions and honest majorities.[4] In a recent preprint [49], the authors provide an efficient noise sampling protocol for passive corruptions and dishonest majorities. The authors of [49] note in passing that their protocols can easily be made secure against active adversaries by implementing them in a framework with active security, such as MP-SDPZ [50], but make no note of the type of CDP this could result in. In that sense, our proposed SIM$^*$-CDP definition offers an immediate answer to that.[5] The work of [15] propose a method for performing Bernoulli trials that

[3] https://extgit.iaik.tugraz.at/krypto/geometric_sampler

[4] In a follow-up work [34], an extension to active corruptions was given but those results require less than a third of the parties to be corrupted.

[5] Another related paper is [2]. It is the only published work of which we are aware that claims to provide a method for achieving CDP in the two-party case in the presence

is asymptotically superior to the one we use (Section 4) however their method relies on implementing oblivious data structures hence making it unsuitable for direct combination with the secret-sharing based MPC schemes that we use.

Another line of work that is of relevance to ours due to it dealing with combining definitions of security for MPC schemes and DP is the string of papers considering MPC with differentially private leakage [46, 55, 44], where the idea is to improve the efficiency of an MPC protocol by allowing the protocol execution itself (not the result) to leak some extra information, but to restrict this leakage to be differentially private. Particularly relevant for our work is the work of [44] where they combine DP with UC security and do so partially with the purpose of introducing an ideal functionality that is not capturable in the security framework of [41]. Our new definition (Section 3) thus is significantly influenced by that of [44], as can be seen by the syntactic resemblance.

Finally, we note that the large line of work on *differentially private data collection*, arguably centered around protocols related to the non-DP aggregation protocol Prio [20, 61, 6, 48], is conceptually relevant due to being an important topic of distributed CDP protocols. On the practical side, however, most of the techniques used there (which yield very efficient and scalable systems) are not applicable to our work. This includes techniques for distributed noise generation via infinitely divisible distributions[6], since it restricts the system to only compute aggregate functions, operate under the assumption of passive corruptions or have the data subjects (also called *clients*) encode their data points with respect to the function that is then to be computed. As expanded on in the next section, we focus primarily on *joint computation of a DP mechanism* where each computational party has a specific part of the input database to the mechanism in the clear, instead of having secret shares of the input as is typically the case for private data collection.

# 2 Preliminaries

## 2.1 Differential privacy

The notion of *differential privacy (DP)* [30, 27] considers a probabilistic function, algorithm, or *mechanism*, that maps *databases*, i.e. sets of elements from some data universe $\chi$, to some output range $R$. We think of databases as ordered sets of some fixed (public) size $N'$, and thus a database $D$ is an element of $\chi^{N'}$. We say that two databases $D, D'$ are *adjacent* if they differ in at most one element, i.e. there exists at most one index $i \in \{1, ..., N'\}$ such that $D_i \neq D'_i$. We recall the standard definition of DP (reformulation of [27]) in Definition 1.

of active adversaries. Upon consideration, it is clear that the method they propose does not fulfill the notion of CDP that they claim to achieve (SIM$^{+}$-CDP) and this is due to their mechanism (Laplace) not being PPT computable.

[6] See, for instance, Section 2.2.5 in the phd thesis of Böhler [10].

**Definition 1 ($\varepsilon$-DP [30, 27]).** *A probabilistic function $\mathcal{M} : \chi^{N'} \to R$ is $\varepsilon$-differentially private if for all pairs $(D, D')$ of adjacent databases in $\chi^{N'}$ and all subsets $S$ of $R$,*

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(D') \in S), \tag{1}$$

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

We refer to DP as defined above as *pure DP*. We primarily use the common relaxation of $(\varepsilon, \delta)$-DP, called *approximate DP*[7], which is the same as the definition above with the sole difference that the inequality also includes the additive term $\delta$. DP is typically studied in what is called the *central model*, of which an illustration can be found in Figure 1. In the central model, the database is simply a set of rows, each of which consists of information about one individual, called a *data subject*. These data subjects send their data to a trusted *data-holder* (without noise) that then computes a mechanism on the accumulated data and then releases the result to an untrusted *data analyst*. In this work, we rather consider DP in a distributed model, namely the *two-party DP* model, as is for instance [58, 56], where each data subject holds two database rows $(x_i, y_i)$, each of which is sent to one of two computational parties (or servers) that then stores their respective row into their database ($\mathbf{x}$ and $\mathbf{y}$ respectively) in the clear. Then these two computational parties together wish to compute the query $f$ on the concatenation of their databases $D := \mathbf{x}||\mathbf{y}$, both learning the result, and they wish to do this in a differentially private manner with respect to their database. An illustration of this model can be seen in Figure 2. We note that the two-party model is slightly but significantly different from the *two-server/multi-server models* [6, 18], also called the *multi-central model* [64], primarily in that those models do not allow any server to have any part of the input dataset in the clear. This difference is of practical relevance because it means the models are suitable for different scenarios. The two-party model is mostly meant for *joint computation* between two entities each holding their own dataset (which may have been collected over time and without respect to the function evaluation in question) whereas the two-server model is rather tailored towards *data collection*, where one entity or more entities are collecting the data specifically for the purpose of performing the computation but wish to do so in a way that they never see any part of the dataset in the clear.

When discussing DP mechanisms, it is critical to consider the usefulness of the mechanism for approximating the query function $f$. We do this by using the following notion of usefulness, which is a reformulation of the notion of usefulness in MPRV [58] to consider probabilistic functions rather than interactive protocol ensembles.

[7] See, for instance, Definition 7.1.4 in [65].

**Definition 2 (Usefulness).** *Let $\{f_\kappa : \mathcal{D}_\kappa \to R_\kappa\}_{\kappa\in\mathbb{N}}$ and $\{\hat{f}_\kappa : \mathcal{D}_\kappa \to \hat{R}_\kappa\}_{\kappa\in\mathbb{N}}$ be ensembles of probabilistic functions. We say that $\{\hat{f}_\kappa\}$ provides $\underline{\nu\text{-usefulness with respect to the predicate } P}$ for $\{f_\kappa\}$ if for every sufficiently large $\kappa$ and for every $D \in \mathcal{D}_\kappa$ is holds that $\mathbb{P}(P(\hat{f}_\kappa(D), f_\kappa(D)) = 0) \leq \nu(\kappa)$, with the probability being over the internal randomness of both $\hat{f}_\kappa$ and $f_\kappa$.*

A specific predicate we will consider is that which induces the notion of $(s, \nu)-$*additive-usefulness*. That is, $P(a, b) = 1$ iff $|a - b| \leq s$. In particular, $s$ can be a function of $\nu(\kappa)$.

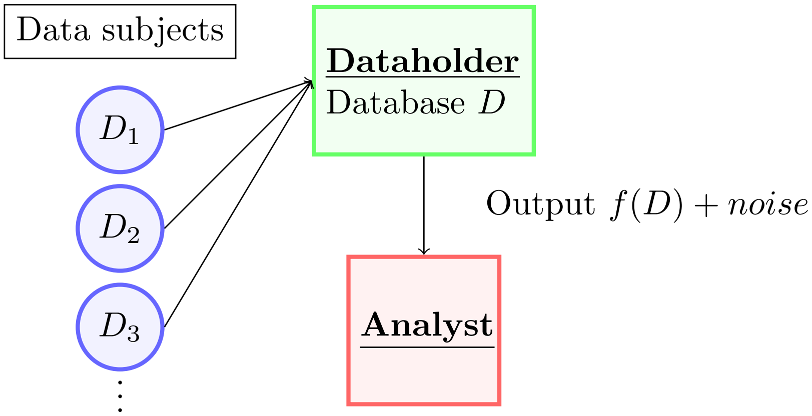

Fig. 1: In the *central model*, the data subjects trust the data holder with their data ($D_i$) but wish to keep it secret from an (possibly adversarial) analyst learning the (possibly noisy) function evaluation.

### 2.2 Mixed binary-arithmetic MPC schemes

In our definitions, we rely on MPC schemes with active security. In particular, we work with MPC protocols with restricted computation domain, either in $\mathbb{F}_p$ for arithmetic or $\mathbb{F}_{2^k}$ for binary circuits. For a discussion of active security in these schemes, we refer to Appendix D. In general, MPC schemes in $\mathbb{F}_p$ provide fast algorithms for addition and multiplication. In contrast, in $\mathbb{F}_{2^k}$, comparisons, bit-wise operations, and non-linear functions can be evaluated cheaply. However, storing larger integers results in substantial overhead, and evaluating arithmetic circuits in the binary domain incurs costs depending on the encoded values' bit size.

Several works have proposed solutions to convert shares between computation domains. First, in ABY [25], the authors propose a semi-honest two-party MPC

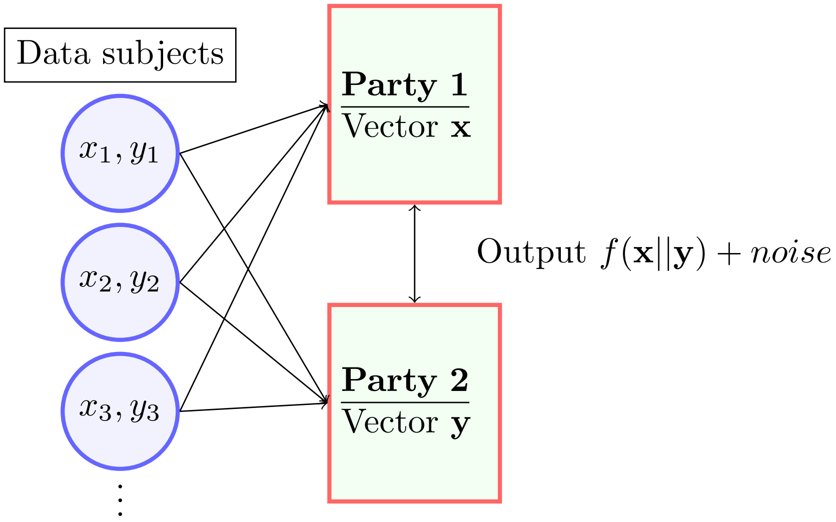

Fig. 2: In the *two-party model*, the data subjects trust two different data holders, which we call *parties*, with a different part of their data, but not with the part of the data that they send to the other data holder. Both parties then learns the noisy function evaluation. Thus, in a sense, each party plays both the role of a data holder and a data analyst.

scheme that allows switching between the binary, arithmetic, and garbled circuit domains (Garbled Circuits allow computation of binary circuits with low communication rounds). More recently, Rotaru and Wood introduced *doubly-authenticated bits* [60] and an efficient procedure to securely sample secret bits in the arithmetic and binary domain in malicious settings. Given the shares of an unknown random bit $(\llbracket b \rrbracket_2, \llbracket b \rrbracket_p)$ we can transfer shared bits from the binary to the arithmetic domain by computing the mask $m \leftarrow \mathsf{Reconstruct}(\llbracket x \rrbracket_2 \oplus \llbracket b \rrbracket_2)$ and setting $\llbracket x \rrbracket_p \leftarrow m + \llbracket b \rrbracket_p - 2 \cdot m \llbracket b \rrbracket_p$. Similarly, converting from arithmetic to binary masks the value by addition and evaluates subtraction in the binary domain. The conversion from the arithmetic to the binary domain gets more expensive, depending on the field size. Subsequent work introduced *extended doubly-authenticated bits (eda-bits)* [37], where masking values are shared along with their binary decomposition in the respective domains. The eda-bits represent an improvement in efficiency when converting larger values, and [37] presents dedicated protocols to speed up comparisons in $\mathbb{F}_p$.

# 3 A new version of simulation-based CDP in the ideal/real paradigm

## 3.1 Wishes for a definition

As noted in the introduction, we consider CDP notions with respect to their strength and ease-of-use in combining *general-purpose* MPC schemes, i.e. schemes that can be used to compute any (efficiently computable) function, and standard DP mechanisms. In other words, the optimal CDP definition would be one that is well-suited for the practice of slotting a common DP mechanism into a state-of-the art MPC protocol, which is the core idea of GED. In particular, ”well-suited” here means that:

- *Ease of use* - The satisfaction of the definition follows directly, or via a very simple process, from the security properties of the MPC scheme and the DP mechanism.
- *Strength* - The definition takes advantage of the strong results for both the involved DP mechanism and MPC scheme, meaning that it is not significantly weaker than directly granted by the constituent parts.
- *Generality* - The definition is able to be realised for a large set of DP mechanisms, and by use of a large part of current state-of-the-art MPC schemes.

Of course the first two properties can, for a specific use-case, be attained by using the ad hoc approach of simply describing the MPC scheme and DP mechanism in use, thus causing implicit CDP definitions. The problem of that approach is that it does not offer generality, meaning partly that comparing different such implicit definitions qualitatively becomes challenging (due to the dependence on all aspects of the specific mechanism and MPC scheme) and partly that the definition in itself as a source of confidence in the system is diminished due to the increased inconvenience in using the definition as the object of theoretic study. That is, whilst we want a definition that is essentially directly satisfied by simply choosing a DP mechanism and an MPC scheme, it is not desirable to have those choices as a part of the definition, because that would remove the fruitful separation between the definition itself as an object of study and the methodology we use to fulfill it. There are three main types of definitions for CDP in the literature:

- *Indistinguishability-based* definitions such as IND-CDP in MPRV[58] and the very similar CDP notion in the full version of [5]. The core idea of these definitions is that all efficient functions taking as input the view of the corrupted parties should be $(\varepsilon, \delta)$-DP with a negligible $\delta$.
- *Simulation-based definitions not using the ideal/real paradigm*, of which SIM-CDP in MPRV is the only definition that we are aware of. The idea here is rather that the view of the adversary is computationally indistinguishable from the output of a (possibly inefficient) simulator with access to the inputs of both parties.

- *Simulation-based definitions using the ideal/real paradigm*, represented by the SIM$^+$-CDP definition in MPRV, where CDP is defined as that the protocol in question securely computes an ideal functionality that is $\varepsilon$-DP.

With respect to our wishlist of properties above, it should be immediate that the first two classes of definitions primarily lack strength, in the sense that if one computes a DP mechanism in a state-of-the-art MPC scheme then already has stronger security/privacy properties than are required by the definition. One example of this is that, as proven in the full version of MPRV, SIM$^+$-CDP implies SIM-CDP, which means that if one realises GED in the sense required by SIM$^+$-CDP, then using SIM-CDP to describe the CDP properties is to under-sell the protocol in question. It should also be clear from the list above that simulation-based CDP by use of the ideal/real paradigm seems to be a very promising definitional approach, due to the intimate connection to the dominant flavor of security definitions in general-purpose MPC. In the following, we consider SIM$^+$-CDP more carefully and remark on details in the definition that makes it seem to not fulfill our wishlist entirely, primarily with regards to the generality aspect.

### 3.2 The original SIM$^+$-CDP definition

As noted, SIM$^+$-CDP is a more direct realisation of GED than IND-CDP and SIM-CDP, partly due to them not having the same clear separation between functionality (mechanism) emulation and the DP properties of the mechanism itself. The definition of SIM$^+$-CDP in MPRV [58] is the following[8].

**Definition 3 (SIM$^+$-CDP, Definition 8 in MPRV [58]).** *An interactive protocol ensemble $\{\langle g^1_\kappa(\cdot), g^2_\kappa(\cdot)\rangle\}_{\kappa\in\mathbb{N}}$ is a $(s,\nu)\varepsilon_\kappa$-SIM$^+$-CDP private two-party computation protocol for $f = (f_1, f_2)$ with respect to the predicate $P$ if there exists an $\varepsilon_\kappa$-DP randomized mechanism $\hat{f} = (\hat{f}_1, \hat{f}_2)$ such that*

- *Mechanism $\hat{f}$ provides $(s,\nu)$-usefulness for $f$ with respect to the predicate $P$.*
- *The protocol ensemble is a secure two-party computation protocol ensemble for the randomized functionality $\hat{f}$ as per the "ideal/real"-style definition of secure two-party computation (see full version of MPRV).*

For more details on the "ideal/real"-paradigm, the reader is in MPRV referred to the standard texts [12, 41]. The full version of MPRV [58][9] provides an exact definition of the used notion of secure two-party computation. To the best of our understanding, the definition that they use is that of [41], with the sole adjustments that the simulator is not required to be efficient. In particular,

[8] For definitions of interactive functions, we refer to [42], and of protocol ensembles to MPRV. For the notion of usefulness with respect to predicates, we refer to the full version of MPRV. Note also that their notion of usefulness is slightly different from the one we use although this is not of any real relevance to the present work.

[9] The full version is available from the authors.

the definition used in MPRV requires *efficiency*, i.e. that each of the parties in the protocol can be computed by a PPT interactive Turing machine (ITM), and *perfect correctness*, meaning that the output of the protocol in an honest execution is identically distributed to $\hat{f}$. These details in the used notion of secure computation leads to that many canonical DP mechanisms cannot be directly slotted into the SIM$^+$-CDP definition, as seen in the following example.

**Infeasibility of GED with the Laplace mechanism in SIM$^+$-CDP.** Consider using SIM$^+$-CDP to describe realising GED with the Laplace mechanism. The main question is whether there exists an efficient protocol that can realise the Laplace mechanism in SIM$^+$-CDP. Unfortunately, there is not, and the problem lie in combining the efficiency requirement of the protocol and the requirement for perfect correctness. The support of the Laplace mechanism is the reals, and thus the output cannot even be written in strict finite time. Thus, the two requirements above directly imply that any mechanism in the SIM$^+$-CDP definition must have a finite support. Further, even the (arguably) most Laplace-like such distribution, the geometric distribution [40] truncated to the output domain, cannot be realised in SIM$^+$-CDP in general, since it requires sampling probabilities that are not multiples of $2^{-poly(\kappa)}$. This means that in order to realise GED with distributions that cannot be sampled exactly in strict polynomial time (as is the case for the Laplace, geometric, Gaussian, discrete Gaussian distributions and truncated versions of them), there needs to be some slack introduced. This could be, for instance, in the shape of allowing a small statistical distance between the output of the ideal functionality and that of the protocol (relaxing correctness) or relaxing the demand for strict polynomial time to expected polynomial time, as is argued in [14].

### 3.3 Our new definition, SIM$^*$-CDP.

We now propose a new version of SIM$^+$-CDP, which we call SIM$^*$-CDP and then discuss its relationship to previous definition further.

**Definition 4 ($(\varepsilon_\kappa, \delta_\kappa)$–SIM$^*$-CDP).** *The two-party protocol $\pi$ is $\underline{(\varepsilon_\kappa, \delta_\kappa)\text{-SIM}^*\text{-CDP}}$ for the ideal functionality $\mathcal{F}$ and a given adjacency notion if $\pi$ UC-realises $\mathcal{F}$ and for all ideal-world adversaries $\mathcal{S}$, the view of $\mathcal{S}$ is $(\varepsilon_\kappa, \delta_\kappa)$-DP with respect to the adjacency notion.*

SIM$^*$-CDP is essentially the same as SIM$^+$-CDP but with the following main changes:

- *UC-security is used as security notion.*
- *The ideal functionality is unspecified (and can be reactive).*
- *Correctness is computational rather than perfect.*
- *The ideal-world adversary (simulator) must be efficient (strict PPT).*

Other minor changes are:

- Using $(\varepsilon, \delta)$-DP instead of $\varepsilon$-DP as the core notion of DP.
- The requirement of usefulness is removed from the CDP definition.

We now expand on the motivation behind these changes.

**Using UC-security.** Although the stand-alone security framework used in SIM$^{+}$-CDP (in a slightly tweaked form) is heavily used, in the last two decades the security analyses of many popular schemes have taken place in the more expressive UC framework [13]. The main merit of this framework is that the security can be proven to be preserved under arbitrary composition of protocols, leading to a stronger notion of security and an increased convenience when proving the security of composed protocols. Thus, using UC security in the CDP notion is natural for cases where this (stronger) type of security is already achieved by the MPC scheme one intends to use. Further, as we will see below, this change in security framework also directly leads to many other benefits.

**An unspecified ideal functionality.** The ideal functionality used in SIM$^{*}$-CDP is explicitly that of the parties jointly computing a differentially private function (with abort). With regards to capturing what it means for a protocol to be computationally DP, this is a significant restriction as compared to IND-CDP and SIM-CDP, where the protocols are defined to be CDP based on properties of the view of the adversary, regardless what the computational task of the protocol is. In particular, both IND-CDP and SIM-CDP allows direct modeling of reactive tasks, and as such our new definition arguably lies closer to those definitions conceptually than SIM$^{+}$-CDP does, in the sense that it remains open to more cryptographic tasks than non-interaction function evaluation being considered CDP. On the practical side, one very relevant reactive functionality is that of SFE with differentially private leakage as in, for instance, [44]. More details about the setting of SFE with DP leakage in Section 3.4.

**Computational correctness.** Another positive consequence of using the UC security framework is that the correctness of the protocol is explicitly computational rather than perfect. This resolves the complications of emulating in polynomial time a trusted dataholder that computes a function which is not computable in polynomial time since the protocol need not generate the exact same output distribution as the dataholder but only something that is computationally indistinguishable from it. We will see in coming sections that one can efficiently sample a distribution that is statistically indistinguishable from a geometric distribution.

**Efficient simulators.** As one main goal of our new definition is to have it align closely to common practice in MPC, we choose to require efficient simulation. Whereas this does make fulfilling the definition harder, it also makes the definition stronger.

**Using approximate DP.** The main motivation for using approximate DP instead of pure DP is to allow for popular mechanisms, like the discrete Gaussian and truncated discrete Laplace, that cannot give pure DP. One interesting question here is whether the relaxation from pure to approximate DP is still needed when the final DP notion we are aiming for is anyhow computational. Our argument for introducing this extra relaxation is that whilst mechanism that in essence have the $\delta$ term introduced due to lack of precision in the sampling of a distribution that would give pure DP (as with the truncated discrete Laplace) could have this inexactness be handled by slack in other places of the CDP notion (such as the correctness slack), this is not the case for mechanisms that are "inherently" not giving pure DP, as is the case for all versions of the Gaussian mechanism.

Another more fundamental question is whether motivational properties of using approximate DP in the first place, such as arriving at more powerful composition results, also can be done in computational DP where the core DP notion is pure DP. That is, given a fixed CDP notion, *can advanced composition theorems such as those for approximate DP be obtained for CDP whilst having pure DP as the core DP notion?* We leave exploring this for future work.

**Not including usefulness in the definition.** A final minor difference between SIM$^*$-CDP and SIM$^+$-CDP is that we choose not to include the requirement for usefulness in the definition of CDP itself. This is done primarily to more closely correspond to how the matter of usefulness is handled for IND-CDP and SIM-CDP in MPRV, namely that the CDP definition is agnostic to the notion of usefulness (Definition 6 in MPRV [58]) and that usefulness is then added later (Definition 7 in MPRV). Another advantage of not having the usefulness as a part of the CDP definition is that one can choose to consider the usefulness simply of the ideal functionality (as is done in SIM$^+$-CDP) or to consider the usefulness of the protocol directly and then take, for instance, failure probabilities of the protocol into account.

To round this subsection off, we re-iterate the standard ideal functionality for SFE with abort, see Figure 3. In Section 5 we propose a protocol for realising this ideal functionality with the geometric mechanism as the functions $f_1$ and $f_2$ and prove it is SIM$^*$-CDP in the presence of active corruptions.

### 3.4 On computing a mechanism with DP leakage in SIM$^*$-CDP

As noted shortly in the previous subsection, one cryptographic task that SIM$^*$-CDP can handle but SIM$^+$-CDP cannot is that of computing a differentially private mechanism whilst allowing the adversary to receive leakage throughout the protocol, as long as that leakage is DP, in particular when some leakage occur before the corrupted party choose their input. Joint computation of functions whilst allowing DP leakage has been studied in a few different setting with

**Functionality $\mathcal{F}^{f}_{SFE}$**

**Parameters:**

- A function $f = (f_1, f_2) : (\{0,1\}^*)^2 \rightarrow (\{0,1\}^*)^2$.

**No corruptions:**

- Upon $\mathbf{x}_1$ from $P_1$ and $\mathbf{x}_2$ from $P_2$, deliver $f_1(\mathbf{x}_1, \mathbf{x}_2)$ to $P_1$ and $f_2(\mathbf{x}_1, \mathbf{x}_2)$ to $P_2$.

**Party $P_c$ corrupted ($P_h$ is honest):**

- Upon $(\mathsf{Input}, \mathbf{x}_h)$ from $P_h$ and $(\mathsf{Input}, \mathbf{x}_c)$ from $P_c$, send $f_c(\mathbf{x}_1, \mathbf{x}_2)$ to $P_c$.
- Upon $(\mathsf{Deliver}, b)$ from $P_c$, if $b = 1$ then send $f_h(\mathbf{x}_1, \mathbf{x}_2)$ to $P_h$, otherwise send $\perp$.

Fig. 3: The ideal functionality for SFE with abort.

regards to output functions and adversarial models [55, 44, 63, 6]. Of particular interest to us is the work of [44] where they propose an ideal functionality in UC for this setting and then realise it with respect to private set intersection (PSI) in the presence of active corruptions. One reason that the ideal functionality of [44] cannot be expressed in the security framework used in SIM$^+$-CDP is that the functionality relaxes the guarantee of *input independence*, meaning that the corrupted party can choose their input based on the input of the honest party.

The PSI protocol of [44] outputs the exact set intersection (to only one of the parties, the other get no output) and therefore their protocol as a whole intuitively cannot be SIM$^*$-CDP. If one would instead realise their ideal functionality for computing a function with leakage, and enforce that all possible combinations of leakage functions and the output function to the corrupted party is DP (when seen as a composition), then SIM$^*$-CDP can be achieved. Below in Figure 4 we re-iterate the ideal functionality from [44] but augmented to have two potentially different classes of leakage functions for each party. The need for this is that since $f_1$ and $f_2$ need not be the same, as in the case when only one of them gets an output, then one can allow the party whose output function is DP with better parameters to have leakage functions that use up more of the privacy budget.

The only work of which we are aware that compute a DP mechanism whilst also allowing DP leakage is [6] where they compute sparse histograms. They claim SIM$^+$-CDP for their protocol but, somewhat implicitly, adapt the definition of SIM$^+$-CDP to have computational correctness and to use approximate DP as the core DP notion. The fact that these changes were necessary (to allow their ideal functionality to compute a truncated discrete Laplacian) and that they redefine the ideal functionality within the SIM$^+$-CDP definition further reinforce our arguments from above about the need for an ideal/real version of CDP where such relaxations and flexibility is built in. Further, it also motivates us below proposing a generalised CDP definition, so that future use-case dependent adjustments to existing ideal/real CDP notions can be phrased as instantiations of this new definition rather than simply as (sometimes implicitly) adjusted forms of SIM$^+$-CDP. Finally we note that the seeming lack of protocols combining DP leakage with computing a DP mechanism presents finding such protocols as an intriguing direction for future work.

**Functionality $\mathcal{F}^{f,\mathcal{L}_1,\mathcal{L}_2}_{\textbf{SFE with leakage}}$**

**Parameters:**

- A function $f = (f_1, f_2) : (\{0,1\}^*)^2 \rightarrow (\{0,1\}^*)^2$.
- Two classes of functions $\mathcal{L}_j = \{(L^{pre}, L^{post})_{j1}, \ldots\}$, $j \in \{1,2\}$, with $L^{pre}_{ji}, L^{post}_{ji} : \{0,1\}^* \rightarrow \{0,1\}^*$.

**No corruptions:**

- Upon $\mathbf{x}_1$ from $P_1$ and $\mathbf{x}_2$ from $P_2$, deliver $f_1(\mathbf{x}_1, \mathbf{x}_2)$ to $P_1$ and $f_2(\mathbf{x}_1, \mathbf{x}_2)$ to $P_2$.

**Party $P_c$ corrupted ($P_h$ is honest):**

- Upon $\mathbf{x}_h$ from $P_h$ and $(\mathsf{Leak}, L^{pre})$ from $P_c$, if there exists an element $(L^{pre}, \cdot)$ in $\mathcal{L}_c$ then send $L^{pre}(\mathbf{x}_h)$ to $P_c$, otherwise send $\bot$.
- Upon $\mathbf{x}_c$ and $(\mathsf{Leak}, L^{post})$ from $P_c$, if there exists an element $(L^{pre}, L^{post})$ in $\mathcal{L}_c$ then send $L^{post}(\mathbf{x}_h)$ to $P_c$, otherwise send $\bot$. Regardless, also send $f_c(\mathbf{x}_1, \mathbf{x}_2)$ to $P_c$.
- Upon $(\mathsf{Deliver}, b)$ from $P_c$, if $b = 1$ then send $f_h(\mathbf{x}_1, \mathbf{x}_2)$ to $P_h$, otherwise send $\bot$.

Fig. 4: The ideal functionality for reactive two-party SFE with abort and leakage.

### 3.5 A more general definition, SIM$^\circ$-CDP

To round of the part of this work concerning CDP definitions, we propose a general definition meant to capture the design paradigm of simulation-based CDP via the ideal/real paradigm, of which both SIM$^+$-CDP and SIM$^*$-CDP are instantiations. The point of introducing such a general definition is firstly to offer a blueprint in which to formulate possible further versions of this type of CDP, thereby hopefully avoiding new definitions being written using syntax making the relation to earlier similar definitions muddled or being formulated solely in terms of how they differ from some given existing definition. Secondly, having a unified generalised definition might make studying its different instantiations more convenient and similarly facilitates discussing the intuitive appeal of this flavor of CDP without being weightened down by concrete details.

**Definition 5 (SIM$^\circ$-CDP).** *The two-party protocol $\pi$ is SIM$^\circ$-CDP with respect to DP notion* $\mathsf{TYPE}$*-DP and ideal/real security notion* $\mathsf{SEC}$ *for the ideal functionality $\mathcal{F}$ and a given adjacency notion if $\pi$ realises $\mathcal{F}$ in the sense of* $\mathsf{SEC}$ *and for all ideal-world adversaries $\mathcal{S}$, the view of $\mathcal{S}$ is* $\mathsf{TYPE}$*-DP with respect to the adjacency notion.*

It is immediately clear that SIM$^+$-CDP is the same as SIM$^\circ$-CDP with pure DP as the core DP notion and their slightly adapted version of the security definition in [41] as security notion. Then the bulk of the discussion in this section thus far can be seen as concerning the ways in which we regard the specific choice of security notion as being inconvenient with respect to GED. Similarly, SIM$^*$-CDP is the same as SIM$^\circ$-CDP with approximate DP and UC security. Further, we note that the adapted CDP notion used in [6] is indeed an instantiation of SIM$^\circ$-CDP using approximate DP and standard standalone security (Definition 7.2.6 in [41]) but with computational correctness. This CDP notion is stronger than SIM$^+$-CDP in that it requires efficient simulators but weaker in the sense of having relaxed correctness and approximate DP. One should also note that since the protocol in [6] is not analysed in the UC framework it cannot claim SIM$^*$-CDP thus causing the need for this 'intermediate' CDP notion.

For future work there might be cause to consider other choices of DP notion, such as f-DP [26] or zero-concentrated DP [32, 9], or security models, such as the stand-alone model [41, 12] either unaltered or tweaked differently than in SIM$^+$-CDP.

## 4 A SIM$^*$-CDP version of the geometric mechanism

To demonstrate the use of our new definition, we now go through in detail how to satisfy it for the standard SFE ideal functionality with the truncated geometric mechanism as the function. Conceptually, this is very simple; one can simply use any PPT algorithm that transforms a random seed into a distribution that has sufficiently small statistical distance to a truncated geometric distribution (of which there are available options such as [40, 3, 33]) and then compute that algorithm in MPC via some general-purpose, active secure, protocol. It is however worth to consider hurdles that arise in the details, such as how to handle the mechanisms dependence on the query function, having a query function whose sensitivity depends on the inputs of both parties and consequences of working with modular arithmetics.

One core step is, naturally, to sample a distribution that is statistically close to a range-truncated geometric distribution. Such a truncated geometric distribution can be found in [40, 3, 33], however they truncate to a range between 0 and some fixed positive integer, which is also the range of the counting queries they consider. Their results and methods however extend to $\mathbb{Z}_q$, and general queries of bounded magnitude.

**Definition 6 (Truncated geometric distribution).** *Define the truncated geometric distribution $Z \sim Geo_{q,\lambda}(\bar{f})$ centered at $\bar{f} \in \mathbb{Z}_q$, truncated to $\mathbb{Z}_q := [\lceil -q/2 \rceil, \lfloor q/2 \rfloor)$, by its pmf:*

$$f_Z(z) = \frac{e^{1/\lambda} - 1}{e^{1/\lambda} + 1} e^{\frac{-|z-\bar{f}|}{\lambda}} \tag{2}$$

*for $z \notin \{\lceil -q/2 \rceil, \lceil q/2 - 1 \rceil\}$, and*

$$f_Z(z) = \frac{1}{e^{1/\lambda} + 1} e^{\frac{-|z-\bar{f}|}{\lambda}} \tag{3}$$

*for $z \in \{\lceil -q/2 \rceil, \lceil q/2 - 1 \rceil\}$.*

**Definition 7 (Range-truncated geometric mechanism).** *Let $\lambda \in \mathbb{N}^{-1}$ and let $f : \mathcal{D} \to \mathbb{Z}_q$ be a deterministic function. The Range-truncated geometric mechanism over $\mathbb{Z}_q$ for $f$ is defined as*

$$\mathcal{M}^{q,f,\lambda}_{RTGeo}(D) := Geo_{q,\lambda}(f(D)). \tag{4}$$

It is easy to verify that $\mathcal{M}^{q,f,\lambda}_{RTGeo}(D)$ is an $(\varepsilon, 0)$-DP mechanism as long as $\lambda = \frac{\varepsilon}{\Delta f}$, where $\Delta f$ denotes the $l_1$-sensitivity of $f$. In line with [3], we only allow $\lambda \in \mathbb{N}^{-1}$, in order to avoid the need to represent real numbers, and this also implies $\varepsilon \in \mathbb{N}^{-1}$. Whereas the mechanism above gives DP, it is inconvenient to sample the noise distribution directly, partly because it requires knowledge of $f(D)$ and partly because it may require sampling probabilities that cannot be generated from a polynomial number of fair coins. Therefore we consider the following mechanism.

**Definition 8 (Subrange-truncated geometric mech.).** *Let $B \in \{1, \ldots, \lceil q/2 \rceil - 1\}$ and $\lambda \in \mathbb{N}^{-1}$. Let the Subrange-truncated geometric mechanism over $\mathbb{Z}_q$ with noise truncation to $\mathbb{Z}_{2B}$, for a function $f : \mathcal{D} \to \mathbb{Z}_q$, be defined as $\mathcal{M}^{2B,f,\lambda}_{SRTGeo}(D) := f(D) + Geo_{2B,\lambda}(0)$, with the addition performed over $\mathbb{Z}_q$.*

In the simple lemma below we give a bound on the statistical distance between the two mechanisms we have introduced this far. The proof is found in Appendix C.1. We note that we need to introduce a bound on the absolute value of the query function, as to not have the sensitivity of the function be affected by the modular arithmetics.

**Lemma 1.** *Let $f^{max} := \max_{D \in \mathcal{D}} |f(D)|$, $B \in \mathbb{N}$, $\lambda \in \mathbb{N}^{-1}$ and $q > 2f^{max} + 2B$. Then the statistical distance between $\mathcal{M}^{2B,f,\lambda}_{SRTGeo}(D)$ and $\mathcal{M}^{q,f,\lambda}_{RTGeo}(D)$ for all $D \in \mathcal{D}$ is at most $e^{-B/\lambda}$.*

We are now one step closer to a functionality that can be efficiently realised, since the noise sampling is no longer dependent on the function evaluation and the support of the noise is potentially much smaller than the entire $\mathbb{Z}_q$ and the support of $f$. The trouble still remains that the probabilities might not be negative polynomial powers of two. In [28, 33] they give distributions that can be exactly sampled under this constraint and that has a small statistical distance from a truncated geometric distribution. We use the procedure FDL (*Finite-range Discrete Laplacian*) introduced in EIKN [33].

**Definition 9 (FDL function and procedure).** *Let* $\mathbf{r} \in \{0,1\}^{Bd+1}$ *be independent fair coins and* $0 < e^{-1/\lambda} < 1$. *Let* $\hat{\alpha}^1 \leftarrow \frac{1-e^{-1/\lambda}}{1+e^{-1/\lambda}}$ *and* $\hat{\alpha}^i \leftarrow 1 - \hat{\alpha}^1$ *for* $i = 2, ..., B$ *be public parameters. Let* $\oplus$ *and* $\wedge$ *denote addition and multiplication over the binary field and let* $\vee$ *be shorthand for computing the OR operation by using binary addition and multiplication. Let all other operands be defined as normally over the arithmetic field* $\mathbb{Z}_q$.

*Define the function* $\texttt{FDL}_{\lambda,B,d} : \{0,1\}^{Bd+1} \rightarrow \mathbb{Z}_{2B} \subseteq \mathbb{Z}_q$ *by the procedure in Algorithm 1. Let* $\alpha = (\alpha_1, \alpha_2, ...)$ *be the bit decomposition of* $\hat{\alpha}$. *The subprocedure* $\texttt{Ber}_{\hat{\alpha}} : \{0,1\}^d \times \{0,1\}^d \rightarrow \{0,1\}$ *for generating approximate Bernoulli trials with parameter* $\hat{\alpha}$ *using a randomness seed in* $\{0,1\}^d$ *is defined the procedure in Algorithm 2.*

```
Procedure FDL
Input: $\mathbf{r} \in \{0,1\}^{Bd+1}$
  1. Sample $B$ approximate Bernoulli trials $\beta_i \leftarrow \texttt{Ber}_{\hat{\alpha}^i}((r_{d(j-1)+1}, ..., r_{dj}))$ for $i = 1, ..., B$.
  2. For $i = 1, ..., B$: set $c_i \leftarrow \wedge_{j=1}^{i} \beta_j$.
  3. Set $l \leftarrow B - \sum_{i=1}^{B} c_i$.
  4. Set $\sigma \leftarrow 2 \cdot r_{Bd+1} - 1$.
  5. Output $\sigma \cdot l$.
```

**Algorithm 1:** The algorithm description for the FDL procedure.

```
Procedure Ber
Input: $\mathbf{r} \in \{0,1\}^d$, $\alpha \in \{0,1\}^d$
  1. For $i = 1, ..., d$, set $c_i \leftarrow \alpha_i \oplus r_i$.
  2. For $i = 1, ..., d$, set $e_i \leftarrow \vee_{j=1}^{i} c_j$.
  3. For $i = 1, ..., d$, set $v_i \leftarrow e_i \oplus e_{i-1}$, with $e_0 \leftarrow 0$.
  4. Set $\beta \leftarrow 1 \oplus_{i=1}^{d} (r_i \wedge v_i)$ and output $\beta$.
```

**Algorithm 2:** The algorithm description for the Ber procedure.

Note that FDL is an exact method for turning $Bd + 1$ fair coins into a sample of a distribution that is statistically close to a truncated geometric one. It is clear that if the number of fair coins is polynomial in $\kappa$ then FDL runs in strict polynomial time. With some abuse of notation, we use FDL to denote both the procedure and the probability distribution it generates upon being given fair coins.[10]

**Definition 10 (FDL mechanism).** *Let $B \in \{1, \dots, \lceil q/2 \rceil - 1\}$. Let the Finite Discrete Laplace mechanism (FDL) over $\mathbb{Z}_q$ for a function $f : \mathcal{D} \to \mathbb{Z}_q$ be defined as $\mathcal{M}_{\mathtt{FDL}}^{\lambda,B,d,f}(D) := f(D) + \mathtt{FDL}_{\lambda,B,d}$, with the addition performed over $\mathbb{Z}_q$.*

The following lemma is proven in EIKN [33]. For completeness, we also include a proof in Appendix C.2.

**Lemma 2.** *Let $f^{max} := \max_{D \in \mathcal{D}} |f(D)|$, $q > 2f^{max} + 2B$ and $B \in \{1, \dots, \lceil q/2 \rceil - 1\}$. If FDL is given independent fair coins and all the arithmetics are done over $\mathbb{Z}_q$, then the statistical distance between $\mathcal{M}_{\mathtt{FDL}}^{\lambda,B,d,f}(D)$ and $\mathcal{M}_{SRTGeo}^{2B,f,\lambda}(D)$ is at most $B \cdot 2^{-d}$.*

Further, we have that $\mathcal{M}_{RTGeo}^{q,f,\varepsilon/\Delta f}(D)$ is a useful approximation of $f$, as we show in the following lemma. The proof is found in Appendix C.3

**Lemma 3.** *Let $q > 2f^{max} + 2B$, $B \in \{1, \dots, \lceil q/2 \rceil - 1\}$. Let $f : \mathcal{D} \to \mathbb{Z}_q$ be an arbitrary deterministic function with $f^{max} := \max_{D \in \mathcal{D}} |f(D)|$ and let $\hat{f}(D) := \mathcal{M}_{RTGeo}^{q,f,\lambda}(D) : \mathcal{D} \to \mathbb{Z}_q$. Then $\hat{f}$ is $\left(\nu, \frac{2e^{-1/\lambda}}{e^{-1/\lambda}+1} e^{-\nu/\lambda}\right)$-additive-useful for $f$ for any positive integer $\nu$.*

## 5 A protocol for the FDL mechanism

From the previous section we know that the FDL mechanism is statistically close to the Range-truncated geometric mechanism, which is pure DP, and that this holds under some restrictions on the query function and on the parameter choices. At the same time, it is immediate that the Range-truncated geometric mechanism is statistically close to the untruncated geometric mechanism (i.e. when the noise is not truncated and that the modular arithmetics thus might cause overflows), as long as the value of the query function is somewhat far away from $q/2$ and $-q/2$. Therefore, there is a choice to be made which mechanism one chooses to have in the ideal functionality (call this the *ideal mechanism*), given that we will of course have the protocol compute the FDL mechanism via general-purpose MPC (in particular, we will use the *arithmetic black-box (ABB)* level of abstraction). The trade-off in this choice is that having RTGeo as the ideal mechanism will lead to $(\varepsilon_\kappa, 0)$-SIM$^*$-CDP as long as the statistical distances mentioned above are negligible in $\kappa$, essentially having the statistical distance be dealt with as part of the slack of the correctness of the protocol. On the other hand can this be avoided by letting $\mathcal{M}_{\mathtt{FDL}}$ be the ideal mechanism, thus leading

[10] We also note that the requirement that $e^{-1/\lambda} < 1$ is equivalent to $\lambda > 0$, which is already guaranteed by $\lambda \in \mathbb{N}^{-1}$.

to $(\varepsilon_\kappa, \delta_\kappa)-$SIM$^*$-CDP where the statistical distance is rather incorporated into the $\delta_\kappa$ term. As having an ideal mechanism as close as possible to standard DP mechanism is to be seen as a more direct realisation of GED, we opt for having RTGeo as the ideal mechanism.

As stated in the preliminaries, we consider two-party computation schemes that operate in $\mathbb{F}_q$ with $q$ being either a prime larger than 2 or a power of 2. We elaborate on active secure schemes for both domains in Appendix D. Implementing the `FDL` algorithm in either domain comes at a significant cost. Note that the `Ber` procedure and the first 2 steps of the `FDL` procedure consist of only binary arithmetics. However, the remainder of the `FDL` procedure consists of integer arithmetic. While there are protocols to evaluate these binary steps in the arithmetic domain, they are usually very costly. On the other hand, evaluating the algorithm in the binary domain comes with two problems: the summation and addition in binary would incur a significant cost, and second, the result would be a shared noise in the binary domain. Thus, applying the noise is limited to the binary domain. The mixed circuit approach (see Section 2.2) gives us a well-performing trade-off while maintaining the highest security guarantees.

We accept inputs represented in the binary domain, perform all operations until the fourth step through a binary circuit, translate all shares to the arithmetic domain, and perform the rest of the operations through an arithmetic circuit. For each of these "phases", we use protocols introduced before. We use SPDZ$_{2^k}$ [21] for the arithmetic computations, the FKOS protocol [38] for binary circuits and *daBits* (doubly-authenticated bits) [60] for translating between the domains. With correct parametrization, we can achieve the same security guarantees in different computation domains. Thus, the feasibility of the mixed circuit approach is easily tested. The mixed circuit approach is feasible if switching between circuits is cheaper than the computation overhead in either domain. In our application (Section 5.1), we will, as typically for DP applications, focus on arithmetic computations. Evaluating the `FDL` mechanism in the binary domain would, therefore, incur a cost that scales with the underlying application. For the arithmetic case, we have an additional cost of assuring all input ranges (e.g., assert that binary coins $\in \{0, 1\}$) and evaluate binary gates with arithmetic circuits. Section 6 has a longer discussion about input validation.

We describe our protocol using the *Arithmetic Black Box (ABB)*, which is an ideal functionality in the UC framework. Very roughly, the ABB is a functionality that can take inputs from the parties and compute linear combinations and multiplications between stored values and output stored values. We use a flavor of the ABB that can do these operations over $\mathbb{F}_{2^k}$ and $\mathbb{F}_q$. Additionally, the ABB can translate values stored as elements of the binary field to binary values within the larger field. More concretely, we use the formulation of the ABB that can be found in [37] and we include a definition of the ideal functionality in Appendix B. Our protocol is presented in Figure 5.

**Protocol $\pi_{\mathcal{M}_{\texttt{FDL}}}$**

**Parameters:** Natural numbers $B, d, q, N$, bit decomposition $\hat{\alpha}^1, ..., \hat{\alpha}^d$ and an efficiently computable function $f : \mathbb{Z}_q^{2N} \to \mathbb{Z}_q$, meaning it can be computed using polynomially many multiplications and linear combinations in $\mathbb{Z}_q$. Assume access to $\mathcal{F}_{ABB}$.

**Initialisation:**

1. Player $i$ locally samples $Bd + 1$ fair coins and stores them as $\mathbf{e}^i$.
2. Player $i$ sends random seed vector $\mathbf{e}^i \in \mathbb{Z}_2^{Bd+1}$ as $Bd + 1$ consecutive inputs to $\mathcal{F}_{ABB}$ to be stored as elements of the binary field.
3. For $j = 1, ..., Bd + 1$ the players compute $r_i \leftarrow e^1_j \oplus e^2_j$ via $\mathcal{F}_{ABB}$.

**Noise sampling:**

1. Each operation in the first two steps of the `FDL` specification is performed via $\mathcal{F}_{ABB}$.
2. In $\mathcal{F}_{ABB}$, the values $c_1, ..., c_B$ and $r_{Bd+1}$ are transformed to elements in the arithmetic field.
3. All remaining operations in the `FDL` specification are performed via $\mathcal{F}_{ABB}$.

**Finishing:**

1. Player 1 sends $\mathbf{x} \in \mathbb{Z}_q^N$ and player 2 sends $\mathbf{y} \in \mathbb{Z}_q^N$ to $\mathcal{F}_{ABB}$ and then $f$ is computed via $\mathcal{F}_{ABB}$ according to its specification. The result is stored as $\bar{f}$.
2. The sum of $\bar{f}$ and the `FDL` sample is computed via $\mathcal{F}_{ABB}$ and the result is output to the players.

Fig. 5: The protocol description for the `FDL` mechanism in the $\mathcal{F}_{ABB}$-hybrid world.

We are now ready to present our main theorem, namely that the protocol we have introduced indeed is $(\varepsilon_\kappa, 0)$-SIM$^*$-CDP. Let *decomp*$(\lambda, d)$ be short for the bit-decomposition of $\lambda$ truncated to $d$ bits.

**Theorem 1.** *Let $q > 2f^{max}_\kappa + 2B_\kappa$, $B_\kappa \in \{1, \dots, \lceil q/2 \rceil - 1\}$, $\lambda_\kappa = \frac{\varepsilon_\kappa}{\Delta f_\kappa}$ and let $e^{-B_\kappa/\lambda_\kappa}$ and $B_\kappa 2^{-d_\kappa}$ be negligible in $\kappa$. Let $\{f_\kappa : \mathbb{Z}_q^{2N} \to \mathbb{Z}_q\}_{\kappa\in\mathbb{N}}$ be an ensemble of efficiently computable deterministic functions with $f^{max}_\kappa := \max\limits_{D\in\mathbb{Z}_q^{2N}} |f_\kappa(D)| \forall\kappa$. Let $\{\hat{f}_\kappa(D)\}_{\kappa\in\mathbb{N}}$ be $\{\mathcal{M}^{q,f_\kappa,\lambda_\kappa}_{RTGeo}(D)\}_{\kappa\in\mathbb{N}}$. Then $\pi_{\mathcal{M}_{\texttt{FDL}}}(B_\kappa, d_\kappa, q, N, decomp(\lambda_\kappa, d_\kappa), f_\kappa)$ is an $(\varepsilon_\kappa, 0)$-SIM$^*$-CDP protocol for the ideal functionality $\mathcal{F}^{\hat{f}_\kappa}_{SFE}$ with respect to the same adjacency notion as in the calculation of $\Delta f_\kappa$ in the $\mathcal{F}_{ABB}$-hybrid world.*

*Proof.* The definition of SIM$^*$-CDP demands two things to be shown, namely that the view of the simulator is approximate DP and that the protocol UC-realises the ideal functionality. The first requirement is fulfilled as the only message sent from $\mathcal{F}^{\hat{f}_\kappa}_{SFE}$ to the corrupted party is $\mathcal{M}^{q,f_\kappa,\lambda_\kappa}_{RTGeo}(D)$ and this is pure DP due to the fact that the range-truncated geometric mechanism is pure DP under the standard parametrisation specified in the theorem. The other parts of the view of $\mathcal{S}$ (like its input and randomness tape) are independent of the inputs of the honest party, thus making the view of $\mathcal{S}$ as a whole $(\varepsilon_\kappa, 0)$-DP. Further, this holds for all types of malicious behavour of $\mathcal{S}$ since, due to the formulation of $\mathcal{F}_{SFE}$, the only way $\mathcal{S}$ can change its view is to refuse to collaborate in the protocol or change its inputs and both of those decisions would have to be made independently of the inputs of the honest party (thus making those decisions $(0, 0)$-DP as well).

The UC-realisation of the ideal functionality follows directly from the use of the arithmetic black-box and the statistical indistinguishability between $\mathcal{M}_{\texttt{FDL}}$ and $\mathcal{M}_{RTGeo}$, which follows from lemmas 1 and 2 together with the assumptions of the theorem. In particular, due to the use of $\mathcal{F}_{ABB}$, the view of the corrupted party in the hybrid world consists of only its input, random coins and the output returned from $\mathcal{F}_{ABB}$, which is exactly $\mathcal{M}_{\texttt{FDL}}$. Similarly, the view of the corrupted party in the ideal world is also only its input, random coins and output returned from $\mathcal{F}_{SFE}^{\hat{f}_\kappa}$. Therefore the simulator that simply outputs its view (after having changed its inputs and/or aborted with respect to its random coins as the hybrid-world adversary does) yields a view that is computationally indistinguishable from that of the hybrid-world adversary. Further, this simulator is strict PPT due to it performing only the same operations as the hybrid-world adversary (choosing input and abort behaviour based on its coins and then receive one $\mathbb{Z}_q$ element), hence the theorem follows.

□

**Asymptotic computational cost.** We consider the computational cost of $\pi_{\mathcal{M}_{\texttt{FDL}}}$ in terms of calls to the ABB, ignoring the cost of computing $f$. This rough model for calculating computation cost is reasonable in two ways: Firstly, local operations are canonically negligible in terms of computation cost compared to operations that require interaction. Secondly, in practice, the instantiation of the ABB greatly influences the computation cost in practical terms. As is shown in EIKN [33], the asymptotic computational cost of the `FDL` function (here in terms of the number of multiplications) is $O(Bd)$. This complexity follows directly from Definition 9 since all steps of the `FDL` procedure are repeated $B$ times (that is, $B$ Bernoulli trials are sampled and there are $B$ elements in the sum) and within the Bernoulli trial subprocedure, all steps consist of $d$ arithmetic operations.

It is important to note that the cost of sampling the noise is independent of the data query $f$. Relative DP usefulness intuitively increases as the number of elements in the input dataset grow. However, the performance of the sampling protocol scales with the number of queries and not with the size of the input dataset, thus amortizing its execution time further.

### 5.1 Application: Integer inner-products with bounded elements

We now compute integer inner-products using the $\pi_{\mathcal{M}_{FDL}}$ protocol. This query type is particularly interesting for a few reasons. First, it is non-linear and cannot be expressed as an aggregate function without knowledge of the other party's inputs. Second, it is a fundamental building block for more complicated queries like matrix multiplications with vast applications in data processing such as machine learning. In order to use $\pi_{\mathcal{M}_{FDL}}$, the query needs a bounded maximal absolute value, and for accuracy, we want the sensitivity of the query to be small. Therefore, we consider only inner products where the input vectors have elements between $a \in \mathbb{Z}_q$ and $b \in \mathbb{Z}_q$. We assume that the difference between $a$ and $b$ is a power of 2, to facilitate inserting an input as a sequence of bits.

We consider DP with the bounded (*'change-one'*) adjacency notion, and the data universe is $([a, b])^*$, such that each input $D$ to $f$ (as well as the protocol and the mechanism) is a tuple of $2N$ elements from $[a, b]$. Let $D := \mathbf{x}||\mathbf{y}$. The inner product $f(D)$ is defined as $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^{N} x_i y_i$ with operations over $\mathbb{Z}_q$. The sensitivity $\Delta f$ of the inner product is $\max(|a^2 - ab|, |b^2 - ab|)$, under the assumption that $|f(\mathbf{x}, \mathbf{y})|$ is smaller than $\lfloor q/2 \rfloor$ such that field operations mimic integer behavior. We also have that $f^{max} = N \cdot \max(a^2, b^2)$.

**Parameter choices.** From the properties above, the following parameter considerations follow: Let the security parameter be the bit-length of a field element, i.e. $\kappa = \lceil \log_2(q) \rceil$, as is canonical. Let both $\varepsilon_\kappa$ and $\Delta f$ (that is, by choice of $a, b$) be independent of $\kappa$. Further, we can set the `FDL` specific parameters as $B = d = \kappa$. Finally, we have $q > 2f^{max} + 2B = 2N \cdot \max(a^2, b^2) + 2B$, where the inequality holds for sufficiently large $\kappa$.

In practice, one strategy is to choose $\kappa$ as a canonical value for statistical security in cryptography, e.g., $\kappa = 40$, and then let this also be $B$ and $d$. The practical choice of $\varepsilon$ is highly challenging, and there is a lively discussion in the literature on it, although consensus is largely lacking [29, 53, 57, 52]. Luckily, there is no direct dependence on the choice of $\varepsilon$ in the other parameters. Finally, this leaves the choices of $a, b$, and $N$. Here, we care about the distance $|a - b|$ and the size of $N$. Both parameters allow for wider usage scenarios when increased. However, increasing $N$ has adverse effects on runtime, and a larger distance causes a higher sensitivity and decreased usefulness (if $\varepsilon$ is kept fixed). Finally, there is a trade-off between $N$ and the sizes of $a, b$ due to their dependence on $q$. In practice, this can be circumvented by increasing the modulus size $q$ in the underlying MPC instantiation.

# 6 Implementation and Practical performance

We tested our protocol by implementing it in the multi-protocol SPDZ (MP-SPDZ) [50] library. Among other things, they provide efficient implementations of the SPDZ$_{2^k}$ [21] and the FKOS [38] MPC schemes, and da-bit [60] and edabit [37] implementations. We implement procedure `Ber` in the FKOS scheme and procedure `FDL` in the mixed-circuit setting with FKOS and SPDZ$_{2^k}$. We find that only one switch between computation domains is necessary, making mixed-circuit computation very competitive in performance. More precisely, this approach is faster than previous instantiations if the conversion cost is lower than the additional overhead in the unfit computation domain. Given the protocol in EIKN [33], circuit conversion has to be faster than the overhead of computing the Bernoulli and prefix-or functionality in the arithmetic domain.

In MPC schemes, communication is typically the bottleneck of efficient function evaluation. While some communication is necessary during the computation, much of the data transfer happens in a pre-processing phase. In our setup, we have three main components that require expensive pre-processing: shared randomness for inputs, authenticated multiplication triples, and doubly authenticated bits. In our inner-product use case, we only generate one `FDL` sample. However, most pre-processing operations come in blocks of size $B$ or $d$. In our implementation, we take special care to minimize the communication rounds and adapt the pre-processing batch sizes to accommodate our protocol execution.

Our setting provides security in the presence of active adversaries. Since these parties can deviate arbitrarily from the protocol, they might send input out of range. It is, therefore, necessary to prove the correctness of the input domain in both the FDL mechanism and the query function. There are different strategies to achieve such a feat. We note that the ABB accepts inputs of two types, either elements in the binary field or the larger finite field. We need to restrict the values to the pre-defined range for inputs in the arithmetic domain. Were we not to perform such an input validation, this would result in an increased sensitivity of the function (in relationship to what is a priori agreed upon by the two parties), thwarting the privacy level of the DP mechanism. In the presence of passive adversaries, however, there is of course no need to validate the inputs since the adversary will per definition not give out-of-range inputs. This requirement of a *proof of function sensitivity* also arises in other scenarios where the sensitivity is directly dependent on the secret data of multiple parties.

To provide such a range-proof of the inputs of each party, we consider two main options: Firstly, one could accept the inputs as elements in the larger field and then perform a zero-knowledge range proof[11] within the MPC domain, and secondly, one could accept the inputs bit-by-bit and re-compose those bits into elements of the larger field. These approaches present a trade-off in input and

[11] For instance, such as described in the Bulletproofs paper [11].

proof complexity. In the first approach, the cost of inputting a value is constant (i.e., depending on $2^k$ in our example) while proving the range is linear in the bound. In the bit-by-bit setting, the input and proving cost are logarithmic in the bound. The second approach is thus more efficient for larger bound values depending on the specific scheme.

### 6.1 Benchmarks

In this section, we present benchmarks of our `FDL` mechanism with $B = d = \kappa$ and measure performance for different settings[12]. Relevant for parameter $\hat{\alpha}$, the bit decomposition of the Bernoulli bias, is the decomposition length $d$. When setting a value $\alpha$, the binary decomposition truncates this value to the predefined precision. Although our code can be instantiated with any number of parties, we fixed the number of parties to 2 as to align with the formalities of earlier section. We provide exemplary data points at 40- and 80-bit, typical statistical security parameters. Next, we evaluate the mechanism at 128-bit, a usual conservative choice as a computational security parameter. Note that the underlying security parameters for SPDZ$_{2^k}$ are fixed to 64-bit computational and 64-bit statistical security. We run all benchmarks on a Linux server with an AMD Ryzen 9 7900X CPU (4.7 GHz). Each party only has access to one thread for computations. We separate our results into single sample computation and amortized evaluation for 1000 sampels. The single sample evaluation is further split into the pre-processing and online phases of MPC, where the pre-processing step consists of generating necessary multiplication triples and da-bits.

Table 1 presents the runtime metrics for different network settings. In Setting 1, we have an unrestricted LAN setup. Setting 2 simulates a less powerful LAN setup by limiting the network to 1Gbit/s and the round-trip time (RTT) to 1ms. Finally, in Setting 3, we simulate a WAN network with 100Mbit/s and 100ms RTT, reflecting a solid but distant connection (e.g., intercontinental). Given the asymptotic complexity $O(Bd)$, the runtime results reflect the expected quadratic growth in the security parameter. Regarding the network settings, communication is needed for inputs, binary AND gates, arithmetic multiplication, secret share conversion, and outputs. Since inputs, conversions, and computations depend on one or both parameters $B$, or $d$, the negative impact of a reduced network speed and increased RTT is increased. Compared to concurrent work [49], our mechanism outperforms their result in runtime and memory for the overall computation.[13] Arguably, their setup heavily optimizes the online phase, making it more efficient if pre-processing can be off-loaded or performed in advance. However, sampling Laplacian noise in MPC can generally be seen as pre-processing since the sensitivity of a function is known before the data is processed, and the

[12] The code for our implementation can be found at `https://extgit.iaik.tugraz.at/krypto/geometric_sampler`.

[13] One should further note that [49] is in the more efficient setting of passive adversaries, thus making direct comparisons skewed in their favor.

parties can already engage in the noise sampling procedure before their inputs to the query function have been fixed. Comparing with [33] is challenging as they only provide asymptotic complexities and base their results on arithmetic evaluations of binary computations from [59]. Our approach, on the other hand, is based on mixed circuits [60] and includes substantial performance improvements by dedicated parameter optimizations.

Table 1: Runtime in seconds of benchmarks with different security levels. Total computes a single sample, while amortized runtime assumes 1000 samples.

| Protocol | $\kappa$ | Prep. | Online | Total | Amort. |
|---|---|---|---|---|---|
| 10 Gbit/s with RTT of 1 ms | | | | | |
| Ours | 40 | 74.7 | 42 | 116.6 | 34.6 |
| | 80 | 94.2 | 119.9 | 214.1 | 118.5 |
| | 128 | 130 | 276.9 | 406.9 | 283.4 |
| [49] | 40 | 1606 | 37.72 | 1 643 | 992$^{\dagger}$ |
| 1 Gbit/s with RTT of 1 ms | | | | | |
| Ours | 40 | 182.9 | 248.4 | 431.2 | 69.7 |
| | 80 | 245.6 | 650.2 | 895.7 | 209.7 |
| | 128 | 345.6 | 1 362 | 1 707 | 520.3 |
| [49] | 40 | 4 707 | 4.81 | – | 4 711$^{\ddagger}$ |
| 100 Mbit/s with RTT of 100 ms | | | | | |
| Ours | 40 | 11 256 | 20 486 | 31 742 | 577.9 |
| | 80 | 15 215 | 51 794 | 67 009 | 1 604 |
| | 128 | 20 795 | 105 350 | 126 145 | 3 558 |
| [49] | 40 | 42 352 | 47.99 | – | 42 400$^{\ddagger}$ |

† Amortized over 40 samples

‡ Amortized over 10 samples, no single sample performance provided.

In Table 2, we present benchmarks for network costs for each security parameter. We see that the network cost of our implementation is lower than that of [49], further showing that their round complexity is much lower than that of the malicious secure SPDZ$_{2^k}$ protocol. Given the network cost, we could further reduce the network bandwidth before its limiting impact equals a slow RTT.

## 7 Conclusions and outlooks

In this work we revisit the idea of generic emulation of the central dataholder (GED) as a method to achieve accuracy equal to that of the central model of DP without the need for a single trusted dataholder. The bulk of our work is spent analysing previous definitions of CDP in the multiparty setting, noting

Table 2: Network cost in MB of different laplace sampling settings. The amortized cost assumes 1000 samples.

| Protocol | $\kappa$ | Prep. | Online | Total | Amort. |
|---|---|---|---|---|---|
| Ours | 40 | 14.7 | 17.9 | 65.3 | 16.7 |
| | 80 | 20.9 | 58.3 | 158.3 | 75.3 |
| | 128 | 29.2 | 143.4 | 345.2 | 173.6 |
| [49] | 40 | – | – | 492.7† | – |

† Run with single sample, no amortized network cost provided.

that whereas they are very well-suited for theoretic study and use with special-purpose MPC schemes, they all fit somewhat suboptimally to the task of GED. Since one of them, SIM$^+$-CDP, appears to fit very well conceptually but has some details preventing its use together with canonical DP mechanisms, we propose both a generalised version of it, SIM$^\circ$-CDP, and another instantiation of that generalised definition, SIM$^*$-CDP, that we argue is more fitting to the current state-of-the-art in both general-purpose MPC and DP.

As always when formulating new definitions in cryptography questions arise, such as whether the definition is intuitive, practically usable, and not overly relaxed or strict. On the usability front, we present evidence that SIM$^*$-CDP is practical since it allows us to design efficient, quite general protocols of natural tasks that fulfill it, and the proof that the definition is satisfied follows essentially directly from the use of general-purpose MPC and a DP mechanism. Further, the definition appears intuitive due to its closeness to both previous definitions and established formalities in both the DP and MPC domains. There is, however, much need for additional scrutiny, and this is the case also for the question about balance in the definition. Interesting open questions here are, for instance, to relate the definition back to previous ones and see whether there is some characteristic trait of DP that is captured in the previous ones but not in SIM$^*$-CDP, and analyse under what criteria the definitions imply each other. Another interesting avenue of questions is that regarding properties of the definition itself, perhaps primarily when it comes to composition. Since both UC security and DP in general are highly advanced when it comes to the composition of protocols, SIM$^*$-CDP gives us a new and more nuanced definition to use when it comes to the analysis of compositional properties.

Regarding more practical outlooks, one interesting avenue of study is the design of protocols fulfilling a version of SIM$^\circ$-CDP for ideal functionalities other than SFE, such as SFE with leakage. Hopefully, the ease with which SIM$^\circ$-CDP can be adapted to specifics in the used MPC scheme and then directly applied can further the ongoing push to combine MPC and DP, leading both to protocols of increased efficiency and more nuanced privacy guarantees.

**Acknowledgements.** We gratefully thank the authors of MPRV [58] for providing a full version of their paper and for answering our questions. In particular, we thank Salil Vadhan for his helpful comments regarding earlier versions of our SIM$^+$-CDP definition. We thank Lea Demelius and Peter Waldert for fruitful discussions on earlier drafts of this paper.

This project was supported by the "DDAI" COMET Module within the COMET – Competence Centers for Excellent Technologies Programme, funded by the Austrian Federal Ministry (BMK and BMDW), the Austrian Research Promotion Agency (FFG), the province of Styria (SFG) and partners from industry and academia. The COMET Programme is managed by FFG.

This project was also in part funded by the CONFIDENTIAL-6G EU project (Grant No: 101096435).

## References

1. Aly, A., Orsini, E., Rotaru, D., Smart, N.P., Wood, T.: Zaphod: Efficiently combining lsss and garbled circuits in scale. In: Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography. p. 33–44. WAHC'19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3338469.3358943, https://doi.org/10.1145/3338469.3358943
2. Anandan, B., Clifton, C.: Laplace noise generation for two-party computational differential privacy. In: 2015 13th Annual Conference on Privacy, Security and Trust (PST). pp. 54–61 (2015). https://doi.org/10.1109/PST.2015.7232954
3. Balcer, V., Vadhan, S.: Differential privacy on finite computers. Journal of Privacy and Confidentiality **9**(2) (Sep 2019). https://doi.org/10.29012/jpc.679, https://journalprivacyconfidentiality.org/index.php/jpc/article/view/679
4. Beaver, D.: Correlated pseudorandomness and the complexity of private computations. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. p. 479–488. STOC '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237996, https://doi.org/10.1145/237814.237996
5. Beimel, A., Nissim, K., Omri, E.: Distributed private data analysis: Simultaneously solving how and what. In: Wagner, D. (ed.) Advances in Cryptology – CRYPTO 2008. pp. 451–468. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
6. Bell, J., Gascón, A., Ghazi, B., Kumar, R., Manurangsi, P., Raykova, M., Schoppmann, P.: Distributed, private, sparse histograms in the two-server model. p. 307–321. CCS '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3548606.3559383, https://doi.org/10.1145/3548606.3559383
7. Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., Seefeld, B.: Prochlo: Strong privacy for analytics in the crowd. In: Proceedings of the 26th Symposium on Operating Systems Principles. ACM (oct 2017). https://doi.org/10.1145/3132747.3132769, https://doi.org/10.1145%2F3132747.3132769

8. Bun, M., Chen, Y.H., Vadhan, S.: Separating computational and statistical differential privacy in the client-server model. In: Proceedings, Part I, of the 14th International Conference on Theory of Cryptography - Volume 9985. p. 607–634. Springer-Verlag, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53641-4_23
9. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Hirt, M., Smith, A. (eds.) Theory of Cryptography. pp. 635–658. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
10. Böhler, J.: Input Secrecy & Output Privacy: Efficient Secure Computation of Differential Privacy Mechanisms. Ph.D. thesis, Karlsruher Institut für Technologie (KIT) (2021). https://doi.org/10.5445/IR/1000141098
11. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 315–334 (2018). https://doi.org/10.1109/SP.2018.00020
12. Canetti, R.: Security and composition of multiparty cryptographic protocols. J. Cryptol. **13**(1), 143–202 (jan 2000). https://doi.org/10.1007/s001459910006, https://doi.org/10.1007/s001459910006
13. Canetti, R.: Universally composable security. J. ACM **67**(5) (sep 2020). https://doi.org/10.1145/3402457, https://doi.org/10.1145/3402457
14. Canonne, C., Kamath, G., Steinke, T.: The discrete gaussian for differential privacy. Journal of Privacy and Confidentiality **12**(1) (Jul 2022). https://doi.org/10.29012/jpc.784
15. Champion, J., shelat, a., Ullman, J.: Securely sampling biased coins with applications to differential privacy. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. p. 603–614. CCS '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3319535.3354256, https://doi.org/10.1145/3319535.3354256
16. Chan, T.H.H., Shi, E., Song, D.: Optimal lower bound for differentially private multi-party aggregation. In: Epstein, L., Ferragina, P. (eds.) Algorithms – ESA 2012. pp. 277–288. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
17. Cheu, A., Smith, A., Ullman, J., Zeber, D., Zhilyaev, M.: Distributed differential privacy via shuffling. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019. pp. 375–403. Springer International Publishing, Cham (2019)
18. Cheu, A., Yan, C.: Necessary Conditions in Multi-Server Differential Privacy. In: Tauman Kalai, Y. (ed.) 14th Innovations in Theoretical Computer Science Conference (ITCS 2023). Leibniz International Proceedings in Informatics (LIPIcs), vol. 251, pp. 36:1–36:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2023). https://doi.org/10.4230/LIPIcs.ITCS.2023.36, https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2023.36
19. Chor, B., Kushilevitz, E.: A zero-one law for boolean privacy. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing. p. 62–72. STOC '89, Association for Computing Machinery, New York, NY, USA (1989). https://doi.org/10.1145/73007.73013, https://doi.org/10.1145/73007.73013
20. Corrigan-Gibbs, H., Boneh, D.: Prio: Private, robust, and scalable computation of aggregate statistics. In: 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17). pp. 259–282. USENIX Association, Boston, MA (2017), https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/corrigan-gibbs

21. Cramer, R., Damgård, I., Escudero, D., Scholl, P., Xing, C.: Spd$F_{2^k}$: Efficient MPC mod $2^k$ for dishonest majority. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10992, pp. 769–798. Springer (2018). https://doi.org/10.1007/978-3-319-96881-0_26, https://doi.org/10.1007/978-3-319-96881-0_26
22. Damgård, I., Nielsen, J.B.: Universally composable efficient multiparty computation from threshold homomorphic encryption. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 247–264. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
23. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. pp. 643–662. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
24. Davis, H., Patton, C., Rosulek, M., Schoppmann, P.: Verifiable distributed aggregation functions. Proceedings on Privacy Enhancing Technologies **2023**, 578–592 (10 2023). https://doi.org/10.56553/popets-2023-0126
25. Demmler, D., Schneider, T., Zohner, M.: ABY - A framework for efficient mixed-protocol secure two-party computation. In: 22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015. The Internet Society (2015), https://www.ndss-symposium.org/ndss2015/aby---framework-efficient-mixed-protocol-secure-two-party-computation
26. Dong, J., Roth, A., Su, W.J.: Gaussian Differential Privacy. Journal of the Royal Statistical Society Series B: Statistical Methodology **84**(1), 3–37 (02 2022). https://doi.org/10.1111/rssb.12454, https://doi.org/10.1111/rssb.12454
27. Dwork, C.: Differential privacy. In: Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II. p. 1–12. ICALP'06, Springer-Verlag, Berlin, Heidelberg (2006). https://doi.org/10.1007/11787006_1, https://doi.org/10.1007/11787006_1
28. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques. p. 486–503. EUROCRYPT'06, Springer-Verlag, Berlin, Heidelberg (2006). https://doi.org/10.1007/11761679_29, https://doi.org/10.1007/11761679_29
29. Dwork, C., Kohli, N., Mulligan, D.: Differential privacy in practice: Expose your epsilons! Journal of Privacy and Confidentiality **9**(2) (Oct 2019). https://doi.org/10.29012/jpc.689, https://journalprivacyconfidentiality.org/index.php/jpc/article/view/689
30. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. vol. Vol. 3876, pp. 265–284 (01 2006). https://doi.org/10.1007/11681878_14
31. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
32. Dwork, C., Rothblum, G.N.: Concentrated differential privacy. CoRR **abs/1603.01887** (2016), http://arxiv.org/abs/1603.01887
33. Eriguchi, R., Ichikawa, A., Kunihiro, N., Nuida, K.: Efficient noise generation to achieve differential privacy with applications to secure multiparty computation.

In: Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part I. p. 271–290. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-662-64322-8_13, https://doi.org/10.1007/978-3-662-64322-8_13

34. Eriguchi, R., Ichikawa, A., Kunihiro, N., Nuida, K.: Efficient noise generation protocols for differentially private multiparty computation. IEEE Transactions on Dependable and Secure Computing **20**(6), 4486–4501 (2023). https://doi.org/10.1109/TDSC.2022.3227568
35. Erlingsson, U., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: From local to central differential privacy via anonymity. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. p. 2468–2479. SODA '19, Society for Industrial and Applied Mathematics, USA (2019)
36. Escudero, D.: An introduction to secret-sharing-based secure multiparty computation. Cryptology ePrint Archive, Paper 2022/062 (2022), https://eprint.iacr.org/2022/062
37. Escudero, D., Ghosh, S., Keller, M., Rachuri, R., Scholl, P.: Improved primitives for mpc over mixed arithmetic-binary circuits. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 823–852. Springer International Publishing, Cham (2020)
38. Frederiksen, T.K., Keller, M., Orsini, E., Scholl, P.: A unified approach to mpc with preprocessing using ot. In: Proceedings, Part I, of the 21st International Conference on Advances in Cryptology – ASIACRYPT 2015 - Volume 9452. p. 711–735. Springer-Verlag, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_29, https://doi.org/10.1007/978-3-662-48797-6_29
39. Ghazi, B., Ilango, R., Kamath, P., Kumar, R., Manurangsi, P.: Towards separating computational and statistical differential privacy. In: 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS). pp. 580–599 (2023). https://doi.org/10.1109/FOCS57990.2023.00042
40. Ghosh, A., Roughgarden, T., Sundararajan, M.: Universally utility-maximizing privacy mechanisms. SIAM Journal on Computing **41**(6), 1673–1693 (2012). https://doi.org/10.1137/09076828X, https://doi.org/10.1137/09076828X
41. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, USA (2004)
42. Goldwasser, S., Sipser, M.: Private coins versus public coins in interactive proof systems. In: Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing. p. 59–68. STOC '86, Association for Computing Machinery, New York, NY, USA (1986). https://doi.org/10.1145/12130.12137, https://doi.org/10.1145/12130.12137
43. Goyal, V., Mironov, I., Pandey, O., Sahai, A.: Accuracy-privacy tradeoffs for two-party differentially private protocols. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013. pp. 298–315. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
44. Groce, A., Rindal, P., Rosulek, M.: Cheaper private set intersection via differentially private leakage. Proceedings on Privacy Enhancing Technologies **2019**, 6–25 (07 2019). https://doi.org/10.2478/popets-2019-0034
45. Haitner, I., Mazor, N., Silbak, J., Tsfadia, E.: On the complexity of two-party differential privacy. In: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing. p. 1392–1405. STOC 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3519935.3519982, https://doi.org/10.1145/3519935.3519982

46. He, X., Machanavajjhala, A., Flynn, C., Srivastava, D.: Composing differential privacy and secure computation: A case study on scaling private record linkage. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. p. 1389–1406. CCS '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3133956.3134030, https://doi.org/10.1145/3133956.3134030
47. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? SIAM Journal on Computing **40**(3), 793–826 (2011). https://doi.org/10.1137/090756090, https://doi.org/10.1137/090756090
48. Keeler, D., Komlo, C., Lepert, E., Veitch, S., He, X.: Dprio: Efficient differential privacy with high utility for prio. Proceedings on Privacy Enhancing Technologies (2023)
49. Keller, H., Möllering, H., Schneider, T., Tkachenko, O., Zhao, L.: Secure noise sampling for dp in mpc with finite precision. Cryptology ePrint Archive, Paper 2023/1594 (2023), https://eprint.iacr.org/2023/1594
50. Keller, M.: Mp-spdz: A versatile framework for multi-party computation. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. p. 1575–1590. CCS '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3372297.3417872, https://doi.org/10.1145/3372297.3417872
51. Keller, M., Orsini, E., Scholl, P.: Mascot: Faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. p. 830–842. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2976749.2978357, https://doi.org/10.1145/2976749.2978357
52. Kifer, D., Abowd, J.M., Ashmead, R., Cumings-Menon, R., Leclerc, P., Machanavajjhala, A., Sexton, W., Zhuravlev, P.: Bayesian and frequentist semantics for common variations of differential privacy: Applications to the 2020 census (2022)
53. Krehbiel, S.: Choosing epsilon for privacy as a service. Proceedings on Privacy Enhancing Technologies **2019**, 192 – 205 (2019)
54. Lipmaa, H., Toft, T.: Secure equality and greater-than tests with sublinear online complexity. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) Automata, Languages, and Programming. pp. 645–656. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
55. Mazloom, S., Gordon, S.D.: Secure computation with differentially private access patterns. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. p. 490–507. CCS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3243734.3243851, https://doi.org/10.1145/3243734.3243851
56. McGregor, A., Mironov, I., Pitassi, T., Reingold, O., Talwar, K., Vadhan, S.: The limits of two-party differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science. pp. 81–90 (2010). https://doi.org/10.1109/FOCS.2010.14
57. Mehner, L., von Voigt, S.N., Tschorsch, F.: Towards explaining epsilon: A worst-case study of differential privacy risks. 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) pp. 328–331 (2021)
58. Mironov, I., Pandey, O., Reingold, O., Vadhan, S.: Computational differential privacy. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009. pp. 126–142. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

59. Nishide, T., Ohta, K.: Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In: Okamoto, T., Wang, X. (eds.) Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4450, pp. 343–360. Springer (2007). https://doi.org/10.1007/978-3-540-71677-8_23, https://doi.org/10.1007/978-3-540-71677-8_23
60. Rotaru, D., Wood, T.: Marbled circuits: Mixing arithmetic and boolean circuits with active security. In: Progress in Cryptology – INDOCRYPT 2019: 20th International Conference on Cryptology in India, Hyderabad, India, December 15–18, 2019, Proceedings. p. 227–249. Springer-Verlag, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-35423-7_12, https://doi.org/10.1007/978-3-030-35423-7_12
61. Roth, E., Noble, D., Falk, B.H., Haeberlen, A.: Honeycrisp: large-scale differentially private aggregation without a trusted core. In: Proceedings of the 27th ACM Symposium on Operating Systems Principles. p. 196–210. SOSP '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3341301.3359660, https://doi.org/10.1145/3341301.3359660
62. Roy Chowdhury, A., Wang, C., He, X., Machanavajjhala, A., Jha, S.: Crypte: Crypto-assisted differential privacy on untrusted servers. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. p. 603–619. SIGMOD '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3318464.3380596, https://doi.org/10.1145/3318464.3380596
63. Schoppmann, P., Vogelsang, L., Gascón, A., Balle, B.: Secure and scalable document similarity on distributed databases: Differential privacy to the rescue. Proceedings on Privacy Enhancing Technologies **2020**, 209 – 229 (2020)
64. Steinke, T.: Multi-central differential privacy (2020)
65. Vadhan, S.: The Complexity of Differential Privacy, pp. 347–450. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-57048-8_7, https://doi.org/10.1007/978-3-319-57048-8_7
66. Wei, C., Yu, R., Fan, Y., Chen, W., Wang, T.: Securely sampling discrete gaussian noise for multi-party differential privacy. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. p. 2262–2276. CCS '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3576915.3616641, https://doi.org/10.1145/3576915.3616641

# A Other CDP definitions

Below we recall the definitions of IND-CDP and SIM-CDP from [58], although reformulated to fit our notation. Let $\mathbf{x}$ and $\mathbf{y}$ denote the inputs of party 1 and party 2 respectively. That is, $D := \mathbf{x}||\mathbf{y} \in \mathcal{D}$.

**Definition 11 (IND-CDP of two-party protocols [58]).** *An interactive protocol ensemble $\{\langle g^1_\kappa(\cdot), g^2_\kappa(\cdot)\rangle\}_{\kappa\in\mathbb{N}}$ ensures for $\{g^1_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ $\underline{\varepsilon_\kappa\text{-IND-CDP}}$ if for every ensemble $\{\tilde{g}^2_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ of efficiently computable randomised interactive functions, and all sufficiently large $\kappa$, it holds that the ensemble $\{\mathsf{VIEW}_{\kappa,\tilde{g}^2_\kappa}(\mathbf{x})\}_{\kappa\in\mathbb{N}}$ provides $\varepsilon_\kappa$-IND-CDP (as in Definition 12) with respect to $\mathbf{x}$. The definition is symmetric for $\{g^2_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$.*

**Definition 12 (IND-CDP of interactive functions [58]).** *An interactive function ensemble $\{g_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ of randomised functions $g_\kappa : \mathcal{D} \to \mathcal{R}_\kappa$ provides $\underline{\varepsilon_\kappa\text{-IND-CDP}}$ if there exists a negligible function $negl(\cdot)$ such that for every non-uniform PPT Turing Machine $A$, every polynomial $p(\cdot)$, every sufficiently large $\kappa$, all datasets $D, D' \in \mathcal{D}$ of size at most $p(\kappa)$ differing in at most one row, and every advice string $z_\kappa$ of size at most $p(\kappa)$, it holds that*

$$\mathbb{P}(A_\kappa(g_\kappa(D)) = 1) \leq e^{\varepsilon_\kappa}\mathbb{P}(A_\kappa(g_\kappa(D')) = 1) + negl(\kappa),$$

*where we write $A_\kappa(x)$ for $A(1^\kappa, z_\kappa, x)$ and the probability is taken over the randomness of mechanism $g_\kappa$ and the adversary $A_\kappa$.*

**Definition 13 (SIM-CDP of two-party protocols [58]).** *An interactive protocol ensemble $\{\langle g^1_\kappa(\cdot), g^2_\kappa(\cdot)\rangle\}_{\kappa\in\mathbb{N}}$ ensures for $\{g^1_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ $\underline{\varepsilon_\kappa\text{-SIM-CDP}}$ if for every ensemble $\{\tilde{g}^2_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ of efficiently computable randomised interactive functions, there exists an ensemble $\{\mathcal{S}_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ of $\varepsilon_\kappa$-DP mechanisms $\mathcal{S}_\kappa(\cdot)$ such that for every $\mathbf{x}$, the probability ensembles $\{\mathsf{VIEW}_{\kappa,\tilde{g}^2_\kappa}(\mathbf{x})\}_{\kappa\in\mathbb{N}}$ and $\{\mathcal{S}_\kappa(\mathbf{x})\}_{\kappa\in\mathbb{N}}$ are computationally indistinguishable.*

# B The arithmetic black-box

In Figure 6 we present the ideal functionality $\mathcal{F}_{ABB}$ of the arithmetic black-box. The ABB is at times formulated slightly differently, such as only operating within the arithmetic domain, not including conversions between the domains or including conversions in both directions in between the binary and arithmetic representations. We choose the flavor of ABB that is used in [37], simply because it includes the operations we need but nothing more. For more details on the ABB see, for instance, [22, 54].

**Functionality $\mathcal{F}_{ABB}$**

**Parameters:** A modulus $q$ that is either a prime or a power of 2.

**Input:**

- Upon input $(\mathsf{Input}, P_i, \mathsf{type}, \mathsf{id}, x)$ from $P_i$ and input $(\mathsf{Input}, P_i, \mathsf{type}, \mathsf{id})$ from all other parties, if $\mathsf{id}$ is a fresh identifier and $(\mathsf{type}, x) \in \{\{\mathsf{binary}\} \times \mathbb{Z}_2, \{\mathsf{arithmetic}\} \times \mathbb{Z}_q)\}$ then store $(\mathsf{type}, \mathsf{id}, x)$.

**Linear combination:**

- Upon input $(\mathsf{LinComb}, \mathsf{type}, \mathsf{id}, (\mathsf{id}_j)_{j=1}^m, c, (c_j)_{j=1}^m)$ from all parties, if
  - each $\mathsf{id}_j$ is stored in memory, and,
  - $c, c_j \in \mathbb{Z}_2$ if $\mathsf{type}$ is $\mathsf{binary}$ and $c, c_j \in \mathbb{Z}_q$ if $\mathsf{type}$ is $\mathsf{arithmetic}$,

  then
  - retrieve $((\mathsf{type}, \mathsf{id}_1, x_1), \ldots, (\mathsf{type}, \mathsf{id}_m, x_m))$,
  - compute $y \leftarrow \sum c_j \cdot x_j \mod 2$ if $\mathsf{type}$ is $\mathsf{binary}$ and $y \leftarrow \sum c_j \cdot x_j \mod q$ if $\mathsf{type}$ is $\mathsf{arithmetic}$,
  - store $(\mathsf{type}, \mathsf{id}, y)$.

**Multiplication:**

- Upon input $(\mathsf{Mult}, \mathsf{type}, \mathsf{id}, \mathsf{id}_1, \mathsf{id}_2)$ from all parties, if $\mathsf{id}_1, \mathsf{id}_1$ are stored in memory then
  - retrieve $(\mathsf{type}, \mathsf{id}_1, x_1), (\mathsf{type}, \mathsf{id}_2, x_2))$,
  - compute $y \leftarrow x_1 \cdot x_2 \mod 2$ if $\mathsf{type}$ is $\mathsf{binary}$ and $y \leftarrow x_1 \cdot x_2 \mod q$ if $\mathsf{type}$ is $\mathsf{arithmetic}$,
  - store $(\mathsf{type}, \mathsf{id}, y)$.

**Converting from binary to arithmetic:**

- Upon input $(\mathsf{ConvertB2A}, \mathsf{id}, \mathsf{id}')$ from all parties, if $\mathsf{id}'$ is present in memory then retrieve $(\mathsf{binary}, \mathsf{id}', x)$ and store $(\mathsf{arithmetic}, \mathsf{id}, x)$.

**Output:**

- Upon input $(\mathsf{Output}, \mathsf{type}, \mathsf{id})$ from all honest parties, if $\mathsf{id}'$ is present in memory then retrieve $(\mathsf{type}, \mathsf{id}, x)$ and output it to the adversary. Wait for input from the adversary of the form $(\mathsf{Deliver}, b)$, where $b \in \mathbb{Z}_2$. if $b = 1$ then output $x$ to all parties, otherwise output $\perp$.

Fig. 6: The ideal functionality for the arithmetic black-box.

# C Proofs

## C.1 Proof of Lemma 1

*Proof.* Let $Z \sim \mathcal{M}^{p,f,\lambda}_{RTGeo}(D)$ and $Y \sim \mathcal{M}^{2B,f,\lambda}_{SRTGeo}(D)$ for arbitrary $\lambda, D$. Let $p_Z$ and $p_Y$ denote the probability density functions of $Z$ and $Y$ respectively and let $F$ denote their cumulative distribution functions in the same manner. Since the parameter restrictions guarantee that the final sum in $Y$ does not overflow (the result is as if the sum was done over the integers), the statistical distance between the two distributions is exactly twice the total probability mass that is affected by the truncation in $Y$. That is,

$$
\begin{aligned}
SD(Z,Y) &= \frac{1}{2} \sum_{z \in \mathbb{Z}_p} |p_X(z) - p_Y(z)| \\
&= \sum_{z \in \mathbb{Z}_p \setminus (\bar{f}-B, \bar{f}+B)} |p_X(z) - p_Y(z)| \\
&= |F_X(\bar{f} - B) + (1 - F_X(\bar{f} + B))| \\
&= \left| \frac{e^{1/\lambda}}{e^{1/\lambda}+1} e^{-(\bar{f}-\bar{f}+B)/\lambda} \right. \\
&+ \left. \frac{1}{e^{1/\lambda}+1} e^{-(\bar{f}+B-\bar{f})/\lambda} \right| \\
&= e^{-B/\lambda},
\end{aligned}
$$

where $\bar{f}$ is shorthand for $f(D)$. The equalities follow by inserting the formulas from Definition 6 and direct simplifications.

□

## C.2 Proof of Lemma 2

*Proof.* Firstly, $\mathtt{Ber}_{\hat{\alpha}}$ exactly samples a Bernoulli trial with parameter equal to the recomposition of the first $d$ elements of $\alpha$. Call this parameter value $\alpha'$. This means that the statistical distance between $\mathtt{Ber}(\hat{\alpha})$ and an exact Bernoulli trial with parameter $\hat{\alpha}$ is the same as between two exact Bernoulli trials with parameter $\hat{\alpha}$ and $\alpha'$, respectively. This statistical distance is equal to $|\hat{\alpha} - \alpha'|$, which is at most $2^{-d}$ since the first $2^d$ bits of their decomposition are identical. Secondly, the statistical distance between $\mathcal{M}^{\lambda,B,d,h}_{\mathtt{FDL}}(D)$ and $\mathcal{M}^{2B,h,\lambda}_{SRTGeo}(D)$ is at most equal to the probability of any of the Bernoulli trials being incorrect, which due to independence is at most $B2^{-d}$.

□

### C.3 Proof of Lemma 3

*Proof.* The additive usefulness follows from a standard tail bound on the geometric distribution, since the truncated geometric is at least as concentrated as the untruncated one:

$$\begin{aligned}\mathbb{P}(|Geo_{q,\lambda}(f(D)) - f(D)| \geq \nu) &= \mathbb{P}(|Geo_{q,\lambda}(0)| \geq \nu)\\ &\leq \mathbb{P}(|Geo_{\lambda}(0)| \geq \nu)\\ &= 2F_{Geo_{\lambda}(0)}(-\nu)\\ &= \frac{2e^{1/\lambda}}{e^{1/\lambda}+1}e^{-\nu/\lambda}.\end{aligned}$$

We may note that the same result holds regardless of how narrowly the truncation is done.

□

## D Techniques for achieving secure MPC

In the context of MPC, we typically distinguish binary and arithmetic protocols. This classification describes the possible computations. In other words, we perform addition and multiplication in $\mathbb{F}_2$ and $\mathbb{F}_p$, respectively. In this work, we rely on secret sharing-based (SS) MPC protocols. More precisely, we use additive secret sharing (ASS). In the following, we will use notation for addition and multiplication, referring to the *XOR* and *AND* operations in the binary domain. In such protocols, secret values $x$ are shared among $n$ parties by sampling $n-1$ random values $x_1, \ldots, x_{n-1} \leftarrow \mathcal{U}(\mathbb{F})$, setting $x_0 \leftarrow x - \sum_{i=1}^{n} x_i$, and distributing $x_i$ to every party $p_i$. We denote secret shared values as $[\![x]\!]$. We further denote $[\![x]\!] \leftarrow \mathsf{Share}(x)$, and $x \leftarrow \mathsf{Reconstruct}([\![x]\!])$ as sharing and reconstructing secrets. ASS schemes are additively homomorphic, allowing the addition of shares without interaction and hiding underlying secrets as long as there is one honest party. To allow multiplications with an ASS, one can use multiplication triples, introduced by Beaver [4]. Triples are three shared values $([\![a]\!], [\![b]\!], [\![c]\!])$, that no party knows and that fulfil $a \cdot b = c$. When multiplying two shared values $([\![x]\!], [\![y]\!])$, one reconstructs masked versions $\alpha \leftarrow \mathsf{Reconstruct}([\![x]\!] - [\![a]\!])$, $\beta \leftarrow \mathsf{Reconstruct}([\![y]\!] - [\![b]\!])$, and computes[14] $[\![z]\!] = \alpha\beta + \beta[\![x]\!] + \alpha[\![y]\!] + [\![c]\!] = [\![x \cdot y]\!]$. Given these ingredients, we can instantiate a malicious secure MPC protocol if we have access to a secure sampling method for multiplication triples, and adversaries cannot tamper with the reconstruction procedure. In the SPDZ paper [23], the authors introduced solutions to both problems. They propose an additively homomorphic encryption scheme for sampling triples and information-theoretic message authentication codes (MACs) to secure the reconstruction procedure. Subsequent work introduced several performance improvements by instantiating

[14] This step requires multiplication and addition with constant terms which follows from the ASS properties.

the ASS over the ring $\mathbb{F}_{2^k}$ [21] or replacing the expensive homomorphic encryption with oblivious transfer [51]. Note that both improvements, to some degree, accept a higher communication for a lower computation complexity.