Practical Two-party Computational Differential Privacy with Active Security.

Fredrik Meisingseth[1,2], Christian Rechberger[1], and Fabian Schmid[1]

[1] Graz University of Technology, Graz, Austria
firstname.lastname@iaik.tugraz.at
[2] Know-Center GmbH, Graz, Austria

Abstract. Distributed models for differential privacy (DP), such as the local and shuffle models, allow for differential privacy without having to trust a single central dataholder. They do however typically require adding more noise than the central model. One commonly iterated remark is that achieving DP with similar accuracy as in the central model is directly achievable by *emulating the trusted party*, using general multiparty computation (MPC), which computes a canonical DP mechanism such as the Laplace or Gaussian mechanism. There have been a few works proposing concrete protocols for doing this but as of yet, all of them either require honest majorities, only allow passive corruptions, only allow computing aggregate functions, lack formal claims of what type of DP is achieved or are not computable in polynomial time by a finite computer. In this work, we propose the first efficiently computable protocol for emulating a dataholder running the geometric mechanism, and which retains its security and DP properties in the presence of dishonest majorities and active corruptions. To this end, we first analyse why current definitions of computational DP are unsuitable for this setting and introduce a new version of computational DP, SIM*-CDP. We then demonstrate the merit of this new definition by proving that our protocol satisfies it. 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 of our protocol 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)* [44], the *shuffle model* [9, 19] and definitions with a computationally bounded adversary, giving guarantees of *computational DP (CDP)* [27, 7, 56]. Each of the different 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, must add much more noise than the standard central model of

DP [44, 33, 18, 7]. One commonly aired remark is that one can use *general 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 [30, 19]. 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 [7, 32, 59], only allowing aggregate functions [21, 45] and/or requiring honest majorities [27]. In this work, we focus on the case of two parties, active (static) corruptions, and require efficient protocols[3] that achieve the same accuracy as in the central model. In particular, we aim for a protocol in which two parties together compute a version of the geometric (discrete Laplace) mechanism [38, 5] that also allows non-aggregate queries.

In order to design practical protocols for GED, we need a CDP 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 [29], geometric, Gaussian [30] and discrete Gaussian [16] mechanisms are not computable exactly in probabilistic polynomial time (PPT) on a finite computer. This means that, since general MPC only allow PPT computable functionalities, the used definition needs to allow either that the protocol does not exactly emulate the dataholder 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 general information-theoretic MPC is impossible [20, 40, 34], the only candidates of a suitable DP definition are the CDP notions introduced in [56]. Since we will refer to it recurrently, let us call the paper [56] *MPRV*, after its authors.

Of the CDP definitions, IND-CDP and SIM-CDP (Definitions 6 and 7 in MPRV) are defined such that they allow protocols that are not efficiently computable, precisely in order to allow protocols for inefficient DP mechanisms. Further, there are, to the best of our knowledge, barely any results under what conditions the definitions are fulfilled as a consequence of security properties of common MPC protocols. Therefore, using these definitions when aiming for GED would require either significant adjustment of the definitions or new theoretic results. The third CDP notion in MPRV, SIM^+-CDP, on the other hand does not allow inefficient protocols and its fulfillment is derivable directly from standard notions of security in MPC. Still, SIM^+-CDP requires that the emulation of the dataholder has perfect honest correctness, which implies that the definition is not satisfiable for

[3] In particular, we require that the protocols are computable in strict polynomial time in a finite computational model, as suggested in [5].

non-PPT mechanisms, and one would need to instantiate it with a finite version of them, for instance using the mechanisms introduced in [5]. 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'. Further, SIM$^+$-CDP is defined using the security definition of [40] while the dominating security definition in MPC currently, arguably, is UC-security [14, 15]. Therefore, realising GED would be made more direct if the CDP notion is rather phrased using the security notion used by state-of-the-art MPC schemes, such as [25, 22, 36, 2].

With this motivation, we propose a new instantiation of SIM$^+$-CDP, which we call SIM*-CDP, and we give a generic protocol for satisfying it by means of a truncated geometric mechanism. Further, we implement the protocol, use it to compute differentially private inner-products and benchmark the implementation, hence showing its practical performance.

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 dataholder that computes an inefficient DP mechanism. With these difficulties in mind we present a new version of SIM$^+$-CDP, which we call SIM*-CDP(Section 3).
- We demonstrate the usability of the new definition by showing how it can be achieved for the geometric mechanism (Section 4) and give an efficient generic MPC protocol (Section 5).
- We use the protocol to compute differentially private two-party inner-products with security against dishonest majorities of active adversaries, being the first to do so with accuracy equal to that in the central model, and provide an open-sourced implementation[4]. We provide benchmarks of the implementation and thereby show that it is efficient in practice (Section 6).

Related works. One popular definition of distributed DP, let us call it *BNO-DP*,[5] was introduced in [7] and then used in [59, 10, 31, 54]. The idea is to define distributed DP by that the view of a passive adversary is (ε, δ)-DP with respect to the input of the honest parties. This information-theoretic definition is particularly suitable when dealing with information-theoretic MPC schemes that have perfect correctness. For general MPC, this necessitates an honest majority (see, for instance, [39, 23, 34]). This means that for the case of dishonest majorities, the functionality needs to be restricted somehow and the fulfillment of the BNO-DP definition must be proven anew. Alternatively, a different definition must be used, for instance one of the CDP notions of MPRV [56]. These definitions have

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

[5] The name is after the initials of the authors of the paper in which it is proposed. More precisely, we refer to Definition 2 of the CRYPTO version of the paper.

seen use both in practice [45, 58, 32] and in theory, perhaps most significantly in a line of work separating statistical and computational DP [54, 42, 43, 37, 11].

The first work that aims to emulate a central trusted party for DP by use of MPC is *Our data, ourselves* [27], 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 [27], other works have also proposed noise sampling protocols for DP in an MPC setting [3, 17, 32] and perhaps the work most related to ours is EIKN [32]. EIKN gives an efficient distributed 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. Further, in a very recent preprint [46], the authors provide an efficient noise sampling protocol for passive corruptions and claim security for dishonest majorities. The authors of [46] 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 [47], but make no note of the type of CDP this could result in. In that sense, our proposed SIM*-CDP definition offers a beginning of an answer to that.[6]

2 Preliminaries

2.1 Differential privacy

The notion of *differential privacy (DP)* [29, 1] considers a probabilistic function, algorithm, or *mechanism*, that maps *databases*, i.e. sets of elements from some data universe χ, 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 [1]):

Definition 1 (ε-DP [29, 1]). *A probabilistic function $\mathcal{M} : \chi^{N'} \to R$ is ε-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.

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

[6] Another related paper is [3]. 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 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.

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 *dataholder* (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 (two-party DP or DP in the two-server model) 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). Then these two computational parties together wish to compute the query h 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 and further details on the model can be found in [56, 8].

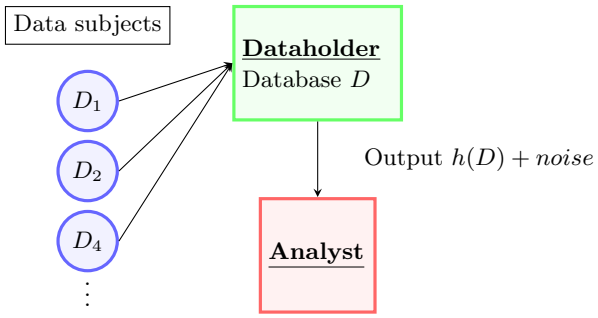

Fig. 1: In the *central model*, the data subjects trust the data holder with their data (D_i) but wish to keep it secret from an (possibly adversarial) analyst learning the (possibly noisy) function evaluation.

When discussing DP mechanisms, it is critical to consider the usefulness of the mechanism for approximating the query function h. We do this by using the following notion of usefulness, which is a reformulation of the notion of usefulness in MPRV [56] to consider probabilistic functions rather than interactive protocol ensembles.

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

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 $\nu(\kappa)$.

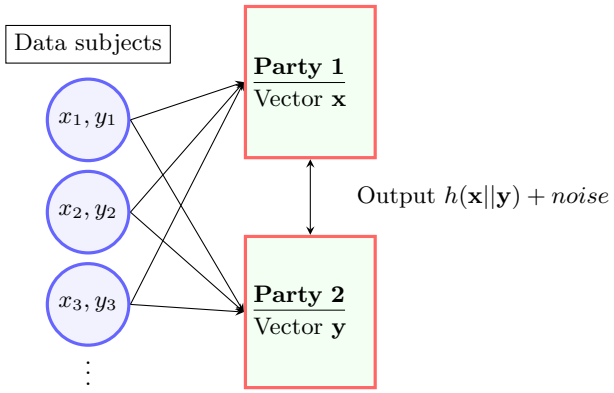

Fig. 2: In the *two-party model*, the data subjects trust two different data holders, which we call *parties*, with a different part of their data, but not with the part of the data that they send to the other data holder. Both parties then learns the noisy function evaluation. Thus, in a sense, each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

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 C. In general, MPC schemes in $\mathbb{F}_p$ provide fast algorithms for addition and multiplication. In contrast, in $\mathbb{F}_{2^k}$, comparisons, bitwise 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 [26], the authors propose a semi-honest two-party MPC 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* [57] 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)* [35], where masking values are shared along with their binary decomposition in the respective domains. The eda-bits repre-

sent an improvement in efficiency when converting larger values, and [35] presents dedicated protocols to speed up comparisons in $\mathbb{F}_p$.

3 A new version of simulation-based CDP

3.1 The original SIM$^+$-CDP definition

The intuition behind SIM$^+$-CDP is that if the protocol execution is indistinguishable from an ideal process (secure by definition) computing a differentially private mechanism then the protocol is also to be seen as being differentially private. Thus, SIM$^+$-CDP is a more direct realisation of GED than IND-CDP and SIM-CDP, which do not have the same clear separation between functionality (mechanism) emulation and the DP properties of the mechanism itself. The definition of SIM$^+$-CDP in MPRV [56] is the following[7].

Definition 3 (Definition 8 in MPRV [56]). *An interactive protocol ensemble* $\{\langle f_\kappa(\cdot), g_\kappa(\cdot)\rangle\}_{\kappa\in\mathbb{N}}$ *is a* $(s,\nu)\varepsilon_\kappa$*-SIM*$^+$*-CDP private two-party computation protocol for* $h = (h_f, h_h)$ *with respect to the predicate* P *if there exists an* ε_κ*-DP randomized mechanism* $\hat{h} = (\hat{h}_f, \hat{h}_g)$ *such that*

- *Mechanism* $\hat{h}$ *provides* (s,ν)*-usefulness for* h *with respect to the predicate* P*.*
- *The protocol ensemble is a secure two-party computation protocol ensemble for the randomized functionality* $\hat{h}$ *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 [13, 40]. The full version of MPRV [56][8] 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 [40], with the sole adjustments that the simulator is not required to be efficient, and that the output to the honest party is not included in random variables that are compared between the real and ideal worlds. In particular, 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 honest correctness*.

Infeasibility of GED with the Laplace mechanism in SIM$^+$-CDP. The definition of SIM$^+$-CDP is quite intuitive and also very general, for instance, in that it is agnostic to the model of computation and allows for inefficient mechanisms and simulators. There are, however, some possible changes that could make it easier to use whilst still, arguably, capturing the spirit of the original

[7] For definitions of interactive functions, we refer to [41], 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.

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

definition.

As an illustrative example of the difficult use of the definition, consider using it to realise 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 problems lie in 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 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 [38] truncated to the output domain, cannot be realised in SIM$^+$-CDP, since it requires sampling of probabilities that are not multiples of $2^{-poly(\kappa)}$.

3.2 Our new definition, SIM*-CDP.

We propose a new version of SIM$^+$-CDP, which we call SIM*-CDP. It is essentially the same as SIM$^+$-CDP but with the following changes:

- *The mechanism and ideal functionality are separated.* We allow that there is a negligible statistical distance between the functionality of the protocol and the DP mechanism. This is done to allow emulating non-PPT mechanisms.
- *Interactive functions are replaced by ITMs.* The computational model is fixed, this is done because we want to directly use a security framework that only allows efficient protocols and thus the usefulness for agnosticism to the computational model is removed.
- *UC-security is used as security notion.* We use UC-security partly because this better represents trends within the field of MPC and partly because it opens up for more nuanced studies of composition.
- *DP preservation is required also when there are active corruptions.* The use of UC-security creates the need to more thoroughly define what types of influence the adversary can be allowed to have on the outputs to the parties, also in the ideal world.

Separating the mechanism and ideal functionality. As outlined in the previous subsection, one main hurdle in using SIM$^+$-CDP to achieve GED for inefficient DP mechanisms is that the definition would need to relax either the demand for an efficient protocol or the demand for perfect correctness. Since we want a protocol that can be readily implemented in practice, we choose to relax the correctness and therefore introduce a separation between the ideal functionality of the protocol and the DP mechanism. By this separation, we mean partly a literal separation within the model and definition but perhaps more significantly a separation in the sense of requiring statistical (or computational)

closeness between the distributions in question rather than them being identical. We introduce this slack in the shape of statistical indistinguishability.[9]

Use UC-security. Since the publication of MPRV [56] in 2009, the dominating framework for defining secure computation has, arguably, become the *Universal Composability (UC)* framework by Canetti [14, 15], rather than [13, 40]. There have been many constructions in the last decade or so that are both practically efficient and proven secure in the UC framework, such as [47, 25, 51]. Thus, it might be suitable to use a notion of secure computation that is based on, or at least explicitly implied by, UC-security.[10] In the following, we use the definitions from [14, 15] if not otherwise specified. In particular, all machines are ITMs (as in Definition 4 in [14, 15]).

DP preservation under active corruptions. In SIM$^+$-CDP, the choice not to include correctness or DP requirements for malicious executions come with the convenient consequence that it is clear how to define the function that is computed by the ideal functionality – it is simply the output it returns to the parties on an honest execution. One reason for this simplicity is that the security model used in SIM$^+$-CDP is explicitly only for secure function evaluation (SFE).

UC-security, however, allows ideal functionalities for many more tasks than SFE and with much more nuanced adversarial influence in the ideal world. For us, this creates the need to 1) define which of the messages sent from the ideal functionality we require to be (indistinguishable from) DP, and 2) define the allowed type of influence of active corruptions on those messages. First, we choose to require that it is the concatenation of the contents of the *messages to the parties*, $OUT_{\mathcal{F},\mathcal{S}}(D) = (OUT^1_{\mathcal{F},\mathcal{S}}(D), OUT^2_{\mathcal{F},\mathcal{S}}(D))$, in an honest execution that is close to the DP mechanism $\hat{h}_\kappa$.[11]

Secondly, we require that when there are active corruptions then for each ideal-world adversary there is an ε_κ-DP mechanism such that the outputs to the parties are statistically indistinguishable from *that* mechanism, which we denote

[9] The main reason for choosing this place for introducing the slack, instead of allowing inefficient ideal functionalities in the security model, is that it allows us to keep the notion of secure computation intact, which is a main argument for the practical usability of the definition.

[10] In the phd thesis of Balcer [4], he also designs MPC protocols with UC security with the motivation of computing DP mechanisms in MPC. However, the protocols are not used or analysed for this purpose and there is no discussion on under which conditions UC security suffice for fulfilling the CDP notions in MPRV. Providing such an analysis is an interesting open problem.

[11] Note that this includes the messages to the corrupted party, but not messages to the environment. This is essentially analogous to the corresponding modelling in the security model of SIM$^+$-CDP.

$\tilde{h}_{\kappa,\mathcal{S}}$. We call this property *DP preservation under active corruptions*. For example, if $\hat{h}_\kappa$ is the geometric mechanism, it is allowed that active corruptions make the outputs to the parties wildly different to that of the geometric mechanism, but they must remain DP (with the same parameters) in the sense that there is a DP mechanism of the same parameters as $\hat{h}_\kappa$ which the output distribution is close to. Additionally, this new mechanism can be different for each corruption strategy. Further motivation for the need to define this property explicitly is found in Appendix A.1.

The requirement for DP preservation under active corruptions might seem too demanding at first. We can however note that it is an implicit property in SIM$^+$-CDP since the only effect active corruptions can have on the output to the honest party, in the ideal world, is to abort. This choice to abort can be made as a function of $\hat{h}_\kappa$, but since $\hat{h}_\kappa$ is DP by definition, the post-processing property of DP implies that so is the adversary's choice whether or not to abort. For the setting of SIM*-CDP, we can assure the DP preservation via similar arguments, by realising SFE using the *Arithmetic Black-Box (ABB)* model for MPC [35, 53, 24]. We prove this in Section 4.

Now we are ready to introduce our new variant of simulation-based CDP.

Definition 4 (ε_κ−SIM*-CDP). *Let π be a PPT two-party protocol and $\mathcal{F}$ be a PPT ideal functionality. We say that π is a ν-useful ε_κ−SIM*-CDP two-party computation protocol for the probabilistic function ensemble $\{h_\kappa : \mathcal{D} \to R\}_{\kappa\in\mathbb{N}}$ with respect to the predicate P if there exists an ensemble of ε_κ-DP mechanisms $\{\hat{h}_\kappa : \mathcal{D} \to R\}_{\kappa\in\mathbb{N}}$ such that*

- *The protocol π UC-realises $\mathcal{F}$.*
- *The ensemble $\{\hat{h}_\kappa\}$ provides ν-usefulness for $\{h_\kappa\}$ with respect to the predicate P, in the sense of Definition 2.*
- *For all passive $\mathcal{S}$ and for all $D \in \mathcal{D}$, the probability distribution ensembles of $\{\hat{h}_\kappa(D)\}$, $OUT^1_{\mathcal{F},\mathcal{S}}(D)$ and $OUT^2_{\mathcal{F},\mathcal{S}}(D)$ are statistically indistinguishable.*
- *For each active $\mathcal{S}$, there exists an ensemble of ε_κ-DP mechanisms $\{\tilde{h}_{\kappa,\mathcal{S}} : \mathcal{D} \to R\}_{\kappa\in\mathbb{N}}$ such that for all $D \in \mathcal{D}$, the probability distribution ensembles of $\{\tilde{h}_{\kappa,\mathcal{S}}(D)\}$, $OUT^1_{\mathcal{F},\mathcal{S}}(D)$ and $OUT^2_{\mathcal{F},\mathcal{S}}(D)$ are statistically indistinguishable.*

More remarks and discussion about details in the definition can be found in Appendix A.2.

4 A SIM*-CDP version of the geometric mechanism

As a generic method for achieving SIM*-CDP, we propose to use a range-truncated geometric mechanism. The core part of the method is, naturally, to sample a distribution that is statistically indistinguishable from a range-truncated geometric distribution. Such a truncated geometric distribution can be found in [38, 5, 32], 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 5 (Truncated geometric distribution). *Define the truncated geometric distribution $Z \sim Geo_{q,\lambda}(\bar{h})$ centered at $\bar{h} \in \mathbb{Z}_q$, truncated to $\mathbb{Z}_q := [-q/2, q/2)$, by its pmf:*

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

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

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

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

Definition 6 (Range-truncated geometric mechanism). *Let $\lambda \in \mathbb{N}^{-1}$ and let $h : \mathcal{D} \to \mathbb{Z}_q$ be a deterministic function. The Range-truncated geometric mechanism over $\mathbb{Z}_q$ for h is defined as $\mathcal{M}^{q,h,\lambda}_{RTGeo}(D) := Geo_{q,\lambda}(h(D))$.*

It is easy to verify that $\mathcal{M}^{q,h,\lambda}_{RTGeo}(D)$ is an ε-DP mechanism as long as $\lambda = \frac{\varepsilon}{\Delta h}$. In line with [5], 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 $h(D)$ 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 7 (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 $h : \mathcal{D} \to \mathbb{Z}_q$, be defined as $\mathcal{M}^{2B,h,\lambda}_{SRTGeo}(D) := h(D) + Geo_{2B,\lambda}(0)$, with the addition performed over $\mathbb{Z}_q$.*

In the 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 B.1.

Lemma 1. *Let $h^{max} := \max_{D \in \mathcal{D}} |h(D)|$, $B \in \mathbb{N}$, $\lambda \in \mathbb{N}^{-1}$ and $q > 2h^{max} + 2B$. Then the statistical distance between $\mathcal{M}^{2B,h,\lambda}_{SRTGeo}(D)$ and $\mathcal{M}^{q,h,\lambda}_{RTGeo}(D)$ 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 h. The trouble still remains that the probabilities might not be negative polynomial powers of two. In [27, 32] 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 [32].

Definition 8 (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 $\texttt{FDL}_{\lambda,B,d} : \{0,1\}^{Bd+1} \to \mathbb{Z}_{2B} \subseteq \mathbb{Z}_q$ *by the following procedure:*

Procedure FDL

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 $\sigma \cdot l$.

Let $\alpha = (\alpha_1, \alpha_2, ...)$ *be the bit decomposition of* $\hat{\alpha}$. *The subprocedure* $\texttt{Ber}_{\hat{\alpha}} : \{0,1\}^d \to \{0,1\}$ *for generating approximate Bernoulli trials with parameter* $\hat{\alpha}$ *is defined by:*

Procedure Ber

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 β.

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 κ then FDL runs in strict polynomial time, and thus it can be computed by an ideal functionality that can be UC-realised. With some abuse of notation, we use FDL to denote both the procedure and the probability distribution it generates upon being given fair coins.[12]

Definition 9 (FDL mechanism). *Let $B \in \{1, \ldots, \lceil q/2 \rceil - 1\}$. Let the Finite Discrete Laplace mechanism over $\mathbb{Z}_q$ for a function $h : \mathcal{D} \to \mathbb{Z}_q$ be defined as $\mathcal{M}_{\texttt{FDL}}^{\lambda,B,d,h}(D) := h(D) + \texttt{FDL}_{\lambda,B,d}$, with the addition performed over $\mathbb{Z}_q$.*

The following lemma is proven in EIKN [32]. For completeness, we also include a proof in Appendix B.2.

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

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

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

In Figure 3 is an ideal functionality for the FDL mechanism, where $\mathcal{D} := \mathbb{Z}_q^{2N}$. In essence it is a restriction of the ABB to the case of performing SFE of the FDL function.

Note that $\mathcal{F}_{\mathcal{M}_{\texttt{FDL}}}$ is obviously PPT as long as $\mathcal{F}_h$ is. In the following lemma we state that the ideal functionality above computes the range-truncated geometric mechanism in the sense required by the SIM*-CDP definition when the ideal-world adversary is passive and that the ideal functionality also is DP preserving under active corruptions. The proof is found in Appendix B.4.

Lemma 4. *Let $q > 2h^{max} + 2B$, $B \geq 1$ and $\lambda = \frac{\varepsilon}{\Delta h}$ and let $e^{-B/\lambda}$ and $B2^{-d}$ be negligible in κ. Let $D := \mathbf{x}||\mathbf{y}$ and let $g : \mathbb{Z}_q^{2N} \to \mathbb{Z}_q$ be an arbitrary deterministic function with $h^{max} := \max_{D \in \mathbb{Z}_q^{2N}} |h(D)|$ and let $\hat{h}(D)$ be $\mathcal{M}_{RTGeo}^{q,g,\lambda}(D) : \mathbb{Z}_q^{2N} \to \mathbb{Z}_q$. Let $\mathcal{F}$ denote $\mathcal{F}_{\mathcal{M}_{\texttt{FDL}}}$. Then:*

- *For all passive $\mathcal{S}$ and for all $D \in \mathbb{Z}_q^{2N}$, the probability distributions $\hat{h}(D)$, $OUT_{\mathcal{F},\mathcal{S}}^1(D)$ and $OUT_{\mathcal{F},\mathcal{S}}^2(D)$ are statistically indistinguishable.*

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

Functionality $\mathcal{F}_{\mathcal{M}_{\texttt{FDL}}}$

Parameters: Natural numbers B, d, λ^{-1}, q, N and an efficiently computable function $h : \mathbb{Z}_q^{2N} \to \mathbb{Z}_q$.
Commands:

- **Input P_1:** Upon $(Input, P_1, id, \mathbf{x})$ from P_1 and $(Input, P_1, id)$ from P_2 with a fresh identifier id and $\mathbf{x} \in \mathbb{Z}_q^N$, store $(id, \mathbf{x})$
- **Input P_2:** Upon $(Input, P_2, id, \mathbf{y})$ from P_2 and $(Input, P_2, id)$ from P_1 with a fresh identifier id and $\mathbf{y} \in \mathbb{Z}_q^N$, store $(id, \mathbf{y})$
- **Compute FDL:** Upon $(Compute, id_{res}, id_1, id_2)$ from both parties, sample $Bd+1$ uniform random coins $\mathbf{r}$, retrieve $(id_1, \mathbf{x})$ and $(id_2, \mathbf{y})$, compute $z \leftarrow h(\mathbf{x}||\mathbf{y}) + \texttt{FDL}_{\lambda,B,d}(\mathbf{r}) \mod q$ and store (id_{res}, z).
- **Output:** Upon $(Output, id)$ from both parties, where id is in the memory, retrieve (id, z) and output it to the adversary. Wait for an input from the adversary. If this is $Deliver$, then output z to both parties, else output $\perp$.

Fig. 3: The ideal functionality of the FDL mechanism.

- *For each active $\mathcal{S}$, there exists an ensemble of ε-DP mechanisms $\tilde{h}_{\mathcal{S}} : \mathcal{D} \to R$ such that for all $D \in \mathbb{Z}_q^{2N}$, the probability distributions $\tilde{h}_{\mathcal{S}}(D)$, $OUT^1_{\mathcal{F},\mathcal{S}}(D)$ and $OUT^2_{\mathcal{F},\mathcal{S}}(D)$ are statistically indistinguishable.*

5 A protocol for the FDL mechanism

As stated before, 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 C. 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 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$_{2^k}$ [22] for the arithmetic computations, the FKOS protocol [36] for binary circuits and *daBits* (doubly-authenticated bits) [57] 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. D 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 [35]. Our protocol is presented in Figure 4.

Protocol $\pi_{\mathcal{M}_{\text{FDL}}}$

Parameters: Natural numbers B, d, q, N, bit decomposition $\hat{\alpha}^1, ..., \hat{\alpha}^d$ and an efficiently computable function $h : \mathbb{Z}_q^{2N} \to \mathbb{Z}_q$. Assume access to $\mathcal{F}_{ABB}$ and an ideal functionality $\mathcal{F}_h$ for evaluating h.

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 $r_i \leftarrow e_j^1 \oplus e_j^2$ 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 $\mathcal{F}_h$, which then sends $\bar{h} \leftarrow h(\mathbf{x}||\mathbf{y})$ to $\mathcal{F}_{ABB}$ to be stored as an element of the arithmetic field.
2. The sum of $\bar{h}$ and the FDL sample is computed via $\mathcal{F}_{ABB}$ and the result is output to the players.

Fig. 4: The protocol description for the FDL mechanism in the $(\mathcal{F}_{ABB}, \mathcal{F}_h)$-hybrid world.

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

Theorem 1. *Assume access to the ideal functionalities $\mathcal{F}_{h_\kappa}, \mathcal{F}_{ABB}$. Let $q > 2h_\kappa^{max} + 2B_\kappa$, $B_\kappa \in \{1, \ldots, \lceil q/2 \rceil - 1\}$, $\lambda_\kappa = \frac{\varepsilon_\kappa}{\Delta h_\kappa}$ and let $e^{-B_\kappa/\lambda_\kappa}$ and $B_\kappa 2^{-d_\kappa}$ be negligible in κ. Let $\{h_\kappa : \mathbb{Z}_q^{2N} \to \mathbb{Z}_q\}_{\kappa\in\mathbb{N}}$ be an ensemble of deterministic functions with $h_\kappa^{max} := \max_{D\in\mathbb{Z}_q^{2N}} |h_\kappa(D)| \forall \kappa$. Let $\{\hat{h}_\kappa(D)\}_{\kappa\in\mathbb{N}}$ be $\{\mathcal{M}_{RTGeo}^{q,h_\kappa,\lambda_\kappa}(D)\}_{\kappa\in\mathbb{N}}$. Then $\pi_{\mathcal{M}_{\texttt{FDL}}}(B_\kappa, d_\kappa, q, N, decomp(\lambda_\kappa, B_\kappa), h_\kappa)$, with ideal functionality $\mathcal{F}_{\mathcal{M}_{\texttt{FDL}}}$, is a $\left(\nu, \frac{2e^{-1/\lambda_\kappa}}{e^{-1/\lambda_\kappa}+1} e^{-\nu/\lambda_\kappa}\right)$-additive-useful ε_κ-SIM*-CDP protocol for $\{h_\kappa\}$, for all positive integers ν.*

Proof. The usefulness follows from the parameter choices and Lemma 3. That $\mathcal{F}_{\mathcal{M}_{\texttt{FDL}}}$ has DP preservation under passive and active corruptions follows from the parameter choices and Lemma 4.

Finally, the UC-realisation in the $(\mathcal{F}_{h_\kappa}, \mathcal{F}_{ABB})$-hybrid world follows directly from noting that $\pi_{\mathcal{M}_{\texttt{FDL}}}$ consists solely of calls to these ideal functionalities, i.e. there are no other messages sent between the parties. Therefore, the security is inherited directly from the fact that the ABB (as well as $\mathcal{F}_{h_\kappa}$) returns either the correct answer or aborts and returns $\perp$ and this is captured in $\mathcal{F}_{\mathcal{M}_{\texttt{FDL}}}$ by allowing the adversary to abort the execution (after having learned the output).

□

Asymptotic computational cost. We consider the computational cost of $\pi_{\mathcal{M}_{\texttt{FDL}}}$ in terms of calls to the ABB (and ignore the cost of realizing the $\mathcal{F}_h$ functionality). 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 [32], the asymptotic computational cost of the `FDL` function (also in terms of calls to the ABB, or rather, the number of multiplications) is $O(Bd)$. This complexity follows directly from Definition 8 since all steps of the `FDL` procedure are repeated B times (e.g., B Bernoulli trials are sampled, 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. 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.

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

Parameter choices. From the properties above, the following parameter considerations follow: The security parameter $\kappa = \lceil \log_2(q) \rceil$. Both ε_κ and Δh are independent of κ. Further, we can set the `FDL` specific parameters as $B = d = \kappa$. Finally, we have $q > 2h^{max} + 2B = 2N \cdot \max(a^2, b^2) + 2B$.

In practice, one strategy is to choose κ as a canonical value for statistical security in cryptography, e.g., $\kappa = 40$, and then let this also be B and d. The choice of ε is highly challenging, and there is a lively discussion in the literature on it, although consensus is largely lacking [28, 50, 55, 49]. Luckily, there is no direct dependence on the choice of ε 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 ε 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) [47] library. Among other things, they provide efficient implementations of the SPDZ_{2^k} [22] and the FKOS [36] MPC schemes, and da-bit [57] and edabit [35] 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 of the wrong domain. Given the protocol in EIKN [32], 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 summarize our approach in D.

6.1 Benchmarks

In this section, we present benchmarks of our FDL mechanism with $B = d = \kappa$ and measure performance for different settings [13]. Relevant for parameter $\hat{\alpha}$, the bit decomposition of the Bernoulli bias, is the decomposition length d. When setting a value α, 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 the pre-processing and online phases of MPC, where the pre-processing step consists of generating necessary multiplication triples and da-bits.

For 1, all computations are performed in a LAN setup with $<$ 1ms round-trip time (RTT). The benchmarks show that our mechanism achieved competitive

[13] Upon acceptance of the paper, the code will be open-sourced.

results for all statistical security parameters we tested. The runtime results reflect the expected quadratic growth given the asymptotic complexity $O(Bd)$. The network results show a similar relation. Compared to concurrent work [46], our mechanism outperforms their result in runtime and memory overall.[14] Regarding network data, the authors did not provide distinct benchmarks for the pre-processing and online phases. 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. If the sensitivity of a function is known before the data is processed, the parties can already engage in a noise sampling procedure. Comparing with [32] is challenging as they only provide asymptotic complexities. However, our mixed-circuit approach represents a substantial performance improvement to the `FDL` mechanism.

Protocol	κ	Runtime[ms]		Network[MB]	
		Prep	Online	Prep	Online
Ours	40	72	39	4.91	5.97
	80	98	112	6.96	19.42
	128	129	269	9.72	47.81
[46]	-	991	1.4	492.72	

Table 1: Runtime and network cost of a single `FDL` execution with different security levels

In 2, we present benchmarks for different network settings. In Setting 1, we simulate a less powerful LAN setup by limiting the network to 1Gbit/s and the RTT to 1ms. In Setting 2, on the other hand, we simulate a WAN network with 100Mbit/s and 100ms RTT, reflecting a solid but distant connection (e.g., intercontinental). Again, the results show the relation to the parameters B and d. 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. Comparing our results to Keller et al. [46], we achieve better results for the slow LAN network but with an increased weakness in the online phase. For the WAN setting, we see the high round complexity of our implementation. With $\kappa = 40$, our implementation still achieves competitive performance. However, increasing κ further adversely impacts the performance of the online phase.

[14] One should however note that [46] use a different notion of DP and also is in the setting of passive adversaries, thus making exact comparions challenging.

Protocol	κ	Setting 1 [ms]		Setting 2 [ms]	
		Prep	Online	Prep	Online
Ours	40	216	273	10 726	20 431
	80	275	788	15 287	51 475
	128	488	1 442	20 807	104 692
[46]	-	4 707	4.81	42 352	47.99

Table 2: Runtime in Setting 1 with 1Gbit/s and 1ms RTT and Setting 2 with 100 Mbit/s and 100ms RTT

7 Outlooks

In this work we introduce a new definition of computational DP in protocols and argue that it is preferable to previous definitions in certain scenarios. 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. Additionally, we consider our definition intuitive, and to argue this we present high-level arguments for why it is similarly intuitive as previous definitions. 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 which 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 practical outlooks, in this work, we applied our definition in practice and achieved competitive performance. However, there are vastly different setup assumptions in the modern data processing landscape. In particular, extending our definitional work into the setting of an arbitrary number of parties remains an interesting open direction. In this regard, the structure of our implementation is flexible and adaptable.

Acknowledgements. We gratefully thank the authors of MPRV [56] for providing a full version of their paper and for answering our questions. In particular, we thank Salil Vadhan for his helpful comments regarding details of the SIM$^+$-CDP definition. We thank Lea Demelius and Peter Waldert for fruitful discussions on earlier drafts of this paper.

Fredrik Meisingseth is 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. Differential privacy, vol. 2006. ICALP (2006). https://doi.org/10.1007/11787006_1
2. 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
3. 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
4. Balcer, V.: Exploring Models for Implementing Differential Privacy. Ph.D. thesis (2022), https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37373743
5. Balcer, V., Vadhan, S.: Differential privacy on finite computers (2017), https://arxiv.org/abs/1709.05396
6. 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
7. 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)
8. 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
9. Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., Seefeld, B.: Prochlo. In: Proceedings of the 26th Symposium on Operating Systems Principles. ACM (oct 2017). https://doi.org/10.1145/3132747.3132769
10. Böhler, J., Kerschbaum, F.: Secure multi-party computation of differentially private median. In: Proceedings of the 29th USENIX Conference on Security Symposium. SEC'20, USENIX Association, USA (2020)
11. 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

12. 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
13. Canetti, R.: Security and composition of multiparty cryptographic protocols. J. Cryptol. **13**(1), 143–202 (jan 2000). https://doi.org/10.1007/s001459910006
14. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Paper 2000/067 (2000), https://eprint.iacr.org/2000/067, https://eprint.iacr.org/2000/067
15. Canetti, R.: Universally composable security. J. ACM **67**(5) (sep 2020). https://doi.org/10.1145/3402457, https://doi.org/10.1145/3402457
16. Canonne, C.L., Kamath, G., Steinke, T.: The discrete gaussian for differential privacy (2020), https://arxiv.org/abs/2004.00010
17. 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
18. 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)
19. 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)
20. 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
21. 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
22. Cramer, R., Damgård, I., Escudero, D., Scholl, P., Xing, C.: Spdz2k: 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
23. Cramer, R., Damgård, I.B., Nielsen, J.B.: Secure Multiparty Computation and Secret Sharing. Cambridge University Press (2015). https://doi.org/10.1017/CBO9781107337756
24. 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)
25. 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)

26. 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
27. 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
28. 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
29. 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
30. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
31. Eigner, F., Kate, A., Maffei, M., Pampaloni, F., Pryvalov, I.: Differentially private data aggregation with optimal utility. p. 316–325. ACSAC '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2664243.2664263, https://doi.org/10.1145/2664243.2664263
32. 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
33. 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)
34. Escudero, D.: An introduction to secret-sharing-based secure multiparty computation. Cryptology ePrint Archive, Paper 2022/062 (2022), https://eprint.iacr.org/2022/062, https://eprint.iacr.org/2022/062
35. 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)
36. 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
37. Ghazi, B., Ilango, R., Kamath, P., Kumar, R., Manurangsi, P.: Separating computational and statistical differential privacy (under plausible assumptions) (2022)
38. Ghosh, A., Roughgarden, T., Sundararajan, M.: Universally utility-maximizing privacy mechanisms (2008), https://arxiv.org/abs/0811.2841

39. Goldreich, O.: Secure multi-party computation. Manuscript. Preliminary Version (1998)
40. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, USA (2004)
41. 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
42. 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)
43. 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
44. 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
45. 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)
46. 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, https://eprint.iacr.org/2023/1594
47. 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
48. 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
49. 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)
50. Krehbiel, S.: Choosing epsilon for privacy as a service. Proceedings on Privacy Enhancing Technologies **2019**, 192 – 205 (2019)
51. Lindell, Y.: Highly-efficient universally-composable commitments based on the ddh assumption. In: Paterson, K.G. (ed.) Advances in Cryptology – EUROCRYPT 2011. pp. 446–466. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
52. Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: Kurosawa, K., Hanaoka, G. (eds.) Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7778, pp. 107–124. Springer (2013). https://doi.org/10.1007/978-3-642-36362-7_8

53. 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)
54. 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
55. 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)
56. 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)
57. 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
58. 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
59. 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

A Remarks on details in the SIM*-CDP definition

A.1 The need for treating DP preservation under active corruptions separately

The formulation of the property of DP preservation under active corruptions is motivated by the idea that the UC-security framework allows for arbitrary adversarial influence in the ideal world, since this world is *secure by definition*. This does however not mean that the phrasing of the ideal functionality corresponds well to our intuitions about differential privacy. For instance, one can think of the example where the adversary is allowed to cause the ideal functionality to output the entire database in the clear, but that an honest execution outputs a constant. Such an ideal functionality is obviously DP in the sense that under passive corruptions, the outputs to the parties are identical to a DP mechanism and it can be secure in the sense that there is perfect indistinguishability between the ideal and real world (for both active and passive corruptions). It is however unappealing to say that such a protocol is computationally DP against active adversaries. On the other hand, one cannot require that the outputs remain very similar to those of the DP mechanisms $\hat{h}_\kappa$ when there are active corruptions. This

is realised by the fact that an adversary can always cause the protocol execution to fail, which the DP mechanism cannot capture since the adversary can simply choose to abort the protocol with different probability depending on its strategy. In the end, the outputs to the parties can be required to be (statistically close to) DP (with unchanged parameters) for each of the ideal-world adversaries but not via the same mechanism. Similarly, the usefulness of these mechanisms for approximating h cannot be guaranteed, since robustness is impossible in general for two-party computation.

A.2 Properties of the new definition

Domains and ranges. The inputs to the protocol π and the ideal functionality $\mathcal{F}$ lie in $\{0,1\}^*$ and are of polynomial size in κ, since $\pi, \mathcal{F}$ are PPT (See Section 3.2 in [15]). Thus, the definition of statistical indistinguishability between ensembles of probability distributions implies that the domain D is also that of polynomially large elements of $\{0,1\}^*$. Similarly, the range of π and $\mathcal{F}$ is also the elements of $\{0,1\}^*$ of polynomial length but this need not imply that the same holds for $\{h_\kappa\}, \{\hat{h}_\kappa\}, \{\tilde{h}_\kappa\}$, since the statistical indistinguishability would still be achievable in the case that $\hat{h}_\kappa$ returns a string in $\{0,1\}^*$ of superpolynomial length with negligible probability. We also note that we require h_κ and $\hat{h}_\kappa$ to be defined for the same range, although not necessarily have the same *support*. Another remark is that we throughout abuse notation and describe all inputs and outputs as elements in $\mathbb{Z}_q$ rather than $\{0,1\}^*$, but that the translation in representation is direct. In particular, we will have D be the $2N$-fold cartesian product of $\mathbb{Z}_q$, and let $R = \mathbb{Z}_q$. Since each element of $\mathbb{Z}_q$, with $\lceil \log_2(q) \rceil = \kappa$, can be represented by κ bits this means that each element in D is of polynomial size in κ.

Relation to previous definitions. The definition of SIM*-CDP is not simply a restriction or a relaxation of SIM$^+$-CDP. Perhaps, SIM*-CDP is to be primarily seen as a less general version of SIM$^+$-CDP, mostly due to that the model of computation is fixed and the notion of usefulness is restricted. Further, one could argue that the definition has been relaxed, since correctness is now computational rather than perfect and the DP mechanism need no longer be expressed as an algorithm or protocol. On the other hand, our definition is stricter than the previous one in that the simulator must be PPT. Due to this seeming lack of direct comparability, we leave deriving more explicit relationships between the two definitions for future work.

We can note, however, that the argument given in the full version of MPRV [56] for that SIM$^+$-CDP implies the weaker notion of SIM-CDP also applies for SIM*-CDP in the case that one uses ITMs as the computational model in the SIM-CDP definition. In short, this is due to that SIM-CDP does not include the demand of perfect correctness and therefore the existence of a simulator for the

SIM*-CDP definition implies the existence of a simulator in the sense needed for SIM-CDP.

B Proofs

B.1 Proof of Lemma 1

Proof. Let $Z \sim \mathcal{M}^{p,h,\lambda}_{RTGeo}(D)$ and $Y \sim \mathcal{M}^{2B,h,\lambda}_{SRTGeo}(D)$ for arbitrary λ, D. 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,

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

where $\bar{h}$ is shorthand for $h(D)$. The first inequality follows from the argument that this choice of z is where the maximum difference between the probability masses occur. The equalities follow by inserting the formulas from Definition 5 and direct simplifications.

□

B.2 Proof of Lemma 2

Proof. Firstly, $\texttt{Ber}_{\hat{\alpha}}$ exactly samples a Bernoulli trial with parameter equal to the recomposition of the first d elements of α. Call this parameter value α'. This means that the statistical distance between $\texttt{Ber}(\hat{\alpha})$ and an exact Bernoulli trial with parameter $\hat{\alpha}$ is the same as between two exact Bernoulli trials with parameter $\hat{\alpha}$ and α', 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 $\mathcal{M}^{\lambda,B,d,h}_{\texttt{FDL}}(D)$ 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 $B2^{-d}$

□

B.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:

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

□

B.4 Proof of lemma 4

Proof. We consider the two cases separately. Let $D \in \mathbb{Z}_q^{2N}$ be arbitrary, and we at times suppress it in notation.

Passive $\mathcal{S}$:
For a passive adversaries, $OUT^i_{\mathcal{F},\mathcal{S}}(D)$ is always equal to $\mathcal{M}_{\texttt{FDL}_{\lambda,B,d}}(D)$, since the passive adversary never chooses to abort. Therefore, letting $\hat{h}(D)$ be $\mathcal{M}^{q,h,\varepsilon/\Delta h}_{RTGeo}$, the statistical indistinguishability follows from lemmata 1 and 2 together with that $e^{-B/\lambda}$ and $B2^{-d}$ are negligible. This is due to the statistical distance satisfying the triangle inequality.

Active $\mathcal{S}$:
For active adversaries, the only change is that they can choose to deny the parties output, instead having $\mathcal{F}$ output $\perp$ to them. This choice, the adversary can make as a function of $\mathcal{M}_{\texttt{FDL}_{\lambda,B,d}}(D)$. Let us denote the probability of $\mathcal{S}$ choosing to abort when given output z $p_{\mathcal{S}} : \mathbb{Z}_q \to [0,1]$. Then we have

$$OUT^i_{\mathcal{F},\mathcal{S}}(D) = \begin{cases} \perp, \textit{ with probability } p_{\mathcal{S}}(\mathcal{M}_{\texttt{FDL}_{\lambda,B,d}}(D)) \\ \mathcal{M}_{\texttt{FDL}_{\lambda,B,d}}(D), \textit{ otherwise.} \end{cases} \tag{4}$$

Consider the mechanism $\tilde{h}_{\mathcal{S}}$ defined as

$$\tilde{h}_{\mathcal{S}} = \begin{cases} \perp, \textit{ with probability } p_{\mathcal{S}}(\hat{h}(D)) \\ \hat{h}(D), \textit{ otherwise.} \end{cases} \tag{5}$$

The double statistical distance between $\tilde{h}_{\mathcal{S}}(D)$ and $OUT^i_{\mathcal{F},\mathcal{S}}(D)$ is

$$
\begin{aligned}
&\sum_{z\in\mathbb{Z}_q\cup\{\perp\}} |\mathbb{P}(\tilde{h}_{\mathcal{S}}=z)-\mathbb{P}(OUT^i_{\mathcal{F},\mathcal{S}}(D)=z)| \\
&=\sum_{z\in\mathbb{Z}_q} |\mathbb{P}(\hat{h}_{\mathcal{S}}=z)(1-p_{\mathcal{S}}(z))-\mathbb{P}(\mathcal{M}=z)(1-p_{\mathcal{S}}(z))| \\
&+|\mathbb{P}(\tilde{h}_{\mathcal{S}}=\perp)-\mathbb{P}(OUT^i_{\mathcal{F},\mathcal{S}}(D)=\perp)| \\
&=\sum_{z\in\mathbb{Z}_q}(1-p_{\mathcal{S}}(z))|\mathbb{P}(\hat{h}_{\mathcal{S}}=z)-\mathbb{P}(\mathcal{M}=z)| \\
&+\left|\sum_{z\in\mathbb{Z}_q}(\mathbb{P}(\hat{h}_{\mathcal{S}}=z)p_{\mathcal{S}}(z))-\sum_{z\in\mathbb{Z}_q}(\mathbb{P}(\mathcal{M}=z)p_{\mathcal{S}}(z))\right| \\
&\leq \sum_{z\in\mathbb{Z}_q}|\mathbb{P}(\hat{h}_{\mathcal{S}}=z)-\mathbb{P}(\mathcal{M}=z)| \\
&+\left|\sum_{z\in\mathbb{Z}_q}p_{\mathcal{S}}(z)(\mathbb{P}(\hat{h}_{\mathcal{S}}=z)-\mathbb{P}(\mathcal{M}=z))\right| \\
&\leq e^{-B/\lambda}+B2^{-d} \\
&+\sum_{z\in\mathbb{Z}_q}|\mathbb{P}(\hat{h}_{\mathcal{S}}=z)-\mathbb{P}(\mathcal{M}=z)| \\
&\leq 2(e^{-B/\lambda}+B2^{-d})\leq \mathsf{negl}(\kappa).
\end{aligned}
$$

The first two equalities follow directly from separation of terms in the sum and from inserting the dependence on $p_{\mathcal{S}}$. The first inequality stems from the fact that probabilities are at most one. The second inequality follows also from this fact together with lemmata 1 and 2 and the triangle inequality. The second to last inequalty is again due to the lemmata and the final inequality follows from the assumptions on negligible statistical distance.

□

C 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 [6]. 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[15] $[\![z]\!] = \alpha\beta + \beta[\![x]\!] + \alpha[\![y]\!] + [\![c]\!] = [\![x \cdot y]\!]$.

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 [25], 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 the ASS over the ring $\mathbb{F}_{2^k}$ [22] or replacing the expensive homomorphic encryption with oblivious transfer [48]. Note that both improvements, to some degree, accept a higher communication for a lower computation complexity.

D On input validation for input-dependent sensitivity

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. The need for input validation in this sense is one reason that previous works that work in the model of passive adversaries, such as [32, 46], cannot be directly translated to give security against active adversaries. In the presence of passive adversaries, there is simply 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.

In order 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[16] within the MPC domain, and secondly, one could accept the inputs bit-by-bit and re-compose those bits

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

[16] For instance, such as described in the Bulletproofs paper [12].

into elements of the larger field. While implementing the inner-product protocol, we opt for the second approach, as we notice (in Section 6) this allows for a highly efficient protocol.

Input validation. First, we looked at so-called Bulletproofs, proposed by Bünz et al. [12]. Translated to MPC, given an input $x \stackrel{?}{\in} [a, b]$, we can calculate a mask $\mu_i = \sum_{i=a}^{b}(x_i - i)x$. This mask is 0 if x is within the range and non-zero otherwise. Such an approach has two main issues: the number of multiplications scales linearly with the input range, and we have to obscure the mask before opening it so as not to leak information to adversarial parties. Next to the Bulletproof-based approach, we translated an idea from Ling et al. [52] to the MPC domain. Let the range $r = b - a$ and $l = \lfloor \log_2(r) \rfloor + 1$, we define the composition basis $\omega \in \mathbb{Z}_q^l$, where ω_i is the i-th element of ω. We set

$$\omega_0 = \left\lceil \frac{r}{2} \right\rceil \text{ and } \omega_i = \left\lceil \frac{r - \sum_{j=0}^{i-1} \omega_j}{2} \right\rceil.$$

Two properties follow: First, given this base, any value $v \in \{0, \ldots, r\}$ can be represented as a binary vector $\mathbf{b}$ of size l. Second, the reconstruction $v = \sum b_i \omega_i$ cannot lead to a value greater than r. In the end, this approach reduces the complexity of checking bound r of value v to checking $b_i \stackrel{?}{\in} [0, 1]$ of the values $\{b_0, \ldots, b_{l-1}\}$. We can reduce the multiplication complexity to $O(\log n)$ for the final check.