SoK: Computational and Distributed Differential Privacy for MPC

Fredrik Meisingseth
Graz University of Technology
Graz, Austria
fredrik.meisingseth@iaik.tugraz.at

Christian Rechberger
Graz University of Technology
Graz, Austria
christian.rechberger@iaik.tugraz.at

Abstract

In the last fifteen years, there has been a steady stream of works combining differential privacy with various other cryptographic disciplines, particularly that of multi-party computation, yielding both practical and theoretical unification. As a part of that unification, due to the rich definitional nature of both fields, there have been many proposed definitions of differential privacy adapted to the given use cases and cryptographic tools at hand, resulting in computational and/or distributed versions of differential privacy. In this work, we offer a systemization of such definitions, with a focus on definitions that are both computational and tailored for a multi-party setting. We order the definitions according to the distribution model and computational perspective and propose a viewpoint on when given definitions should be seen as instantiations of the same generalised notion. The ordering highlights a clear, and sometimes strict, hierarchy between the definitions, where utility (accuracy) can be traded for stronger privacy guarantees or lesser trust assumptions. Further, we survey theoretical results relating the definitions to each other and extend some such results. We also discuss the state of well-known open questions and suggest new open problems to study. Finally, we consider aspects of the practical use of the different notions, hopefully giving guidance also to future applied work.

Keywords

Differential Privacy, Multi-party Computation, Systematization of Knowledge

1 Introduction

The applications of Differential Privacy (DP) and Multi-party Computation (MPC) have essentially orthogonal goals, namely that with MPC one wishes to make sure that when performing a joint computation, no information is learned by the adversary *except for that which can be learned from the allowed computation output*, whereas DP concerns bounding the *privacy loss incurred from said output* [4, 22]. In the words of Beimel, Nissim and Omri [4], MPC tells us *how* to compute something privately and DP tells us *what* can be privately computed. Therefore, combining them is an appealing prospect as it can potentially enable protocols that provide privacy protection with respect to both their execution and their outputs. In addition to the case where one is a priori interested in achieving the privacy goals of both MPC and DP, the topic of combining DP with MPC (a part of what is at times called *DP-cryptography* [69]) also concerns improving a scheme in one of the disciplines by utilising tools and perspectives from the other. This has proven a fruitful endeavor and this success is rooted in that not only the goals but also the typical challenges within the two fields are mostly distinct. The literature on using MPC techniques to improve DP systems mostly focuses on removing the need to fully trust a single central computational party. Avoiding this trust assumption can be done without relying on MPC techniques, for instance by using the local or shuffle models of DP (see Section 3), but these models sometimes do not admit accuracy similar to (or even close to) that in the central model. If one does use MPC, however, then the clients can distributedly simulate the central dataholder, thereby avoiding the main trust assumption in DP without lowering the accuracy of the protocol [4, 5, 22]. Using DP to improve MPC schemes is typically done to improve the efficiency of the scheme. The efficiency problems in MPC can intuitively be seen as caused by the need to spread out the secret information such that, at all times, all sufficiently small coalitions of parties cannot learn *any* information about the underlying secret. Oftentimes, such efficiency problems can be reduced if one relaxes the demand that no information should leak, to that the information that is leaked is from a differentially private function of the secret inputs [5, 40, 46, 71]. When unifying the formalities in MPC and DP, some hurdles arise however. Firstly, DP is typically studied in a *statistical* (information-theoretic) setting whereas multi-party protocols must for certain settings rather work with *computational* guarantees. Therefore, when deploying DP in those settings, the usual definition of *statistical DP (SDP)* must be turned into *computational DP (CDP)*.[1] Secondly, DP is formulated with respect to single probabilistic algorithms, called *mechanisms*, rather than with respect to algorithms interacting with one another within a protocol.

There are two main motivations behind writing a systematization of knowledge paper on this particular topic at this particular time, one regarding theory and one regarding the application of the theory in practice. On the theoretical side, there have now been so many distributed CDP definitions proposed that getting an overview of them by either following the early definitional works and their follow-ups or by following related work sections of recent work is getting increasingly laborious. This together with recent rapid developments [32, 37, 41, 42] on fundamental open problems, such as those posed in [67], leads to a need to survey the topic for the benefit of new theoretical work. On the practical side, there is a growing literature [5, 6, 27, 70] on concrete protocols combining DP and MPC and this causes a need to discuss the theory with

Proceedings on Privacy Enhancing Technologies 2025(1), 420–439

https://doi.org/10.56553/popets-2025-0023

[1]We use 'DP' to refer to both computational and statistical definitions. When referring specifically to definitions that are either statistical or computational, we call them SDP and CDP definitions, respectively.

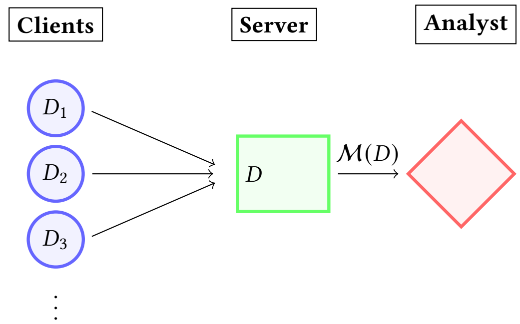

Figure 1: Overview of the central model of DP. The single trusted server (dataholder) gets the whole dataset D in the clear, then computes the DP mechanism $\mathcal{M}$ and sends the result to the untrusted analyst.

respect to its relationship to practice. This applies particularly to concerns about what settings the various CDP definitions are suited for and the choice of parameter regime. As an illustrative example, take the recent paper [5] from CCS'22. There an efficient protocol for computing DP histograms in a two-party model is proposed and it is shown that the protocol is CDP by formulating an ideal functionality that is SDP and then proving that the protocol realises it, which is a common and natural design strategy. The illustrative part of this example is that, since the ideal/real paradigm is used, SIM$^+$-CDP is chosen as the CDP notion as it gives the strongest guarantees and also is geared towards this specific proof strategy. However, the SIM$^+$-CDP definition needs to be adapted (which is done implicitly) since the original definition uses $(\varepsilon, 0)$-SDP in the functionality and requires perfect correctness but the protocol, for practical reasons, can only give statistical correctness and realises an ideal function which is (ε, δ)-SDP with non-zero δ. This implicit adaptation of the CDP definition shows that there is a need to explicitly discuss the details of the CDP definition from the point of view of application, to make it clear for practitioners when specific flavours of CDP can be used and when not.

1.1 Characterising DP Definitions

Traditionally, DP is studied in the central model (see Figure 1), where the data of all *clients* is held in the clear by a central trusted *server* (or *dataholder*). On this dataset the *mechanism* is run and the output is given to an untrusted *analyst*. DP is a property of the mechanism, as is seen in the following definition (details follow in Section 2). We formulate the definition with the adjacency notion being variable. For a very general definition of what an adjacency notion is, see Definition 2.2.

Definition 1.1 ((ε, δ)-SDP [21, 23]). Let $\varepsilon \geq 0, \delta \in [0, 1]$ and ADJ be an adjacency notion on the input domain $\mathcal{D}$. A probabilistic algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is (ε, δ)-differentially private (SDP) if for all pairs (D, D') of adjacent databases (with respect to ADJ) in $\mathcal{D}$ and all measurable subsets S of $\mathcal{R}$,

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

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

On a very high level, DP defines a property of a probabilistic algorithm that relates a notion of closeness between input pairs (the adjacency notion) to some requirement of closeness[2] between the respective output distributions. When placing the mechanism within the context of an interaction between parties in a protocol, a DP definition additionally specifies what parts of the data involved in the interaction are to be seen as input to the mechanism and what parts are to be seen as the output. Further, in the case that some party that gets mechanism output is computationally bounded, one can relax the requirement that the process creating those outputs actually is SDP and rather say it has to 'look SDP' to the computationally bounded party. In summary, we consider a DP definition to be determined by:

- The *distribution model* – what probabilistic function in the interaction should be DP?
- The *notion of adjacency* – how is the condition that two inputs are close to each other formalised?
- The *notion of output closeness* – how is the requirement that two output distributions are similar formalised?
- The *computational perspective* – are there *computational limitations* on the party receiving the output? If yes, what does it formally mean for a mechanism to *'look SDP'* to a computationally bounded party?

Typically, some of these characteristics are fixed in a definition and some are kept variable, meaning that the same definition can be instantiated with different choices of them. As an example, we can consider the definition of SDP above, in which the distribution model, computational limitations and output closeness notion are implicitly kept fixed (changing them results in a new definition). In particular, the distribution model is fixed to the central model, all parties are allowed to be computationally unbounded and output closeness is defined by Equation (1). The adjacency notion on the other hand is kept variable and changing it can be done without it being seen as proposing a new DP definition.

Scope. Throughout this survey, we follow this practice of letting the adjacency notion be variable within a DP definition and we fix the output closeness to the standard notion as in (ε, δ)-SDP. We let the other two properties define new definitions. It is not always clear when a change in those properties warrants the resulting definition to be seen as new, for instance in the case when changing a computationally bounded party from being uniform to being non-uniform, or when the distribution model shifts from involving three parties to involving four parties. We will see two definitions that we consider to only differ in such details (i.e. in details that are perhaps technically crucial but do not change the intuitive appeal or raison d'être of the definition) as being two instantiations of the same definition, even if they were not originally proposed as such.[3] Since there are essentially infinitely many possible and a very large number of potentially relevant choices of distribution model and computational perspective, we need also restrict our discussion

[2]Technically, the notion of output closeness is typically a *divergence*, i.e. some non-negative function of two probability distributions which is 0 if and only if the distributions are identical.
[3]The choice of when to consider two definitions to be the same in this sense is a quite informal one. Of course it would be preferable to avoid turning to such informalities although we have not found a way to do so without gravely restricting the overview of the space of definitions by restricting all comparisons to very specific settings.

to certain choices of them. Firstly, we include only choices within each dimension that have been treated in the literature on either distributed or computational DP (ignoring choices only present in the non-DP literature on secure computation or central-model SDP). Secondly, as our motivating use case is the combination of DP and MPC, our focus is on DP definitions that are *both* distributed *and* computational. Therefore, we include only distributed SDP notions and central-model CDP notions to the extent needed to provide understanding and context for the distributed CDP definitions. For a brief list of definitions we have chosen not to include, see Appendix B. Regarding the adjacency notion, we leave it arbitrary in all DP definitions although almost all results we survey use an adjacency notion where datapoints are either binary strings or integers of bounded size and two datasets are adjacent if at most one element is changed.[4] It is also worth noting that we formulate all CDP definitions considering protocols with respect to an arbitrary number of parties, in order to increase the generality of our discussion, even though they have primarily been presented, used and analysed in the two-party setting.

Relations between definitions. After surveying the field of definitions and deciding which of them to consider, we turn to relating the definitions to each other, with the goal of ordering them with respect to the guarantees they provide and the types of computational tasks that can be solved whilst satisfying them. The literature contains primarily two types of such results: results showing *direct implications (DI)*, i.e. that all mechanisms satisfying one definition fulfill another as well, and results showing that one definition is *more expressive (ME)* than another, i.e. that all tasks solvable with the first definition can also be solved with the other. These types of results are interesting both in their positive form (then we call them *implications*) and in their negative form (then we call them *separations*). Roughly, DI-results concern establishing properties of *every* mechanism satisfying a given definition whereas ME-results concern whether there exists *any* mechanism solving a given task whilst satisfying the DP definition. Therefore positive DI-results are strictly stronger than positive ME-results and the other way around for separations. We define the different types of relations in Section 2.

1.2 Related Work

The two works most related to this are the 2020 survey by Desfontaines and Pejó [19] and the 2017 survey by Vadhan [67]. Two key works that we will refer to heavily (and discuss later in more depth), since they largely initiated the formal study of distributed and computational DP are [4, 62]. In the following, we will refer to [62] and [4] as *MPRV* and *BNO*, respectively, after their authors.

Relation to Desfontaines and Pejó's survey. In [19] hundreds of definitions of DP are surveyed and categorised according to seven dimensions, including (with other names) the adjacency notion, output closeness and computational perspective. Additionally *variations in the privacy loss* (letting different inputs enjoy different types of privacy guarantees), *background knowledge* (making assumptions on the amount/type of background knowledge the adversary has), *formalism changes* (using different formalities in measuring how much knowledge the adversary can gain) and *relativising knowledge gain* (relating the knowledge gain to structures or correlations within the data) is considered. We ignore these four other dimensions in our survey because combining variations within them with computational or distributed versions of DP has occurred only very rarely. Variations to the distribution model are considered out of scope in [19] and therefore the overlap in the definitions covered there and in this work is quite small. Further, since we have a more concentrated scope, this holds true also with regard to results and discussions about the few definitions that are included in both works.

Relation to Vadhan's survey. Vadhan's 2017 monograph [67] gives a broad introduction to the relationship between differential privacy and computational complexity theory, giving an overview of the literature, including distributed and computational DP, and formulating open problems. The sections on distributed DP and CDP focus on relating these areas to complexity theory and SDP, and therefore do not discuss the wide range of proposed definitions. In this work, we fill the gap by surveying the definitions of distributed CDP and results relating them to each other. As many such recent results are tied to open problems posed by Vadhan, we also provide a status update on those problems.

2 Notation and Preliminaries

For any natural number N, let $[N] := \{1, \dots, N\}$. For a probability distribution Dist, let $a \leftarrow \mathrm{Dist}$ denote sampling a from Dist. We refer to a function from the naturals to the non-negative reals as *negligible* if it approaches 0 faster than the inverse of any polynomial. We reserve the notation negl and *poly* for arbitrary negligible or polynomial functions, respectively.

2.1 Protocols, Algorithms and Corruption Models

We follow the convention set in MPRV and describe a protocol simply as a set of parties, $\{P_1, ..., P_N\}$, where each party is an interactive probabilistic function. This abstraction is sufficient for our discussion, except for in two crucial aspects, namely when it comes to computational *efficiency* and *defining secure computation*, where a more nuanced model of protocol execution is needed. In formalising efficiency we use non-uniform algorithms (e.g. Turing Machines), as is the standard within the literature on CDP.[5] We let *PPT* stand for *probabilistic polynomial time* and *PPTM* stand for *non-uniform PPT Turing Machine*. If a function is computable by a PPTM, then we call it *efficiently computable*. We call a distribution *efficiently samplable* if there exists a PPTM mapping 1^κ, with κ being the security parameter, to a sample of the distribution. For more discussion on the efficient samplability of distributions in the context of CDP, see [61]. Regarding secure computation, we consider both *active* (also called *malicious* or *byzantine*) and *passive* (also called *semi-honest*) corruptions but assume they are

[4] Intuitively, the use of a change-one binary adjacency notion can be seen as being the smallest possible choice since if one considers machines that work in binary then all other changes to a dataset can be translated to a sequence of changed bits. Therefore, under the group privacy property of SDP, results with respect to single binary changes should offer easy translation to most other common adjacency notions.

[5] Note that all DP definitions which consider efficient computation can be instantiated just as well with respect to uniform PPT without changing the spirit of the definition. Such changes may however potentially invalidate some of the results that we survey.

static (i.e. the set of corrupted parties is fixed before the protocol execution starts). We represent a corruption model $\mathsf{COR} = (a, \mathcal{C})$ where $a \in \{\text{active}, \text{passive}\}$ and $\mathcal{C} \subseteq \mathsf{Powerset}([N])$. The set $\mathcal{C}$ defines what subsets of parties may be corrupted at the same time. For a protocol $\pi = \{P_1, ..., P_N\}$ and a set $C \in \mathcal{C}$, let $\{P_C\}$ denote $\{P_i : i \in C\}$ (the corrupted parties in π) and let $\{P_{-C}\}$ denote $\{P_i : i \notin C\}$ (the honest parties in π). The information available to the coalition of corrupted parties is formalised in their joint *view*, as defined below. The reason for exchanging $\{P_C\}$ with $\{\tilde{P}_C\}$ is to model active corruptions, i.e. where the corrupted parties do not follow their instructions.

Definition 2.1 (VIEW, reformulation from [4]). Let $\pi = \{P_1, ..., P_N\}$ be a protocol and $\mathcal{A}$ be an adversary corrupting a set $C \subset [N]$ of parties. For inputs $D = (D_1, ..., D_N) \in \mathcal{D}$, the view in π of the corrupted parties $\{\tilde{P}_C\}$, denoted $\mathsf{VIEW}^{\mathcal{A}}_{\pi,C}(D)$ is defined as the random variable containing the inputs of the parties in C, their random coins and the messages that they receive during the execution of the protocol $\{\tilde{P}_C\} \cup \{P_{-C}\}$ on inputs D. The randomness in $\mathsf{VIEW}^{\mathcal{A}}_{\pi,C}(D)$ is over the random coins of the honest parties $\{P_{-C}\}$.

For defining secure computation of protocols we use the standard definitions in the ideal/real-paradigm, in both the standalone [11, 35, 57] and UC frameworks [12, 18]. A brief introduction to those frameworks is found in Appendix A.

2.2 Adjacency

For a protocol with N parties we consider an input dataset as an ordered set D in the domain $\mathcal{D} := \chi^N$, for a data universe χ. We define a notion of adjacency as a set of pairs of datasets in the following fashion.

Definition 2.2 (Adjacency notion). An adjacency notion ADJ on the dataset domain $\mathcal{D}$ is a set in $\mathcal{D} \times \mathcal{D}$ that is symmetric, i.e. if $(D, D') \in \mathsf{ADJ}$ then so is (D', D), and $\forall D \in \mathcal{D}, (D, D) \in \mathsf{ADJ}$. If $(D, D') \in \mathsf{ADJ}$ the we say that D and D' are adjacent with respect to ADJ.

The adjacency notion is typically clear from context and hence we will most of the time simply say that D and D' are adjacent, without further specification. Note that the definition above can be instantiated to practically all commonly used adjacency notions, for instance those when each each party holds an integer and two datasets are considered adjacent if at most one player changes or removes their value. For protocols, it is commonplace to consider DP with an adjacency notion that is agnostic to the inputs of the corrupted parties, as formalised below.

Definition 2.3 (C-adjacency [4]). Let $\{P_1, \ldots, P_N\}$ be a set of parties, each with their own input $D_i \in \chi$, and $C \subset [N]$ be a proper subset of the indices of the parties. Let $D = \{D_1, \ldots, D_N\} \in \chi^N$, $D_{-C} := \{D_i : i \notin C\}$ and analogously for $D' \in \chi^N$. We say that D and D' are C-adjacent with respect to an adjacency notion ADJ if D_{-C}, D'_{-C} are adjacent with respect to ADJ.

2.3 Relations between DP Definitions

Most of this work concerns the relations between various DP definitions, and such relations can be either implications (showing that the two definitions are similar in some sense) or separations (showing the contrary). We consider two such types of results – *direct implications*, showing that any protocol satisfying the first definition also satisfies the second one, and *more expressiveness*, showing that all computational tasks solvable whilst satisfying the one definition are also solvable whilst satisfying the other. In other words, a direct implication requires that any task solvable with the first type of DP can also be solved with the other type of DP *by using the same mechanism*, whereas expressiveness allows the task to be solved by a different mechanism for the other type of DP. Therefore a direct implication result should imply a result of more expressiveness, and below we see that for our formalisation of these notions, it does. Both of these types of results are conditional on parameter regimes and we follow the convention of [19] in letting η, β both be collections of parameter tuples. For instance can η be $\{(\varepsilon_\kappa, \delta_\kappa) \in \mathbb{R}^2_+ : \varepsilon_\kappa > 0, \delta_\kappa = \mathsf{negl}(\kappa)\}$. Letting Def_1 be a DP definition, when the protocol π (run with security parameter κ) satisfies Def_1 with the parameters in η for the same κ , then we say that π satisfies η-Def_1.

Definition 2.4 (Direct implication (DI)). Let Def_1 and Def_2 be two DP definitions and η, β be parameter regimes for them, respectively. We say that η-Def_1 directly implies (DI) β-Def_2, denoted $\eta\text{-}\mathsf{Def}_1 \underset{(DI)}{\Longrightarrow} \beta\text{-}\mathsf{Def}_2$, if all protocols π satisfying η-Def_1 also satisfy β-Def_2. If $\eta\text{-}\mathsf{Def}_1 \underset{(DI)}{\Longrightarrow} \beta\text{-}\mathsf{Def}_2$ and $\eta\text{-}\mathsf{Def}_1 \underset{(DI)}{\Longleftarrow} \beta\text{-}\mathsf{Def}_2$, we say that they are equivalent.

In order to discuss expressiveness, we must define what it means for a protocol to solve a task. A *task* is defined with respect to a *utility function*, which we quite generally choose to formalise as a binary deterministic function mapping a dataset and a mechanism output to 1 iff the output was a "good" solution. Since DP mechanisms are probabilistic, we measure the utility as the probability that a mechanism will output a good solution.

Definition 2.5 (Utility function [9, 32]). A utility function is an efficiently computable deterministic function $u : \mathcal{D} \times \mathcal{R} \to \{0, 1\}$. Let Dist be a probability distribution on the domain $\mathcal{D}$. A mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is α-useful for u with respect to Dist if:

$$\underset{y \leftarrow \mathcal{M}(D), D \leftarrow \mathsf{Dist}(\mathcal{D})}{\mathbb{P}}(u(D, y) = 1) \geq \alpha. \tag{2}$$

If the inequality holds for all distributions Dist then we omit it from notation. An important detail here is that we require the utility function to be efficiently computable. This restriction is needed to rule out pathological separations between DP definitions, as argued in [9], and therefore to save the meaningfulness of results about expressiveness. Further, the restriction strengthens the practical appeal of the notion of utility, since it means that a mechanism is only seen as significantly more useful (having higher utility) than another if it can be feasibly tested that such is the case.

Definition 2.6 (Task). A task is a tuple $\tau = (\alpha, u, \mathsf{Dist})$ as in Definition 2.5. For a mechanism $\mathcal{M}$, we say that $\mathcal{M}$ solves the task τ if $\mathcal{M}$ is α-useful for u with respect to Dist.

An example of a task is to compute the mean of a vector of natural numbers to within 10% additive error, and do so for each $D \in \mathcal{D}$ with probability at least 0.9. In that example, $u(D, y) = 1$ iff $y \in [0.9 \cdot \mathsf{mean}(D), 1.1 \cdot \mathsf{mean}(D)]$ and $\alpha = 0.9$. When α is constant and the utility function is defined as a bounded distance between y and a function evaluation $f(D)$, then we call the distance bound the *accuracy* of $\mathcal{M}$ for computing f.

Definition 2.7 (More expressive (ME)). Let Def_1 and Def_2 be two DP definitions and η, β be parameter regimes for them, respectively. We say that β-Def_2 is more expressive (ME) than η-Def_1, denoted η-$\text{Def}_1 \underset{(ME)}{\Longrightarrow} \beta$-$\text{Def}_2$, if all tasks solvable whilst satisfying η-Def_1 are also solvable whilst satisfying β-Def_2. That is, for all τ such that $\exists \pi$ that satisfies η-Def_1 and solves τ, there also exists a π' that satisfies β-Def_2 and solves τ. If η-$\text{Def}_1 \underset{(ME)}{\Longrightarrow} \beta$-$\text{Def}_2$ and η-$\text{Def}_1 \underset{(ME)}{\Longleftarrow} \beta$-$\text{Def}_2$ then we say that the definitions are equally expressive.

A direct consequence of the definitions of DI and ME is that DI-results imply ME-results, in the desired way.

Corollary 2.8. *For any $\eta, \beta, \text{Def}_1, \text{Def}_2$ as above, if η-$\text{Def}_1 \underset{(DI)}{\Longrightarrow} \beta$-$\text{Def}_2$ then η-$\text{Def}_1 \underset{(ME)}{\Longrightarrow} \beta$-$\text{Def}_2$.*

Both DI-results and ME-results are amendable to computational assumptions, for instance, it is often crucial to make certain complexity assumptions for a specific task to be solvable whilst satisfying a given CDP notion. If an ME-result is established under a complexity assumption, we say that it is an assumption-dependent ME (ADME) result, and analogously for other types of results. Finally, we also speak of definitions being *direct relaxations* of some other definition. By this we simply mean that the relaxed definition is directly implied by the other and that this relationship is apparent from the definitions.

3 Distributed Statistical DP (Variations in Distribution Model)

On a high level, a distribution model is a description of the different entities in a protocol interaction and the roles they play. Roughly, we consider there to be three different roles in a protocol – *clients*, *servers* and *analysts*. A client is a party (typically trusted) who has input, i.e. holds at the start of the protocol a part of the dataset D of interest for the DP mechanism. A server is a party that receives a function evaluation of some inputs and then sends some function of the results further. An analyst is an untrusted party that receives some mechanism output. It is the analyst that the mechanism is supposed to be DP against. These roles are not necessarily disjoint, meaning that one party can have several of them at once.

3.1 The Local Model

One major drawback of the central model of DP is the need to fully trust a central dataholder. In order to remove this trust assumption, a model was introduced where each client introduces some noise to their data before giving it to someone else. Since here the DP mechanism (also called the *local randomiser*) is computed by each client themselves, this is known as the *local model*, of which an illustration is found in Figure 2. One consequence of this approach, however, is that in many situations one must add much more noise than in the central model [4, 15, 52, 53]. Due to the very extensive literature about the local model, providing a broader faithful summary here is infeasible and since adopting the local model is

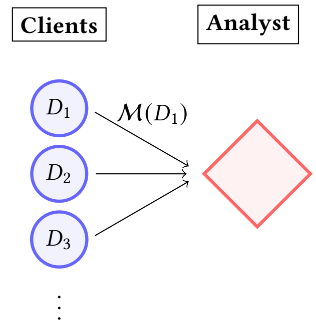

Figure 2: Overview of the local model. Here there are no servers so the clients send their processed datapoints directly to the analyst.

mainly an *alternative* to using MPC rather than a complement of it, we mostly consider the local model as out of scope for the rest of this work. There is great importance of the local model for the definitions we will study in more detail, though, namely in that it serves as a worst-case scenario on accuracy, meaning that using local randomisers directly leads to being DP in all the other distributed models as well and for a mechanism to be considered non-trivial it has to at least have higher utility than possible in the local model.

Definition 3.1 (DP in the local model [4, 23, 52]). A protocol π is (ε, δ)-SDP in the local model if the clients communicate exclusively with the analyst (who has no input) and the view of the analyst is (ε, δ)-SDP.

One important remark about the choice of distribution model in practice is that they are, to some extent, often geared towards different use cases directly. For instance, the local model can be seen as providing protection during the *collection* of data rather than merely the *disclosure* of information about an already assembled dataset. One illustrative example of this difference is the comparison between the US census bureau's system for releasing population statistics with DP [10] and systems from tech giants like Google and Apple for collecting user data with DP [20, 28, 72]. In the case of the Census bureau, their collection of population data is largely uncontroversial, it is even their legal obligation to do so, and therefore the central model of DP is suitable. For the large tech companies on the other hand, already the collection of detailed user information is arguably problematic and therefore it may be desirable to reduce trust in the one collecting the data by having the users themselves do local randomisation, i.e. to use the local (or shuffle) model of DP.

3.2 The Shuffle Model

The *shuffle model* of DP [3, 7, 16, 36, 71, 72] is an intermediate model between the central and local models, where the mechanism is run locally but there is still a central server (sometimes called curator) performing some computations. The point would be that this server, called the *shuffler*, now only needs to be semi-honest and performs only the relatively simple task of shuffling (randomly permuting) the dataset and forwarding it to the analyst (also called

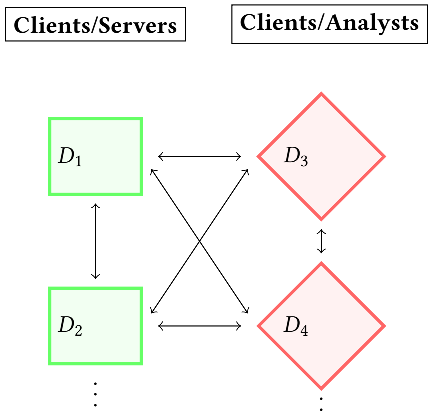

Figure 3: Description of the general distribution model as in BNO [4], which describes an N-party protocol where all parties hold inputs (can be void) and some of the parties are corrupted (become analysts rather than servers).

the aggregator). Nonetheless, one can in certain settings achieve accuracy much better than that in the local model and even at times the same as in the central model. Again, we do not discuss the shuffle model further in this work due to the size of the literature and lack of direct relevance to distributed CDP definitions.

3.3 Distributed SDP as in BNO

In BNO [4], the formal study of DP in a distributed protocol setting is largely initiated. As a part, the authors introduce what has come to be arguably the main definition of SDP for protocols, a definition we call *BNO-SDP*. The idea of this definition is a natural one, namely that a protocol is to be seen as SDP if whatever information the adversary gains from the protocol is an SDP mechanism of the inputs of the honest parties.[6] An illustration of the distribution model is found in Figure 3.

Definition 3.2 (BNO-SDP, reformulation and generalisation of [4]). Let COR $= (a, \mathcal{C})$ be a corruption model. We say that an N-party protocol π is $(\varepsilon_\kappa, \delta_\kappa)$-BNO-SDP with respect to COR if for all (also inefficient) adversaries $\mathcal{A}$ following COR and corrupting the parties in $C \in \mathcal{C}$, for all C-adjacent D, D' (Definition 2.3) and for all possible subsets S of combined views of the parties in C, we have

$$\mathbb{P}(\mathsf{VIEW}^{\mathcal{A}}_{\pi,C}(D) \in S) \le e^{\varepsilon_\kappa} \mathbb{P}(\mathsf{VIEW}^{\mathcal{A}}_{\pi,C}(D') \in S) + \delta_\kappa, \quad (3)$$

where the probabilities are taken over the randomness in π. If $\mathcal{C}$ is exactly all subsets of $[N]$ of size at most t and the corruptions are passive, we say that π is $(t, \varepsilon_\kappa, \delta_\kappa)$-BNO-SDP.

A special case of BNO-SDP is introduced in [26], called *distributed DP (DDP)*. The parties are classified as either clients or servers with only the clients having inputs and the servers getting shares of each client's inputs. DDP is then precisely BNO-SDP except for that only servers may be corrupted.

[6]The authors of BNO also propose a computational version of this definition, which we discuss in Section 5. We note that whilst the definitions are stated originally only with respect to passive corruptions, leading to a simpler treatment, they are in later works (such as MPRV) extended to also apply to the case of active corruptions. Also, BNO-SDP was originally introduced with respect to pure SDP (δ fixed to 0).

3.4 Other Distribution Models

BNO-SDP is sometimes weakened by requiring the transcript of the protocol (i.e. the messages sent) rather than the entire view of the corrupted parties (also including their initial inputs and random coins) to be SDP with respect to the inputs of the honest parties [60]. This is called *DP against an external observer*, which we denote BNO$_{ext}$-SDP. In [44], a version of BNO$_{ext}$-SDP is defined for *oracle-aided protocols*. There the analyst, additionally to the protocol transcript, also has query access to the same oracle (i.e. an abstract functionality that answers a specific type of queries) instance that is used in the protocol. Other settings for SDP that arguably are to be seen as variations in distribution model are those that consider interaction between the dataholder and the analyst. Such variations are used, for instance, in the context of adaptive composition of mechanisms [45, 51, 66] and DP under continual release [14, 24, 65].

3.5 Relations between Different Distribution Models

The choice of distribution model induces a trade-off between utility and trust assumptions where less restrictive trust assumptions lead to lower optimal utility. On this scale, the local and central models constitute the edges, meaning that no distribution model allows better optimal utility than the central one and no model has less trust assumptions than the local one. The utility gap between the distribution models is critically dependent on the task at hand. For instance, for computing integer sums, the additive error (with constant probability) can be $O(1/\varepsilon_\kappa)$ in central $(\varepsilon_\kappa, 0)$-SDP (by the Laplace mechanism) but for all local $(\varepsilon_\kappa, 0)$-SDP protocols, it is $\Omega(\sqrt{n}/\varepsilon_\kappa)$ [4]. The shuffle model lies strictly in between the local and central models, and the optimal accuracy achievable within it ranges from that of the local model to that of the central model, depending on the task and constraints on the communication complexity. The BNO-SDP model can be seen as a generalisation of the others, since depending on how one specifies π and COR one gets models that are equivalent to each of the others. For instance, one gets exactly the non-interactive local model if all parties but one hold input except for one, which is also the only corruptible one, and the parties with input can only send messages to the corruptible one. In practice, an often crucial difference between the distribution models is the efficiency of the various involved parties. For instance, if one goes from using the central to the local model then a, potentially significant, computational burden is shifted from the dataholder to the clients. Similarly, if one uses MPC (and therefore BNO-SDP) to avoid trusting a dataholder withot using the local model, then the computational costs rise quickly as the number of clients increase, therefore making it less feasible than using the local model in systems with many users (clients).

4 Central-model CDP (Variations in Computational Perspective)

Variations in computational perspective all formalise what it means for a mechanism to 'look DP' to an efficient analyst interacting with it. There are two main flavors of formalisations, *indistinguishability-based* ones and *simulation-based* ones. The indistinguishability-based definitions are created by taking the usual SDP definition and weakening the requirement that output closeness holds for all subsets of the range to that output closeness holds for the output distribution of any efficient distinguisher (analyst) acting on the mechanism output. Since the first requirement can be re-formulated as the output of any (possibly inefficient) distinguisher satisfying output closeness, the change amounts exactly to limiting the distinguisher to be efficient. Formally, indistinguishability-based CDP (IND-CDP) in the central model is defined as below. Note that the computational boundedness induces an asymptotic perspective in the definitions and we therefore must consider mechanism ensembles (indexed by the security parameter) as well as parameter ensembles $(\varepsilon_\kappa, \delta_\kappa)$ with possible dependence on κ. [7]

Definition 4.1 (IND-CDP for mechanisms, MPRV [62]). An ensemble $\{g_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ of mechanisms $g_\kappa : \mathcal{D} \to \mathcal{R}_\kappa$ is $\underline{(\varepsilon_\kappa, \delta_\kappa)\text{-IND-CDP}}$ if for every efficient distinguisher T, every sufficiently large κ, all adjacent $D, D' \in \mathcal{D}$ of polynomial size in κ, it holds that

$$\mathbb{P}(T(g_\kappa(D)) = 1) \leq e^{\varepsilon_\kappa}\mathbb{P}(T(g_\kappa(D')) = 1) + \delta_\kappa,$$

with the probability being over the randomness in g_κ and T.

Note that IND-CDP was originally introduced with δ_κ being fixed as negligible in κ and that setting $\delta_\kappa = 0$ causes the definition to collapse into being equivalent to $(\varepsilon_\kappa, 0)$-SDP.[8] Simulation-based CDP is perhaps a more direct formalisation of the idea that the mechanism looks SDP to any PPT distinguisher, because here the requirement is that there exists an SDP mechanism (called the *simulator*) from which the mechanism is computationally indistinguishable.

Definition 4.2 (SIM-CDP for mechanisms, MPRV [62]). An ensemble $\{g_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ of mechanisms $g_\kappa : \mathcal{D} \to \mathcal{R}_\kappa$ is $\underline{(\varepsilon_\kappa, \delta_\kappa)\text{-SIM-CDP}}$ if there exists an ensemble $\{\mathcal{M}_\kappa(\cdot)\}_{\kappa\in\mathbb{N}}$ of $(\varepsilon_\kappa, \delta_\kappa)$-SDP mechanisms $\mathcal{M}_\kappa : \mathcal{D} \to \mathcal{R}_\kappa$ such that for every sufficiently large κ and every $D \in \mathcal{D}$ of polynomial size in κ, it holds that $g_\kappa(D)$ and $\mathcal{M}_\kappa(D)$ are computationally indistinguishable.

Here we note that SIM-CDP was first defined with $\delta_\kappa = 0$ and that, as opposed to IND-CDP, this does not cause the definition to be equivalent to SDP. One intuitive explanation for this is that for this definition, the computational relaxation lies in the simulation, rather than in the output distribution of the real-world mechanism. In MPRV, an intermediate definition is also proposed, called $\text{SIM}_{\forall\exists}$-CDP. It is the same as SIM-CDP except that the order of the quantifiers is swapped, i.e. instead of there existing a simulator $\mathcal{M}$ for all datasets, it is allowed that each dataset has its own simulator associated with it. This definition is introduced as a technical tool used to study the relationship between IND-CDP and SIM-CDP.

[7]When clear from context we often suppress such dependencies and speak of mechanism (and protocol) ensembles simply as single mechanisms (and protocols).

[8]For completeness we include a proof of the equivalence between IND-CDP and SDP when $\delta_\kappa = 0$ in Appendix C.

4.1 Relations between CDP Definitions in the Central Model

It is easy to see that $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP directly implies $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP (via $(\varepsilon_\kappa, \delta_\kappa)$-$\text{SIM}_{\forall\exists}$-CDP) and this is shown already in MPRV [62] for $\delta_\kappa = 0$. The result extends immediately to arbitrary values of δ_κ, which we show in Appendix C.1.1. In MPRV it is also shown that $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP directly implies $(\varepsilon_\kappa, 0)$-$\text{SIM}_{\forall\exists}$-CDP when $\varepsilon_\kappa \in O(\log(\kappa))$. Thus the two definitions are equivalent for such ε_κ and $\delta_\kappa = 0$ but until now there are no known results on how they relate for non-zero δ_κ. It was left open in MPRV to separate $\text{SIM}_{\forall\exists}$-CDP and SIM-CDP with any type of result. Further, it is immediately clear that $(\varepsilon_\kappa, \delta_\kappa)$-SDP directly implies $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP for any parameter choice. Thus, the remaining questions to discuss are how $(\varepsilon_\kappa, \delta_\kappa)$-SDP, $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP and $(\varepsilon_\kappa, \delta_\kappa)$-$\text{SIM}_{\forall\exists}$-CDP relate. Summaries of relationships between the definitions can be seen in Figure 4 and Table 1.

4.2 Separating SIM-CDP from SDP

The first results about separating SIM-CDP and SDP were negative and quite general. It was shown in 2011 [39] (and then strengthened in 2016 [9]) that, roughly, they are equally expressive with respect to the set of tasks defined by a utility function which is a bound on an L_p norm for a low-dimensional output domain. That is, for such tasks, $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-SIM-CDP ME-implies $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-SDP and they are thus equivalently expressive. This means, essentially, that if there is a task for which there is a SIM-CDP mechanism that has significantly better utility than the best SDP mechanism, then *the dimension of the range of $\mathcal{M}$ must be large, or the utility function must be of a different kind.* Importantly, this remains true regardless of the complexity assumptions one relies on. Another aspect of the barriers established in [39] is that they consider mechanisms in the two settings of *roughly the same efficiency.* This suggests a third way of avoiding those barriers, namely to propose a task for which all SDP mechanisms have to be vastly slower than the most efficient SIM-CDP mechanism. This is done in 2016 [9] when it is proven that there are tasks for which there is an efficient CDP mechanism but all SDP mechanisms are inefficient. That is, it is established an *assumption-dependent infeasibility separation* between SDP and SIM-CDP. The task used in [9] is constructed specifically for the purpose of providing the desired separation. Therefore Vadhan poses the following open problem.

Open problem 1 (Vadhan's open problem 10.7 [67], reformulated – Still open). *Can an infeasibility separation between $(\varepsilon_\kappa, \delta_\kappa)$-SDP and $(\varepsilon_\kappa, 0)$-SIM-CDP be obtained using a more 'natural' utility function, such as the absolute error when answering counting queries?*

4.3 Separating SIM-CDP from IND-CDP

It is also remarked in [9] that if there is an ME-separation between IND-CDP and SDP, then that must imply an ME-separation between SIM-CDP and IND-CDP. No such separation was known until 2023 when Ghazi et al. [32] provided an ADME separation. This result simultaneously solves Vadhan's open problems 10.6 and 10.8.

$$\begin{tikzcd}[column sep=large] \boxed{(\varepsilon_\kappa, 0)\text{-SDP}} \arrow[r, shift left, "\text{DI}"] & \boxed{(\varepsilon_\kappa, 0)\text{-SIM-CDP}} \arrow[l, shift left, "/" marking, "\text{AD-Infeasibility [9]}"] \arrow[r, shift left, "\text{DI [62]}"] & \boxed{(\varepsilon_\kappa, 0)\text{-SIM}_{\forall\exists}\text{-CDP}} \arrow[l, shift left, "/" marking, "\text{ADME [32]}"] \arrow[r, shift left, "\text{DI [62]}"] & \boxed{(\varepsilon_\kappa, \mathsf{negl}(\kappa))\text{-IND-CDP}} \arrow[l, shift left, "\text{DI [62]}"] \end{tikzcd}$$

Figure 4: Overview of relations in the central model, with respect to arbitrary queries and complexity assumptions. Note that we state the results for constant $\varepsilon_\kappa > 0$ and $\delta_\kappa = 0$ although each of them extend also to wider parameter regimes(see Table 1).

Closed problem 1 (Vadhan's open problem 10.6 [67], reformulated – Closed positively in [32]). *Is there a computational task that is solvable in the central model in CDP but is impossible in SDP? In our terminology; is there an ADME-separation between $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-SDP and $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP?*

Closed problem 2 (Vadhan's open problem 10.8 [67], Reformulated – Closed negatively in [32]). *For every efficiently computable $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP mechanism $\mathcal{M} : \mathcal{D} \to R$, is there an $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-SDP mechanism $\mathcal{M}' : \mathcal{D} \to R$ such that for all $D \in \mathcal{D}$, $\mathcal{M}(D)$ is computationally indistinguishable from $\mathcal{M}'(D)$?*

Note that this question is equivalent to asking if all $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP mechanisms also are $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-SIM-CDP. The ADME-separation of Ghazi et al. is with respect to non-standard but arguably plausible complexity assumptions, thus suggesting investigations into what assumptions are needed to establish such ADME-separations. In particular, the question is posed what the *minimal assumption* needed for the separation is.

Open problem 2 (From discussion in [32]). *What is the minimal complexity assumption needed to ADME-separate $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP from $(\varepsilon_\kappa, \delta_\kappa)$-SDP?*

As a starting point, [39] shows that black-box use of a certain type of computational assumptions is not enough to arrive at any ADME-separations between IND-CDP and SDP. In particular, it is shown that black-box use of one-way functions and similar primitives[9] is not sufficient. This negative result means that for there to be an ADME-separation at all, the SIM-CDP mechanism must either make white-box use of the primitives or rely on stronger cryptographic assumptions (Ghazi et al. does the latter). The separation of Ghazi et al. uses a non-uniform task, meaning that the utility function u is dependent on κ, which suggests the open problem of finding other tasks (particularly uniform ones) on which the notions can be separated.

Open problem 3 (From discussion in [32]). *Establish an ADME-separation between $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP and $(\varepsilon_\kappa, 0)$-SDP, as the one in [32], for another task, such as a "more natural" task[10] or one that is uniform. In particular, are $(\varepsilon_\kappa, 0)$-SDP and $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP equally expressive in the set of uniform tasks in the central model?*

We finish this section by noting that since most of the known implications and separations are only for $\delta_\kappa = 0$ or with a non-zero δ_κ in only one of the involved definitions, an immediate open research area is to extend these results to other parameter regimes. We formulate this in two open problems.

[9] In particular, the primitives considered are those that can be instantiated as a random object. This includes *trapdoor permutations* and *collision-resistant hash functions*.
[10] Here, 'more natural' means essentially any task which is not specifically construction as to give the separation.

Open problem 4 (New). *Are $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP and $(\varepsilon_\kappa, \delta_\kappa)$-SIM$_{\forall\exists}$-CDP equivalent in the central model for all $\varepsilon_\kappa \in O(\log(\kappa))$, $\delta_\kappa > 0$?*

Open problem 5 (New). *For non-zero $\delta_\kappa, \delta'_\kappa$, establish an ADME-separation between $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP and $(\varepsilon_\kappa, \delta'_\kappa)$-SIM-CDP.*

5 Definitions of Distributed CDP

We now turn to CDP definitions outside the central model and focus on the distribution model of BNO. This is due to the generality of that model and it being the primary model used when combining DP and MPC. Just as in the central model, non-central CDP definitions are often categorised as either indistinguishability-based or simulation-based. Here, it is however useful to distinguish definitions whose formalisation of simulation is in the ideal/real paradigm of secure computation from those whose is not. We refer to those that use the ideal/real paradigm as *ideal/real-based CDP*.

5.1 Indistinguishability-based CDP

Additionally to the notion of BNO-SDP (Definition 3.2), BNO [4] also proposes a computational version of it, which we can call BNO-CDP. Later in MPRV, the definition of IND-CDP (Definition 4.1) is extended to the case of two-party protocols. It is also directly extendable to the case of multiple parties, resulting in a definition that is the same as that of BNO-CDP except that the adjacency notion is different, algorithms are non-uniform and corruptions are not necessarily passive. Since in this work, we ignore such differences (see Section 1.1), we consider BNO-CDP and IND-CDP (for protocols) to be two different instantiations of the same definition and for the rest of this work, we refer only to IND-CDP.

Definition 5.1 (IND-CDP for protocols, reformulation from [4, 67]). Let COR $= (a, C)$ be a corruption model. We say that an N-party protocol π is $(\varepsilon_\kappa, \delta_\kappa)$-IND-CDP with respect to COR if for all efficient adversaries $\mathcal{A}$ following COR and corrupting the parties in $C \in \mathcal{C}$, for all efficient distinguishers T, every sufficiently large κ and for all C-adjacent D, D', we have

$$\mathbb{P}\left(T\left(\mathsf{VIEW}^{\mathcal{A}}_{\pi,C}(D)\right) = 1\right) \leq e^{\varepsilon_\kappa}\mathbb{P}\left(T\left(\mathsf{VIEW}^{\mathcal{A}}_{\pi,C}(D')\right) = 1\right) + \delta_\kappa. \quad (4)$$

The probabilities are taken over the randomness in π, $\mathcal{A}$ and T.

There are two direct relaxations of IND-CDP in the literature, both of which are introduced specifically to aid in providing separations between IND-CDP and some other notion. The first such definition, call it IND$_{ext}$-CDP [42], is the computational analog of BNO$_{ext}$-SDP mentioned in Section 3, meaning that it is exactly as IND-CDP except that the distinguisher only has access to the transcript of the protocol. The other restricted IND-CDP definition,

Table 1: Summary of implications and separations in central-model CDP.

Result	Parameter restrictions	Paper	Comment
$(\varepsilon_\kappa, \delta_\kappa)$-SDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP	-	-	By definition
$(\varepsilon_\kappa, 0)$-SIM-CDP $\underset{(AD-Infeasibilty)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-SDP	$\varepsilon_\kappa \in O(\log(\kappa))$, $\delta_\kappa \leq 1/poly(\kappa)$	[9]	Non-uniform task
$(\varepsilon_\kappa, 0)$-SIM-CDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP	-	MPRV	
$(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP	$\varepsilon_\kappa \in O(\log(\kappa))$	This work	See Appendix C.1.1
$(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, 0)$-SIM$_{\forall\exists}$-CDP	$\varepsilon_\kappa \in O(\log(\kappa))$	MPRV	
$(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP $\underset{(ADME)}{\not\Longrightarrow}$ $(\varepsilon'_\kappa, \delta_\kappa)$-SIM-CDP	$\varepsilon_\kappa, \varepsilon'_\kappa > 0$ constant, $\delta_\kappa \leq 1/\kappa^{27}$	[32]	Non-uniform task

we call IND$_{sub}$-CDP [37] and is the same as IND-CDP except that the guarantee is only required to hold for a subset of protocol executions, for instance only requiring output closeness when all honest parties get output. In [14] a local model CDP definition is proposed which is quite similar to that of IND-CDP but has some important differences. In particular, the corruptions are modeled as a process (not necessarily static) and the probabilities in the output closeness inequality are also taken over the randomness in the corruption process.

5.2 Simulation-based CDP

Just as for IND-CDP, MPRV also proposes a version of SIM-CDP (Definition 4.2) for two-party protocols that can easily be extended to multi-party protocols. It is worth noting that the words 'simulation' and 'simulator' are used quite differently in the SIM-CDP definitions to what is the custom in simulation-based security definitions (such as those in the ideal/real paradigm). Essentially, the simulator (SDP mechanism) in SIM-CDP fulfills a role more akin to that of the ideal functionality (describe the desired behaviour of the protocol) than that of the simulator (map functionality outputs to something similar to party views) in a simulation-based definition of secure computation.

Definition 5.2 (SIM-CDP for protocols, reformulation from MPRV). Let $\mathsf{COR} = (a, \mathcal{C})$ be a corruption model. We say that an N-party protocol π is $\underline{(\varepsilon_\kappa, \delta_\kappa)\text{-SIM-CDP with respect to }\mathsf{COR}}$ if for all efficient adversaries $\mathcal{A}$ following COR and corrupting the parties in $C \in \mathcal{C}$, for all efficient distinguishers T and for all C-adjacent D, D', there exists an ensemble $\{\mathcal{M}_\kappa(\cdot)\}_{\kappa \in \mathbb{N}}$ of $(\varepsilon_\kappa, \delta_\kappa)$-SDP mechanisms $\mathcal{M}_\kappa : \mathcal{D} \to \mathcal{R}_\kappa$ such that for every sufficiently large κ and every $D \in \mathcal{D}$ of size polynomial in κ, it holds that $\mathsf{VIEW}^{\mathcal{A}}_{\pi,C}(D)$ and $\mathcal{M}_\kappa(D)$ are indistinguishable to T.

5.3 Ideal/real-based CDP

IND-CDP and SIM-CDP are formulated quite differently to the usual ways of defining secure computation in the MPC literature. Therefore another CDP definition, SIM$^+$-CDP, is proposed in MPRV which incorporates DP into the ideal/real paradigm. In the following, we assume familiarity with standard definitions of secure computation, for a brief introduction to the ideal/real paradigm we refer to Appendix A and references therein. One main advantage of operating within the ideal/real paradigm is that the entire possible influence of an adversary on the protocol execution is specified, such as how it can change the output of the protocol, rather than it only being regulated how much information the adversary can gain. In the ideal/real paradigm, the adversarial effect on the protocol is defined by the ideal functionality that dictates all intended properties of the protocol and thus it is a natural definitional approach to also incorporate DP in this ideal world. The SIM$^+$-CDP definition is as follows.[11]

Definition 5.3 (SIM$^+$-CDP, Reformulation of MPRV). Let u be a utility function. An N-party protocol π is $\underline{(\alpha, \varepsilon_\kappa, \delta_\kappa)\text{-SIM}^+\text{-CDP}}$ for u if there exists an $(\varepsilon_\kappa, \delta_\kappa)$-SDP mechanism $\mathcal{M}$ such that:

- the mechanism $\mathcal{M}$ is α-useful for u, and,
- π is a secure protocol for the functionality $\mathcal{F}^{\mathcal{M}}_{SFE}$ (the ideal functionality that evaluates $\mathcal{M}$ when given inputs from all parties) as per Definition A.2 (standalone security with perfect correctness, efficient protocols and a potentially inefficient simulator).

In 2024, [61] argues that the used notion of secure computation is slightly too restrictive for SIM$^+$-CDP to be achievable when $\mathcal{M}$ is a canonical SDP mechanism. The argument is based on that the security notion used in SIM$^+$-CDP demands that π runs in strict polynomial time and that the protocol has perfect correctness, i.e. has exactly the same output distribution as the ideal functionality when there are no active corruptions. This leads to that many standard SDP mechanisms (such as the Laplace and Gaussian mechanisms) cannot take the role of $\mathcal{M}$ in the SIM$^+$-CDP definition, since they cannot be sampled exactly in strict polynomial time. In light of this, a new ideal/real-based CDP definition is proposed where, among other things, the notion of secure computation and the correctness requirement are changed.

Definition 5.4 (SIM*-CDP, Reformulation of [61]). An N-party protocol π is $\underline{(\varepsilon_\kappa, \delta_\kappa)\text{-SIM}^*\text{-CDP for the ideal functionality }\mathcal{F}}$ and a given adjacency notion ADJ if π UC-realises $\mathcal{F}$ and for all efficient ideal-world adversaries $\mathcal{S}$, the view of $\mathcal{S}$ is $(\varepsilon_\kappa, \delta_\kappa)$-SDP with respect to ADJ.

In [61], a more generalised version called SIM$^\circ$-CDP is also introduced, which is identical to SIM*-CDP except that the definition of secure computation is kept variable. An direct relaxation of

[11] The formulation in MPRV is quite different from the one we have here, in particular in the modeling of protocols and with regard to usefulness, where a different formulation of utility is used. The adapted definition of standalone security is defined in the long version of the MPRV paper, which is available from the authors. We thank them for providing it and for answering our questions.

SIM$^+$-CDP is used in [5], where the correctness is relaxed to be computational rather than statistical (i.e. an inefficient adversary can violate the correctness of the honest parties' output with non-negligible probability). It is immediate that SIM$^+$-CDP directly implies this relaxed notion and that it strictly stronger than SIM-CDP in the same way that SIM$^+$-CDP and SIM*-CDP are.

6 Relations between the Distributed CDP Definitions

When showing relations between distributed CDP notions, one must do so with respect to a given family of functionalities, distribution model and corruption model. All of the results in this section are with respect to passive (semi-honest) corruptions and the functionality of *secure function evaluation (SFE)*, i.e. the ideal functionality that evaluates a fixed function when given inputs from all parties. The choice to work with passive corruptions is not only because it is easier but also because most results are in the shape of (or follow from) lower bounding the error in a protocol and if one can establish those with respect to passive corruptions, then the results carry over to the case of active corruptions. Restricting the study to SFE is similarly due to it being an extremely general functionality. There are however reactive functionalities that cannot be reduced to (non-reactive) SFE [40, 47, 48]. Better understanding the relations between CDP definitions for other types of corruptions and functionalities are exciting open research areas. Further, almost all results we survey are established with respect to δ_κ fixed as either 0 or negligible and also here extending the results to other parameter regimes lies largely open. Overviews of known results are given in Figure 5 and Table 2.

6.1 Separating the CDP Definitions

The results regarding IND-CDP and SIM-CDP directly extend from the central model, meaning that $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP directly implies $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP and that $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP is ADME-separated from $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP by the result of Ghazi et al. [32]. $(\varepsilon_\kappa, 0)$-SIM$^+$-CDP is a strictly stronger definition than both of them, as is already shown in MPRV. More precisely, $(\varepsilon_\kappa, 0)$-SIM$^+$-CDP directly implies $(\varepsilon_\kappa, 0)$-SIM-CDP but the same does not hold in the other direction. Further, it is easy to see that $(\varepsilon_\kappa, \delta_\kappa)$-SIM$^+$-CDP must also be ME-separated from $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP for some parameters since SIM-CDP allows inefficient protocols but SIM$^+$-CDP does not. We show one such simple separation in Proposition C.3. Some results relating SIM*-CDP to the other definitions are given in [61], more precisely is it shown that $(\varepsilon_\kappa, \delta_\kappa)$-SIM*-CDP directly implies $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP for all parameters and that there are parameters such that $(\varepsilon_\kappa, 0)$-SIM-CDP is ADME-separated from $(\varepsilon_\kappa, \delta_\kappa)$-SIM*-CDP. Further, it is shown that $(\varepsilon_\kappa, 0)$-SIM$^+$-CDP and $(\varepsilon_\kappa, 0)$-SIM*-CDP are ADME-separated from each other in both directions. None of these results are really surprising, considering that SIM*-CDP can be seen as a version of SIM$^+$-CDP that has been both relaxed (using computational rather than perfect correctness) and strictened (using UC instead of standalone security).

6.2 Separating CDP and BNO-SDP

Already in BNO [4], several ADME-separations are obtained and thus the understanding of the relationship between SDP and CDP in the multi-party setting was always a few steps ahead of that in the central model. This is unsurprising since there is already a large and well-understood literature on what functionalities can be computed under what complexity assumptions in multi-party protocols. For instance, it is known that for the case of dishonest majorities (such as in the two-party case), there are functionalities (such as evaluating an AND gate) that cannot be realised without complexity assumptions whereas any efficiently computable functionality can be realised under the assumption that there exists a protocol for oblivious transfer (OT) [17, 56]. On the other hand, if there are more than two parties and a majority of them are honest, then any PPT functionality can be securely realised even without computational assumptions, and thus for the discussion about relating the different DP definitions, we focus exclusively on the two-party case. The main strategy for separating BNO-SDP from various CDP definitions is to choose the task of computing an N-ary PPT computable function with the same error as in the central model (up until a negligible decrease) and then derive a lower bound on the error in BNO-SDP that rules out solving said task. The existence of general-purpose MPC implies that it can be solved under the assumption of OT and since a protocol that computes an SDP mechanism with perfect correctness and computational security in the standard standalone security model directly satisfies SIM$^+$-CDP, an ADME-separation between BNO-SDP and SIM-CDP and IND-CDP follows via SIM$^+$-CDP. An analog argument holds with respect to SIM*-CDP (with the OT protocol being UC-secure). This strategy has proven remarkably successful and has yielded ADME-separations between BNO-SDP and all CDP variants for integer sums [4, 13], binary inner-products [60] and general boolean functions [38]. This success motivates the search for the minimal sufficient assumption to arrive at an ADME-separation for a given functionality. That is, we know that OT is sufficient but could it be enough to assume, say, the existence of one-way functions (OWF) or key-agreement protocols (KA)? This is captured in one of Vadhan's open problems:

Open problem 6 (Open problem 10.3 in [67], Reformulated). *What is the minimal complexity assumption needed to construct a task that can be solved by a CDP protocol but is impossible for any SDP protocol? In our words; What is the weakest complexity assumption under which there is a task that ADME-separates $(\varepsilon_\kappa, \delta_\kappa)$-BNO-SDP and $(\varepsilon_\kappa, \delta_\kappa)$-IND-CDP?*

There has been much progress on this question, mostly in the shape of results proving that a given complexity assumption is necessary for a given class of functionalities, with respect to $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP and $(\varepsilon_\kappa, 0)$-BNO-SDP. We now overview such results and summarise them in Table 3. Note that these results are only partial answers to the open problem above. In particular, understanding the necessary and sufficient assumptions for a separation with respect to other functionalities (or larger families of functions), other versions of CDP and parameter regimes is almost entirely open.

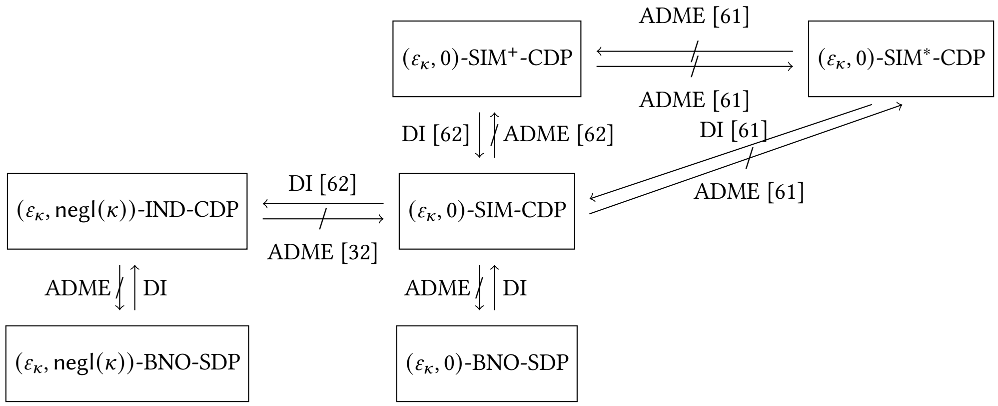

Figure 5: Overview of implications and separations for the setting of two-party SFE. All results are for passive adversaries except for the ADME separations to SIM*-CDP. Note that we state the results for $\delta_\kappa = 0$ or $\delta_\kappa = \mathsf{negl}(\kappa)$ although some of them extend also to larger δ_κ (see Table 2).

Table 2: Summary of relationships between definitions in distributed CDP for two-party SFE. All results are for passive adversaries if not otherwise stated.

Result	Parameter restriction	Paper	Comment
$(\varepsilon_\kappa, \delta_\kappa)$-BNO-SDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-IND-CDP	-	-	By definitions directly
$(\varepsilon_\kappa, \delta_\kappa)$-IND-CDP $\underset{(ADME)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-BNO-SDP	-	E.g. [4]	By impossibility of inf.-theoretic 2PC
$(\varepsilon_\kappa, \delta_\kappa)$-BNO-SDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP	-	-	By definitions directly
$(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP $\underset{(ADME)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-BNO-SDP	-	E.g. [4]	By impossibility of inf.-theoretic 2PC
$(\varepsilon_\kappa, 0)$-SIM-CDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP	-	MPRV [62]	
$(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP	$\varepsilon_\kappa \in O(\log(\kappa))$	This work	See Appendix C.1.1
$(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP $\underset{(ADME)}{\not\Longrightarrow}$ $(\varepsilon'_\kappa, \delta_\kappa)$-SIM-CDP	$\varepsilon_\kappa, \varepsilon'_\kappa > 0$, $\delta_\kappa \leq 1/\kappa^{27}$	[32]	By applying central-model result
$(\varepsilon_\kappa, \delta_\kappa)$-SIM^{+}-CDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP	-	MPRV [62]	
$(\varepsilon_\kappa, 0)$-SIM-CDP $\underset{(DI)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, 0)$-SIM^{+}-CDP	-	MPRV [62]	
$(\varepsilon_\kappa, 0)$-SIM-CDP $\underset{(ME)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, 0)$-SIM^{+}-CDP	Fixed ε_κ only	This work	Proposition C.3
$(\varepsilon_\kappa, \delta_\kappa)$-SIM*-CDP $\underset{(DI)}{\Longrightarrow}$ $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP	-	[61]	
$(\varepsilon_\kappa, 0)$-SIM-CDP $\underset{(ADME)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, 0)$-SIM*-CDP	Fixed ε_κ only	[61]	Only for active corruptions.
$(\varepsilon_\kappa, 0)$-SIM^{+}-CDP $\underset{(ADME)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, 0)$-SIM*-CDP	Fixed ε_κ only	[61]	Only for active corruptions.
$(\varepsilon_\kappa, 0)$-SIM*-CDP $\underset{(ADME)}{\not\Longrightarrow}$ $(\varepsilon_\kappa, 0)$-SIM^{+}-CDP	Fixed ε_κ only	[61]	By SIM^{+}-CDP needing PPT functionality

Table 3: Summary of sufficient and necessary assumptions for there existing a $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP protocol for the function in question with optimal accuracy (equal to that in the central model with $(\varepsilon_\kappa, 0)$-SDP). For AND, the known largest necessary assumption is different between optimal accuracy a non-trivial accuracy (i.e. the best possible with $(\varepsilon_\kappa, 0)$-BNO-SDP). The result in paranthesis is with respect to $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND$_{ext}$-CDP. *OT* stands for *oblivious transfer* and *KA* for *key agreement*.

Function	Weakest known sufficient assumption	Strongest known necessary assumption
XOR	OT	OT [37, 38, 41, 43, 55]
AND	OT	KA (optimal accuracy) [55]/OWFs (non-trivial accuracy) [38]
BIP	OT (KA [42])	KA [42]

Assumptions needed for separations via boolean functions. We have the best understanding for functionalities that evaluate boolean functions. For them, utility is typically measured in the probability that the protocol computes the function in question correctly. For any boolean function, the *optimal utility* (i.e. the best attainable utility with pure SDP in the central model) is $\alpha^* := \lambda/(\lambda+1)$, with $\lambda := e^{\varepsilon}_{\kappa}$, by the optimality of the randomised response mechanism [54]. On the other hand, in two-party $(\varepsilon_\kappa, 0)$-BNO-SDP, the best possible utility for XOR is $\alpha^{XOR} := \frac{1+\lambda^2}{(1+\lambda)^2}$ and for AND is $\alpha^{AND} := \frac{\lambda(\lambda^2+\lambda+2)}{(1+\lambda)^3}$, as shown in [38]. Therefore, the quest for minimal assumptions is to find out when the accuracy lies in $[\alpha^*, \alpha^{XOR})$ or $[\alpha^*, \alpha^{AND})$, respectively. In a string of results [37, 38, 41, 43, 55], it is shown that for XOR, OT is indeed not only sufficient but also necessary to achieve a non-trivial accuracy, i.e. accuracy non-negligibly above α^{XOR}. For AND, the current understanding is that to get optimal accuracy, assuming that key agreement protocols exist is necessary [55], and for a non-trivial accuracy, one needs at least OWFs [38].

Open problem 7 (From discussion in [41]). *What is the minimal complexity assumption sufficient for the existence of a two-party protocol computing AND with non-trivial accuracy and $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP?*

Assumptions needed for separations via inner-products. For binary inner-products (BIPs), the utility has typically been measured as the additive error occurring with constant probability, averaged over uniform inputs. For this setting, the optimal error is $O(1/\varepsilon_\kappa)$ by use of, say, the geometric mechanism [33] and the best possible error with $(\varepsilon_\kappa, 0)$-BNO-SDP is $\Omega\left(\frac{\sqrt{n}}{\lambda \log(n)}\right)$, with n being the number of elements in the vectors [60]. In 2022, [42] showed that in order to do significantly better than this lower bound, one must assume the existence of a key agreement protocol. In particular, any $(\varepsilon_\kappa, 1/n^2)$-IND-CDP protocol for BIP with error $O(\sqrt{n})$ can be used to construct a key agreement protocol. Further, it is shown that for the relaxed notion of $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND$_{ext}$-CDP, key agreement is both necessary and sufficient.[12]

The strategy above of establishing a separation between notions by showing a strict gap in the best achievable utility or accuracy within each notion also suggests measuring the size of that separation as the size of the gap. Doing so lets us understand the practical implications of having SDP rather than CDP protocols (or distributed SDP instead of central model SDP).

Open problem 8 (Question 3 in [60], reformulated). *What is the largest gap in accuracy between statistical and computational two-party protocols? In our terminology; Given a measure of accuracy, a distributed CDP definition and parameter regimes, what is the largest difference in the accuracy of any function between the best two-party protocols satisfying BNO-SDP and CDP in the given parameter regimes, respectively (under arbitrary complexity assumptions)?*[13]

[12] Before 2022, the state of understanding was limited to that there exists no protocol avoiding the lower bound of [60] in the random-oracle model [44].

[13] As a partial answer to this problem, [60] shows that with accuracy measured with respect to additive error (with constant probability), there exists a function over two n-bit string for which there is a linear gap (in n) between $(\varepsilon_\kappa, 0)$-BNO-SDP and $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP.

7 Discussion – Practical Differences between Distributed CDP Definitions

7.1 On the Semantics of the Definitions

We now discuss how one might go about choosing a distributed DP definition and instantiating it for a given use case. Firstly, we note that the choice of distribution model is essentially entirely decided by the problem at hand and therefore we consider only the choice of computational perspective. As is clear from the previous section, there is a more or less strict ordering in the expressiveness of the CDP definitions, with the indistinguishability-based definitions allowing better utility than the simulation-based and ideal/real-based ones. This means that it could be that for the functionality and utility measure one has, the maximum utility one can achieve is higher if one opts for, say, IND-CDP rather than SIM-CDP or SIM$^+$-CDP. Similarly, opting for a CDP guarantee rather than BNO-SDP (with comparable parameters, more on that below) can lead to higher utility, and never worse. When it comes to the privacy guarantees, we similarly know that, in theory, there is an inverted ordering between the CDP definitions to the one regarding utility. In practice, however, we are aware of no results on the practical impact of such differences. If one considers the ideal/real-based definitions, the picture becomes slightly different because those definitions do not only demand privacy (in the sense of bounding the information learned by the adversary) but also security and correctness, in the sense of having clear specifications of the influence an adversary can have on the computation. Since those extra requirements are not only theoretical but also practical, these definitions do have a clear practical advantage over, say, IND-CDP and SIM-CDP in the guarantees they make. On the other hand, one can analyse security properties of a protocol separately from its differential privacy, say, by proving the protocol is both IND-CDP and securely realises a given functionality. There is however not only a theoretical and intuitive advantage in having the DP guarantees part of the specification of the ideal world but also a practical one, since then being DP is also a property of the protocol which is preserved under composition, contrary to when the DP property is analysed solely in the real world.

In summary, whereas the theoretical relationships between the various CDP definitions are starting to become better understood, the practical impact of the theoretical differences is mostly unexplored. Therefore, a pragmatic approach to choosing a CDP definition to work with would be to simply choose the strongest one known which readily follows from the techniques one intends to use. In particular, if one uses well-known MPC techniques to implement an SDP mechanism, it is likely that one can directly deduce that the resulting protocol will satisfy SIM$^+$-CDP, SIM*-CDP, or a version of them. If on the other hand, one uses techniques that do not directly yield security guarantees such as the ones required by the ideal/real-based CDP definitions (say if the security of one's protocol is asserted by a game-based proof) then it is likely that one is better off analysing the views of the adversary directly and from that derive an IND-CDP or SIM-CDP guarantee.

Open problem 9 (New). *Evaluate a CDP definition with respect to its guarantees of protection against a class of attacks, such as reconstruction attacks or membership inference attacks, and compare it to the corresponding guarantees of SDP or another CDP definition.*

7.2 On Parameter Choices

As in the general literature on DP, the questions about what constitutes "good" parameter choices and what the qualitative differences are between parameter regimes remain poorly understood also with respect to distributed and computational DP definitions. For CDP, understanding different parameter regimes is arguably even harder than in statistical DP, because the parameters (especially δ) now play a somewhat dual role in that they can be either solely DP parameters or function also as a computational slack. For instance, when going from IND-CDP to SIM-CDP, a negligible δ term can be converted into a computational distance in the simulation, as seen in that the analog to $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP is $(\varepsilon_\kappa, 0)$-SIM-CDP. One way of largely avoiding the different interpretations and roles of the parameters within the different CDP definitions is to stick to the parameter regimes in which they were originally proposed, with $\delta_\kappa = \mathsf{negl}(\kappa)$ in BNO-SDP and IND-CDP and $\delta_\kappa = 0$ in the others. The problem with this approach, however, is that for practical reasons one might strongly prefer using, say, SIM$^+$-CDP with non-zero (and non-negligible) δ_κ, such as in [5]. The practical reasons might, for instance, be that one's system is highly composed and thus can achieve higher utility by using $(\varepsilon_\kappa, \delta_\kappa)$-SDP or that one wants to approximate an $(\varepsilon_\kappa, 0)$-SDP mechanism to decrease the runtime. Therefore it is of large practical importance to build a better understanding of what happens theoretically when the CDP definitions are relaxed to work in other parameter regimes than in which they were originally posed.

8 Conclusion

We have surveyed the literature on distributed and computational DP definitions, reformulated them to unify notation and highlight their used distribution model and computational perspective, and summarised known results on the relations between the definitions. The CDP definitions (both in the central model and in a multi-party setting) can be sorted in a rough hierarchy where a loss in utility can be traded for improved privacy parameters or lesser trust assumptions. Whether the ordering is strict or not depends on the specifics of the functionality, distribution model and parameter regime. A clear characterisation of when the various definitions are separated is, however, lacking for all but a few functionalities and settings. While much progress has been made on understanding the definitions from a theoretical angle in recent years, there are still many research directions lying largely unexplored, and this holds true also on the practical side. Two such broad directions are formulated in the open problems below.

Open problem 10 (New). *Find separations between CDP definitions under other constraints than complexity assumptions or protocol runtime, such as the efficiency of the simulator or the runtime of the adversary.*[14]

[14]For instance, in [32] is it noted that the separation established there between central-model SDP and CDP does not hold against quasi-polynomial adversaries.

Open problem 11 (New). *Relate the CDP notions to one another within stricter adversarial models, such as with active or adaptive corruptions.*

Whilst the lack of understanding for practical separations is unsatisfactory, it may also be seen as indication that in practical settings, the choice of which CDP definition to use can with reason be made according to how conveniently it fits the techniques one intends to use. For instance, we know of no practically relevant task that can be solved with IND-CDP instead of SIM-CDP and therefore one's choice between them, for a practical use case, will likely not affect whether the task can be solved or not. Similarly, we know of little good reason to have less faith in the concrete privacy guarantees given by IND-CDP than those given by SIM-CDP, even though IND-CDP is theoretically weaker.

Besides the two above, we have posed three other new problems (Open problems 4,5 and 9 – about extending the study of CDP definitions to new parameter regimes and relating the CDP guarantees to specific attack vectors). We have also revisited the four open problems proposed by Vadhan [67] regarding CDP, out of which two have been essentially solved (Closed problems 1 and 2 – about finding ME-separations between IND-CDP and SDP in the central model) and two are still mostly open (Open problems 1 and 6 – about finding a more natural infeasibility separation between SDP and SIM-CDP and minimal complexity assumptions needed for separations in the two-party model). Finally, we have re-iterated and reformulated four open problems from the discussions in recent works [32, 41, 60] (Open problems 2 and 7 – about finding minimal complexity assumptions for ME-separating IND-CDP and SDP in the central model or for getting non-trivial accuracy for the AND gate in the two-party model, and Open problems 3 and 8 – about finding a more natural task for separating IND-CDP and SDP in the central model or finding the largest accuracy gap between them in the two-party model).

There is a deep connection between distributed and computational DP and other areas in the theory of computing, such as randomness extractors, pseudodensity and communication complexity [41, 60, 62]. This together with the increasing practical maturity of DP and MPC, makes us hopeful that there will be much interesting work about understanding and using the notions we have surveyed and we hope that our survey and discussion may serve as a useful guide and introduction to researchers entering the field.

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. We thank Lea Demelius and Peter Waldert for reading preliminary drafts of this work and providing many helpful comments. We also thank the anonymous PoPETS reviewers for suggestions that have greatly improved the clarity in this work and its connection to real-world use cases.

References

[1] Archita Agarwal, Maurice Herlihy, Seny Kamara, and Tarik Moataz. 2019. Encrypted Databases for Differential Privacy. *Proceedings on Privacy Enhancing Technologies* 2019 (2019), 170–190. https://doi.org/10.2478/popets-2019-0042

[2] Alexandros Bakas, Antonis Michalas, and Tassos Dimitriou. 2022. Private Lives Matter: A Differential Private Functional Encryption Scheme. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy* (Baltimore, MD, USA) *(CODASPY '22)*. Association for Computing Machinery, New York, NY, USA, 300–311. https://doi.org/10.1145/3508398.3511514

[3] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2019. The Privacy Blanket of the Shuffle Model. In *Advances in Cryptology – CRYPTO 2019*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer International Publishing, Cham, 638–667.

[4] Amos Beimel, Kobbi Nissim, and Eran Omri. 2008. Distributed Private Data Analysis: Simultaneously Solving How and What. In *Advances in Cryptology – CRYPTO 2008*, David Wagner (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 451–468.

[5] James Bell, Adrià Gascón, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Mariana Raykova, and Phillipp Schoppmann. 2022. Distributed, Private, Sparse Histograms in the Two-Server Model. In *CCS '22* (Los Angeles, CA, USA). Association for Computing Machinery, New York, NY, USA, 307–321. https://doi.org/10.1145/3548606.3559383

[6] Ari Biswas and Graham Cormode. 2023. Interactive Proofs For Differentially Private Counting. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda (Eds.). ACM, Copenhagen, 1919–1933. https://doi.org/10.1145/3576915.3616681

[7] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. 2017. Prochlo: Strong Privacy for Analytics in the Crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles* (Shanghai, China) *(SOSP '17)*. Association for Computing Machinery, New York, NY, USA, 441–459. https://doi.org/10.1145/3132747.3132769

[8] Dan Boneh, Amit Sahai, and Brent Waters. 2011. Functional Encryption: Definitions and Challenges. In *Theory of Cryptography*, Yuval Ishai (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 253–273.

[9] Mark Bun, Yi-Hsiu Chen, and Salil Vadhan. 2016. Separating Computational and Statistical Differential Privacy in the Client-Server Model. In *Theory of Cryptography*, Martin Hirt and Adam Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 607–634.

[10] Population Reference Bureau, the U.S. Census Bureau's 2020 Census Data Products, and Dissemination Team. 2023. *Why the Census Bureau Chose Differential Privacy*. Technical Report. US Census Bureau. https://www.census.gov/library/publications/2023/decennial/c2020br-03.html

[11] Ran Canetti. 2000. Security and Composition of Multiparty Cryptographic Protocols. *J. Cryptol.* 13, 1 (jan 2000), 143–202. https://doi.org/10.1007/s001459910006

[12] Ran Canetti. 2000. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Paper 2000/067. https://eprint.iacr.org/2000/067

[13] T-H. Hubert Chan, Elaine Shi, and Dawn Song. 2012. Optimal Lower Bound for Differentially Private Multi-party Aggregation. In *Algorithms – ESA 2012*, Leah Epstein and Paolo Ferragina (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 277–288.

[14] T. H. Hubert Chan, Elaine Shi, and Dawn Song. 2012. Privacy-Preserving Stream Aggregation with Fault Tolerance. In *Financial Cryptography and Data Security*, Angelos D. Keromytis (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 200–214.

[15] Albert Cheu, Adam Smith, and Jonathan Ullman. 2021. Manipulation Attacks in Local Differential Privacy. *Journal of Privacy and Confidentiality* 11, 1 (Feb. 2021). https://doi.org/10.29012/jpc.754

[16] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. 2019. Distributed Differential Privacy via Shuffling. In *Advances in Cryptology – EUROCRYPT 2019*, Yuval Ishai and Vincent Rijmen (Eds.). Springer International Publishing, Cham, 375–403.

[17] B. Chor and E. Kushilevitz. 1989. A zero-one law for Boolean privacy. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing* (Seattle, Washington, USA) *(STOC '89)*. Association for Computing Machinery, New York, NY, USA, 62–72. https://doi.org/10.1145/73007.73013

[18] Ronald Cramer, Ivan Bjerre Damgrd, and Jesper Buus Nielsen. 2015. *Secure Multiparty Computation and Secret Sharing* (1st ed.). Cambridge University Press, USA.

[19] Damien Desfontaines and Balázs Pejó. 2020. SoK: Differential privacies. *Proceedings on Privacy Enhancing Technologies* 2020 (2020), 288 – 313.

[20] Apple differential privacy team. 2017. Learning with privacy at scale. https://machinelearning.apple.com/research/learning-with-privacy-at-scale

[21] Cynthia Dwork. 2006. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II* (Venice, Italy) *(ICALP'06)*. Springer-Verlag, Berlin, Heidelberg, 1–12. https://doi.org/10.1007/11787006_1

[22] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology - EUROCRYPT 2006*, Serge Vaudenay (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 486–503.

[23] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography* (New York, NY) *(TCC'06)*. Springer-Verlag, Berlin, Heidelberg, 265–284. https://doi.org/10.1007/11681878_14

[24] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (Cambridge, Massachusetts, USA) *(STOC '10)*. Association for Computing Machinery, New York, NY, USA, 715–724. https://doi.org/10.1145/1806689.1806787

[25] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. 2010. Boosting and Differential Privacy. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS '10)*. IEEE Computer Society, USA, 51–60. https://doi.org/10.1109/FOCS.2010.12

[26] Fabienne Eigner, Aniket Kate, Matteo Maffei, Francesca Pampaloni, and Ivan Pryvalov. 2014. Differentially private data aggregation with optimal utility. In *Proceedings of the 30th Annual Computer Security Applications Conference* (New Orleans, Louisiana, USA) *(ACSAC '14)*. Association for Computing Machinery, New York, NY, USA, 316–325. https://doi.org/10.1145/2664243.2664263

[27] Reo Eriguchi, Atsunori Ichikawa, Noboru Kunihiro, and Koji Nuida. 2023. Efficient Noise Generation Protocols for Differentially Private Multiparty Computation. *IEEE Transactions on Dependable and Secure Computing* 20, 6 (2023), 4486–4501. https://doi.org/10.1109/TDSC.2022.3227568

[28] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 1054–1067. https://doi.org/10.1145/2660267.2660348

[29] Ferran Alborch Escobar, Sébastien Canard, Fabien Laguillaumie, and Duong Hieu Phan. 2024. Computational Differential Privacy for Encrypted Databases Supporting Linear Queries. Cryptology ePrint Archive, Paper 2024/048. https://eprint.iacr.org/2024/048

[30] Daniel Escudero. 2022. An Introduction to Secret-Sharing-Based Secure Multiparty Computation. Cryptology ePrint Archive, Paper 2022/062. https://eprint.iacr.org/2022/062

[31] Johannes Gehrke, Edward Lui, and Rafael Pass. 2011. Towards Privacy for Social Networks: A Zero-Knowledge Based Definition of Privacy. In *Theory of Cryptography*, Yuval Ishai (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 432–449.

[32] Badih Ghazi, Rahul Ilango, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. 2023. Towards Separating Computational and Statistical Differential Privacy. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, Santa Cruz, CA, 580–599. https://doi.org/10.1109/FOCS57990.2023.00042

[33] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2012. Universally Utility-maximizing Privacy Mechanisms. *SIAM J. Comput.* 41, 6 (2012), 1673–1693. https://doi.org/10.1137/09076828X

[34] Oded Goldreich. 2001. *Foundations of Cryptography: Volume 1, Basic Techniques*. Cambridge University Press, Cambridge.

[35] Oded Goldreich. 2004. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, USA.

[36] Dov Gordon, Jonathan Katz, Mingyu Liang, and Jiayu Xu. 2022. Spreading the Privacy Blanket: Differentially Oblivious Shuffling for Differential Privacy. In *Applied Cryptography and Network Security: 20th International Conference, ACNS 2022, Rome, Italy, June 20–23, 2022, Proceedings* (Rome, Italy). Springer-Verlag, Berlin, Heidelberg, 501–520. https://doi.org/10.1007/978-3-031-09234-3_25

[37] Vipul Goyal, Dakshita Khurana, Ilya Mironov, Omkant Pandey, and Amit Sahai. 2016. Do Distributed Differentially-Private Protocols Require Oblivious Transfer?. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 55)*, Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 29:1–29:15. https://doi.org/10.4230/LIPIcs.ICALP.2016.29

[38] Vipul Goyal, Ilya Mironov, Omkant Pandey, and Amit Sahai. 2013. Accuracy-Privacy Tradeoffs for Two-Party Differentially Private Protocols. In *Advances in Cryptology – CRYPTO 2013*, Ran Canetti and Juan A. Garay (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 298–315.

[39] Adam Groce, Jonathan Katz, and Arkady Yerukhimovich. 2011. Limits of Computational Differential Privacy in the Client/Server Setting. In *Theory of Cryptography*, Yuval Ishai (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 417–431.

[40] Adam Groce, Peter Rindal, and Mike Rosulek. 2019. Cheaper Private Set Intersection via Differentially Private Leakage. *Proceedings on Privacy Enhancing Technologies* 2019 (07 2019), 6–25. https://doi.org/10.2478/popets-2019-0034

[41] Iftach Haitner, Noam Mazor, Ronen Shaltiel, and Jad Silbak. 2019. Channels of Small Log-Ratio Leakage and Characterization of Two-Party Differentially Private Computation. In *Theory of Cryptography*, Dennis Hofheinz and Alon Rosen (Eds.). Springer International Publishing, Cham, 531–560.

[42] Iftach Haitner, Noam Mazor, Jad Silbak, and Eliad Tsfadia. 2022. On the Complexity of Two-Party Differential Privacy. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (Rome, Italy) *(STOC 2022)*. Association for Computing Machinery, New York, NY, USA, 1392–1405. https://doi.org/10.1145/3519935.3519982
[43] Iftach Haitner, Kobbi Nissim, Eran Omri, Ronen Shaltiel, and Jad Silbak. 2018. Computational Two-Party Correlation: A Dichotomy for Key-Agreement Protocols. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, Paris, 136–147. https://doi.org/10.1109/FOCS.2018.00022
[44] Iftach Haitner, Eran Omri, and Hila Zarosim. 2016. Limits on the Usefulness of Random Oracles. *J. Cryptol.* 29, 2 (apr 2016), 283–335. https://doi.org/10.1007/s00145-014-9194-9
[45] Samuel Haney, Michael Shoemate, Grace Tian, Salil Vadhan, Andrew Vyrros, Vicki Xu, and Wanrong Zhang. 2023. Concurrent Composition for Interactive Differential Privacy with Adaptive Privacy-Loss Parameters. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (, Copenhagen, Denmark,) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 1949–1963. https://doi.org/10.1145/3576915.3623128
[46] Xi He, Ashwin Machanavajjhala, Cheryl Flynn, and Divesh Srivastava. 2017. 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* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1389–1406. https://doi.org/10.1145/3133956.3134030
[47] Martin Hirt, Ueli Maurer, and Vassilis Zikas. 2008. MPC vs. SFE: Unconditional and Computational Security. In *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology* (Melbourne, Australia) *(ASIACRYPT '08)*. Springer-Verlag, Berlin, Heidelberg, 1–18. https://doi.org/10.1007/978-3-540-89255-7_1
[48] Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. 2006. On combining privacy with guaranteed output delivery in secure multiparty computation. In *Proceedings of the 26th Annual International Conference on Advances in Cryptology* (Santa Barbara, California) *(CRYPTO'06)*. Springer-Verlag, Berlin, Heidelberg, 483–500. https://doi.org/10.1007/11818175_29
[49] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. 2020. Local Differential Privacy for Evolving Data. *Journal of Privacy and Confidentiality* 10, 1 (Jan. 2020). https://doi.org/10.29012/jpc.718
[50] Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. 2021. A new analysis of differential privacy's generalization guarantees (invited paper). In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Virtual, Italy) *(STOC 2021)*. Association for Computing Machinery, New York, NY, USA, 9. https://doi.org/10.1145/3406325.3465358
[51] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. Secure Multi-party Differential Privacy. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc., Montreal. https://proceedings.neurips.cc/paper_files/paper/2015/file/a01610228fe998f515a72dd730294d87-Paper.pdf
[52] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What Can We Learn Privately? *SIAM J. Comput.* 40, 3 (2011), 793–826. https://doi.org/10.1137/090756090
[53] Fumiyuki Kato, Yang Cao, and Masatoshi Yoshikawa. 2021. Preventing Manipulation Attack in Local Differential Privacy Using Verifiable Randomization Mechanism. In *Data and Applications Security and Privacy XXXV*, Ken Barker and Kambiz Ghazinour (Eds.). Springer International Publishing, Cham, 43–60.
[54] Marcel Keller, Emmanuela Orsini, and Peter Scholl. 2016. MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 830–842. https://doi.org/10.1145/2976749.2978357
[55] Dakshita Khurana, Hemanta K. Maji, and Amit Sahai. 2014. Black-Box Separations for Differentially Private Protocols. In *Advances in Cryptology – ASIACRYPT 2014*, Palash Sarkar and Tetsu Iwata (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 386–405.
[56] Joe Kilian. 1988. Founding cryptography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing* (Chicago, Illinois, USA) *(STOC '88)*. Association for Computing Machinery, New York, NY, USA, 20–31. https://doi.org/10.1145/62212.62215
[57] Yehuda Lindell. 2017. *How to Simulate It – A Tutorial on the Simulation Proof Technique*. Springer International Publishing, Cham, 277–346.
[58] Yehuda Lindell. 2021. Secure Multiparty Computation. *Commun. ACM* 64, 1 (dec 2021), 86–96. https://doi.org/10.1145/3387108
[59] Sahar Mazloom and Dov Gordon. 2018. Secure Computation with Differentially Private Access Patterns. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 490–507. https://doi.org/10.1145/3243734.3243851
[60] Andrew Mcgregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil P. Vadhan. 2011. The Limits of Two-Party Differential Privacy. *Electronic Colloquium on Computational Complexity* 18 (2011), 106.
[61] Fredrik Meisingseth, Christian Rechberger, and Fabian Schmid. 2025. Practical Two-party Computational Differential Privacy with Active Security. *Proceedings on Privacy Enhancing Technologies* (2025), to appear. https://eprint.iacr.org/2024/004
[62] Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. 2009. Computational Differential Privacy. In *Advances in Cryptology - CRYPTO 2009*, Shai Halevi (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 126–142.
[63] Jack Murtagh and Salil Vadhan. 2016. The Complexity of Computing the Optimal Composition of Differential Privacy. In *Theory of Cryptography*, Eyal Kushilevitz and Tal Malkin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 157–175.
[64] Arjun Narayan, Ariel Feldman, Antonis Papadimitriou, and Andreas Haeberlen. 2015. Verifiable Differential Privacy. In *EuroSys '15* (Bordeaux, France). Association for Computing Machinery, New York, NY, USA, Article 28, 14 pages. https://doi.org/10.1145/2741948.2741978
[65] Vibhor Rastogi and Suman Nath. 2010. Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data* (Indianapolis, Indiana, USA) *(SIGMOD '10)*. Association for Computing Machinery, New York, NY, USA, 735–746. https://doi.org/10.1145/1807167.1807247
[66] Thomas Steinke. 2022. Composition of Differential Privacy and Privacy Amplification by Subsampling. https://doi.org/10.48550/ARXIV.2210.00597
[67] Salil Vadhan. 2017. *The Complexity of Differential Privacy*. Springer International Publishing, Cham, 347–450. https://doi.org/10.1007/978-3-319-57048-8_7
[68] Salil Vadhan and Tianhao Wang. 2021. Concurrent Composition of Differential Privacy. In *Theory of Cryptography*, Kobbi Nissim and Brent Waters (Eds.). Springer International Publishing, Cham, 582–604.
[69] Sameer Wagh, Xi He, Ashwin Machanavajjhala, and Prateek Mittal. 2021. DP-Cryptography: Marrying Differential Privacy and Cryptography in Emerging Applications. *Commun. ACM* 64, 2 (jan 2021), 84–93. https://doi.org/10.1145/3418290
[70] Chengkun Wei, Ruijing Yu, Yuan Fan, Wenzhi Chen, and Tianhao Wang. 2023. Securely Sampling Discrete Gaussian Noise for Multi-Party Differential Privacy. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (, Copenhagen, Denmark,) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2262–2276. https://doi.org/10.1145/3576915.3616641
[71] Mingxun Zhou, Elaine Shi, T.-H. Hubert Chan, and Shir Maimon. 2023. A Theory of Composition for Differential Obliviousness. In *Advances in Cryptology – EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III* (Lyon, France). Springer-Verlag, Berlin, Heidelberg, 3–34. https://doi.org/10.1007/978-3-031-30620-4_1
[72] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. *Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity.* SIAM, Philadelphia, PA, 2468–2479. https://doi.org/10.1137/1.9781611975482.151

A The Ideal/real Paradigm, Standalone and UC Security

This section is a short introduction to the real/ideal-world paradigm of security and its most popular versions, the *standalone security model* and the *universal composability (UC) security model*. We will not be able to describe them in full formal detail, due to their complexity, and we refer to [12, 18] for details on the UC model and to [11, 35, 57] for details on the standalone model. For other brief introductions to the topic, we recommend [30, 58].

The core idea of the ideal/real paradigm of security is to define an *ideal world* that is secure by definition, i.e. which formulates what computations are supposed to be done and what it means to have that done securely. This includes, for example, specifying what types of information leakage are not to be seen as a violation of security. The security of the real protocol, defining the *real world*, is asserted by a simulation proof that the adversary cannot know if it is interacting with the ideal world or the real world. The thought is that if the adversary cannot tell if it is interacting with the real protocol or a version of the protocol that is secure by definition, then the protocol should be seen as secure also.

In the ideal world, there is an incorruptible third party called the *ideal functionality* which is given the inputs of all of the parties. This functionality performs the computation in question (potentially incorporating some well-defined allowed adversarial influence) and then gives the results to the players. Since the functionality cannot be corrupted, it thus defines what it means to be secure and what computational task should be achieved. When observing the protocol execution (either from the outside or as someone who takes part), it is potentially quite simple to tell apart the ideal world from the real world, for example by observing the number of messages sent. Therefore, the ideal world also must include a *simulator* (also called an *ideal-world adversary*) whose mission is to construct a view indistinguishable from the view of the real-world adversary. It must do this whilst only having access to the information available to it in the ideal world (essentially, the information given to it by the ideal functionality).

There are different ways to quantify the strength of such a simulation argument. One measure is the efficiency (say, in terms of runtime) of the simulator, since it describes how much work is needed to turn the allowed information leakage into the real information leakage. A faster simulator gives a stronger guarantee of security since then, intuitively, the real information leakage is more similar to the allowed one. Therefore, it is commonplace to require the simulator to be efficient in the sense of running in strict polynomial time, although this is not always the case for CDP using the ideal/real-world paradigm. Most notably, in MPRV [62], the simulators are allowed to be inefficient (computationally unbounded), for instance in the definition of SIM$^+$-CDP.

So, the core idea of the paradigm is to capture the notion of secure computation as that the ideal world (with parties, ideal functionality and simulator), in some sense, looks similar to the real world (with parties and adversary). This begs the question; who is the distinguisher? This is where is where the standalone and UC security models start diverging. In the standalone model, the distinguisher is essentially the adversary, meaning that the distinguisher itself takes part in the protocol. That is, the distinguisher tries to figure out which world it is in from the *inside*. The task of the simulator is to use only information available in the ideal world and generate an output distribution that is indistinguishable from the view[15] of the real-world adversary.

Definition A.1 (Standalone security, reformulation of Def. 4 in [11]). We say that a protocol π is a secure protocol for the functionality $\mathcal{F}$ if for all efficient adversaries $\mathcal{A}$, there exists an efficient simulator $\mathcal{S}$ (corrupting the same parties as $\mathcal{A}$) such that the joint output of the honest parties and $\mathcal{A}$ in the real world is computationally indistinguishable from the joint output of the honest parties and $\mathcal{S}$ in the ideal world, i.e. when the outputs distributions in the ideal and real worlds are computationally indistinguishable.

There are various different versions of the security definition, for instance varying the type of indistinguishability (like requiring the distributions to be identical or have negligible statistical distance). Other times the *correctness* requirement is changed (such as requiring that the outputs in the real and ideal worlds are identical or statistically close if there are no corruptions, as done in [35, 57]). The version used in MPRV [62] within the definition of SIM$^+$-CDP (Definition 5.3) has such an extra correctness requirement, as well as demanding efficient protocols and removing the efficiency requirement of the simulator.

Definition A.2 (Standalone security as in MPRV [62], Reformulated). We say that a protocol π is a secure protocol for the functionality $\mathcal{F}$ if it fulfills Definition A.1 with the following changes:

(1) π must be *efficiently computable*;
(2) π must have *perfect correctness*, that is, in an honest execution of π, its output distribution is identical to that of $\mathcal{F}$;
(3) the simulator is allowed to be inefficient.

In the standalone model, the security of the protocol is considered in isolation. That is, since the distinguisher is a part of the protocol execution, the protocol is studied under the assumption that the distinguisher does not run other protocols concurrently to the one being studied. Making such an assumption makes proving security technically much easier, for instance, it allows so-called rewinding techniques. The drawback of the model is precisely that it considers protocol security in isolation, opening up the possibility that a protocol thought to be secure loses all of its security properties when it is run in parallel to some other processes. Since it can be argued that such composition of protocols and processes is the rule rather than the exception in modern computer systems, it is highly desirable to be able to prove that a protocol remains secure also when other protocols are run in composition to it.

There are many ways to compose protocols and some of them are easier to deal with than others. For example does the usual formulations of the standalone model guarantee that security is preserved under *sequential composition*, i.e. as long as all protocols are run one after another. The most powerful type of composition results are those when the security of the protocol is preserved regardless of how the surrounding protocols are executed. This is called *universal composition* and the entire point of the UC (Universal Composability) security framework is that protocols proven within it remain secure under universal composition. In particular, if a protocol π realises the ideal functionality $\mathcal{F}$, then any other protocol that uses $\mathcal{F}$ as a subprocedure does not lose its security properties if $\mathcal{F}$ is replaced by a copy of π. In the UC framework, the distinguisher no longer is a part of the protocol execution per se, it is rather an external entity that observes and interacts with the system. This entity is called *the environment*. In more detail, it is an entity in both worlds that selects the initial inputs to all parties, interacts arbitrarily with the adversary and then, based on the outputs, tries to distinguish between the two worlds. In other words, the environment gets to play with one of the worlds and depending only on the input-output behaviour of this world it tries to determine if it is playing with real or the ideal world.

[15] In Definition A.1 it is the output rather than the view of the adversary that is considered. These two formulations are equivalent since the adversary is allowed to simply output its entire view as output.

Definition A.3 (UC security [12, 40]). We say that a protocol π UC-securely realises the ideal functionality $\mathcal{F}$ if for all PPT real-world adversaries $\mathcal{A}$ there exists a PPT simulator[16] $\mathcal{S}$ (corrupting the same parties as $\mathcal{A}$) such that for all PPT environments E, the statistical distance between E's output when interacting with the ideal world and that when interacting with the real world is negligible in the security parameter κ.

A.1 Complexity Assumptions

Now that we have seen the security definitions, we consider what assumptions one must make for them to be attainable. First of all, what kind of security one can prove of a protocol is directly dependent on the functionality one wants to realise. It is also dependent on the ideal functionalities one assumes are available to the parties, since such functionalities also define the communication channels present in the protocol execution. As a basis, the *plain model* assumes access to no other ideal functionality than authenticated channels, meaning that the parties can send messages to each other (point-to-point) and be sure who the messages come from and that it has not been tampered with but there are no guarantees that the contents of the messages have not been leaked. In the plain model, quite a few fundamental functionalities can be realised, such as *secure transfer* which is the same as authenticated transfer except that the contents of the messages are now hidden from an eavesdropper. There are however many important functionalities that cannot be realised in the plain model, unless one makes certain assumptions on the types of corruption that are being made (in particular, one has to assume an honest majority). In such cases, one has to leave the plain model and claim access to some other ideal functionality, i.e. one makes the assumption that there exists a protocol that realises that 'helping functionality'. Such assumptions are typically in the form of *complexity assumptions*, meaning that one assumes some given specific computation that the adversary (or environment) would have to do to mount a specific attack is computationally infeasible.

Such complexity assumptions have been very deeply studied, and they are commonly ordered after their relative strength, i.e. by proving that one assumption is stronger (or larger) than another in the sense that the first one implies the other but not the other way around. How various complexity assumptions relate to each other is quite well understood, and this is also true for what assumptions are needed for general-purpose MPC (i.e. where any PPT functionality can be realised) to be possible in various distribution and corruption models. In the main body, we mostly discuss three common complexity assumptions (listed from weakest to strongest):

- *The existence of one-way functions (OWFs).* A OWF is, intuitively, a function that can be computed efficiently but for which it is hard to find pre-images. That is, if one is given an evaluation of the function, no efficient adversary can predict the input which resulted in that evaluation with probability non-negligibly above that when purely guessing. For a definition and more detail, see [34].
- *The existence of a key-agreement (KA) protocol.* A KA protocol is, intuitively, a protocol in which two parties who at the beginning share no secret information with each other, send some sequence of messages to each other which results in them at the end both knowing a secret key but that this key is not known to an eaves-dropping adversary which sees only the transcript of the protocol.
- *The existence of an oblivious transfer (OT) protocol.* An OT protocol is, intuitively, a protocol between two parties, one of which has a number of information pieces (say, rows in a database) and the other wants to learn one of them. This should be done, however, without the party holding the data knowing which information the other one has learned. In that way, the sender is oblivious to the request of the receiver. The importance of OT is that it allows the construction of general-purpose MPC protocols when all but one of the parties are corrupted [56].

[16] Also called *ideal-world adversary.*

B Other DP Definitions and Distribution Models

We now very briefly discuss topics that to some extent concern DP outside of the central model or are dependent on computational relaxations of SDP. As noted in the introduction of this paper, the definitions discussed in this list here were excluded from our main body due to them, for one reason or another, not being directly relevant for studying definitions of CDP in multi-party settings. We include them here in order to offer a wider context to the definitions included in the main body.

B.1 Adaptive Query Choices, Interactive DP and DP under Continual Observation

One important aspect of DP in all models is that of *composition* of mechanisms, i.e. how the DP guarantees are affected by multiple DP mechanisms being run (sequentially or concurrently) on the same database [25, 50, 54, 63]. Whereas it is often studied in the simple non-adaptive setting where the DP mechanisms are chosen independently of the outcomes of the others, it is also studied in the adaptive setting, where an adversary can choose what mechanism execution to request dependent on the previous mechanism outputs. Such adaptive choices induce a notion of interaction even into the central model, turning such an interaction between the dataholder and the analyst into an asymmetric two-party protocol. Therefore, such interactive situations are at times studied explicitly as a distribution model separate from the central model, resulting in an explicit notion of *interactive DP*. This is, for instance, done recently in [45, 68], where interactive DP is defined precisely as in BNO-SDP except for that the DP guarantees are one-sided. Similarly, it is straight-forward to adapt the notion of interactive DP into a one-sided CDP definition.

Another commonly studied DP model in which interaction plays a crucial role is that of *DP under continual observation* and similar models [14, 24, 49]. As opposed to interactive DP above, here the mechanism is not adaptively chosen by the analyst but rather is the mechanism itself faced with the mission to release multiple outputs over time such that the overall mechanism is DP and at all times, the released output in that timestep has high utility. The core insight is that, since the outputs at different timesteps are highly correlated (as they concern the same dataset) one might be able to achieve a much higher utility than allowed by directly applying theorems from the literature on the composition of mechanisms.

B.2 Multiparty Protocols with DP Leakage

It was noted already in MPRV that SIM-CDP (as well as IND-CDP) allows partial intermediate results/information to be leaked as long as it is DP whereas SIM$^+$-CDP does not. That is, they allow there to be some non-negligible information leakage during the protocol (which is not allowed traditionally in MPC), as long as the leakage is CDP. This idea of having DP leakage during a protocol execution has also been combined with MPC protocol whose function evaluation is *not* DP, thus resulting simply in a controlled relaxation of the usual requirements in secure computation. This is done in order to improve efficiency and is particularly relevant when the function output, for use case specific reasons, is anyhow required to be exact. Some papers in this space where definitions are proposed are:

- [46] – As far as we are aware, this was the first definition of MPC with DP leakage. A DP definition is introduced (*output constrained DP*) and then it is specialised as *DP for record linkage (DPRL)*. This is simply IND-CDP with a new adjacency notion, namely one where only databases that evaluate a given function to the same value are considered adjacent.
- [59] – A more general definition of MPC with DP leakage is proposed which uses the standalone model of the ideal-real paradigm. It is essentially the same as the standard definition of secure two-party computation in the standalone model except that the simulator also learns an additional DP function of the input dataset.
- [40] – The MPC-with-leakage definition of [59] is adjusted to use UC-security instead of standalone security and also the leakage is allowed to occur before the corrupted party sends its inputs, thus relaxing the guarantee of input independence (see, for instance [58]).

B.3 CDP with Zero-knowledge Proofs

Just as there has been much work on combining DP with MPC, there has been work on combining it with the field of zero-knowledge proofs. One such line of work considers a notion of privacy called *zero-knowledge privacy*, which shares many similarities to DP, and another considers *verifiable DP*, where a DP mechanism output is given together with a proof that it has been faithfully generated.

Zero-knowledge Privacy. In [31], a stricter privacy notion related to SDP is proposed with the name *zero-knowledge privacy (ZKPr)*, which, very roughly, is made stronger than SDP by requiring that the view of the adversary can be approximated well by a simulator that only has access to some aggregate information about an adjacent database. A computational version of the notion is also proposed, let us call it CZKPr, by requiring that the simulator, adversary and function generating the aggregate information are PPT. The authors show that ZKPr is strictly stronger than SDP and that this remains true for CZKPr as long as the mechanism in question is efficient. We do not include it in the main body of this work because (C)ZKPr differs from SDP not only with regard to its computational perspective but also fundamentally with respect to the concept of adjacent databases and the information available to the simulator. Additionally, we are only aware of the definition being used in the central model.

Verifiable DP. Verifiable DP (VDP) is first proposed in 2015 [64] when a system called *VerDP* is proposed which answers a restricted set of DP queries whilst also proving that the mechanism output is both from a DP mechanism and consistent with the supposed database. Since the system uses cryptographic tools with computational security, the authors note that the DP guarantees in the end are computational (referring to MPRV) but the discussion about details here is limited due to that the focus of the paper is largely on the practical aspects of the given system and implementation. In 2023, another paper [6] re-introduced VDP and this time there is a substantial focus on defining VDP as a notion in itself and some fundamental impossibility-results are established, such as the impossibility of statistical VDP (for a DP system to be verifiable, it has to be computational). Additionally, the notion is also studied in a multi-party setting, with a first protocol being proposed and implemented.

B.4 CDP with Functional Encryption

DP has also been combined with the field of functional encryption [8], first in [1, 2] with statistical DP and then in [29] with CDP. In particular, in [29] IND-CDP is incorporated into a new definition of functional encryption. Then a general mechanism is proposed which satisfies the new definition and it is particularly studied for the case of linear queries.

C Proofs

C.1 Proofs Omitted in Section 4

C.1.1 $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP directly implies $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP for non-zero δ_κ. We now prove that the direct implication from SIM-CDP to IND-CDP with a negligible increase in the additive parameters also holds for non-zero δ_κ. That it holds for $\delta_\kappa = 0$ was shown already in MPRV and [67]. These proofs carry over directly to the more general setting under a mild condition on ε_κ. We state the implication only for central model mechanisms but it extends directly to the distributed setting since the result concerns only the formulation of output closeness, which is unchanged by the distribution model.

Proposition C.1. *Let $\delta_\kappa \in [0,1]$ be arbitrary and let $\varepsilon_\kappa \in O(\log(\kappa))$. Then any mechanism $\mathcal{M}$ that is $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP is also $(\varepsilon_\kappa, \delta_\kappa + \mathsf{negl}(\kappa))$-IND-CDP.*

Proof. Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be a mechanism that is $(\varepsilon_\kappa, \delta_\kappa)$-SIM-CDP. This implies that there exists an $(\varepsilon_\kappa, \delta_\kappa)$-SDP mechanism $\tilde{\mathcal{M}}$ such that the output distributions of $\mathcal{M}(D)$ and $\tilde{\mathcal{M}}(D)$ are computationally indistinguishable. That is, the output distributions of any PPT distinguisher when given $\tilde{\mathcal{M}}(D)$ and $\mathcal{M}(D)$, respectively, have a negligible statistical distance. This gives us, for any adjacent $D, D' \in \mathcal{D}$ and any PPT distinguisher T:

$$\begin{aligned}\mathbb{P}(T(\mathcal{M}(D)) = 1) &\leq \mathbb{P}(T(\tilde{\mathcal{M}}(D)) = 1) + \mathsf{negl}(\kappa)\\ &\leq e^{\varepsilon_\kappa}\mathbb{P}(T(\tilde{\mathcal{M}}(D')) = 1) + \delta_\kappa + \mathsf{negl}(\kappa)\\ &\leq e^{\varepsilon_\kappa}\left(\mathbb{P}(T(\mathcal{M}(D')) = 1) + \mathsf{negl}(\kappa)\right) + \delta_\kappa + \mathsf{negl}(\kappa)\\ &\leq e^{\varepsilon_\kappa}\mathbb{P}(T(\mathcal{M}(D')) = 1) + \delta_\kappa + \mathsf{negl}'(\kappa).\end{aligned}$$

The first and third inequalities follows from that $\mathcal{M}$ and $\tilde{\mathcal{M}}$ are computationally indistinguishable and the second from that $\tilde{\mathcal{M}}$ is $(\varepsilon_\kappa, \delta_\kappa)$-SDP. The final inequality follows from the assumption that any negligible function remains negligible when multiplied by e^{ε_κ}. In particular, $\mathsf{negl}(\kappa)$ is an arbitrary negligible function and $\mathsf{negl}'(\kappa) := e^{\varepsilon_\kappa} \cdot \mathsf{negl}(\kappa) + \mathsf{negl}(\kappa)$.

□

C.1.2 $(\varepsilon_\kappa, 0)$-IND-CDP is equivalent to $(\varepsilon_\kappa, 0)$-SDP. The reason to introduce a non-zero δ_κ parameter is that $(\varepsilon_\kappa, 0)$-IND-CDP is equivalent to $(\varepsilon_\kappa, 0)$-SDP and $(\varepsilon_\kappa, 0)$-BNO-SDP in the central and distributed models, respectively. We reiterate the argument here for completeness. We give the proposition and proof for the central model only as the extension to the distributed case is immediate.

Proposition C.2 (Reformulation of MPRV). *In the central model, $(\varepsilon_\kappa, 0)$-IND-CDP and $(\varepsilon_\kappa, 0)$-SDP are equivalent but $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP and $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-SDP are not. That is:*

1. *Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$, with $\mathrm{y} \in \mathcal{R}$ of polynomial size. If $\mathcal{M}$ is $(\varepsilon_\kappa, 0)$-IND-CDP then it is also $(\varepsilon_\kappa, 0)$-SDP, and vice versa.*
2. *There exist a mechanism $\mathcal{M}$ that is $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-IND-CDP but not $(\varepsilon_\kappa, \mathsf{negl}(\kappa))$-SDP.*

Proof. We start by proving the first statement in the proposition, namely that IND-CDP and SDP are equivalent when $\delta_\kappa = 0$. That all SDP mechanisms are also IND-CDP with unchanged parameters is immediate so what remains to show is the opposite direction. The argument below is a reformulation of a discussion in MPRV [62].

Part 1 (From MPRV):
Assume that $\mathcal{M}$ is $(\varepsilon_\kappa, 0)$-IND-CDP. Let $T^S(\eta)$, for some arbitrary $S \subset \mathcal{R}$, be the distinguisher that outputs $\mathbb{1}\{\eta \in S\}$. That $\mathcal{M}$ is $(\varepsilon_\kappa, 0)$-IND-CDP implies that, for all S such that T^S is PPT, we have

$$\mathbb{P}(T^S(\mathcal{M}(\mathbf{x})) = 1) \leq e^{\varepsilon_\kappa}\mathbb{P}(T^S(\mathcal{M}(\mathbf{x}')) = 1),$$

which implies

$$\mathbb{P}(\mathcal{M}(\mathbf{x}) \in S) \leq e^{\varepsilon_\kappa}\mathbb{P}(\mathcal{M}(\mathbf{x}') \in S).$$

Hence, for all sets $S \subset \mathcal{R}$ for which checking membership is efficient, the first part of the proposition holds. The assumption that the elements of $\mathcal{R}$ are of polynomial length implies that this is the case for all $S \subset \mathcal{R}$ (by use of, for instance, a binary search tree).

Part 2 (From [9]):
Consider the counter-example of the mechanism $\mathcal{M}$ that takes one bit x as input and if $x = 1$ it output a uniformly random 2κ-bit string and if $x = 0$ it outputs a pseudorandom 2κ-bit string by use of a pseudorandom generator (PRG) (see, for instance [34]). This mechanism is $(0, \mathsf{negl}(\kappa))$-IND-CDP but by the definition of a PRG, it is not $(0, \delta)$-SDP for any $\delta < 1 - \mathsf{negl}(\kappa)$, since an unbounded distinguisher can distinguish a PRG from a generator of truly random strings arbitrarily well. □

We remark that the restriction to output domains of polynomial-sized elements is very mild since the distinguisher is always assumed to be PPT, meaning that if the output of the mechanism is not of polynomial size, then the distinguisher cannot even read its whole input.

C.2 Proofs Omitted in Section 6

We now prove an ME-separation between SIM-CDP and SIM$^+$-CDP for the case where $\delta_\kappa = 0$. The idea is that SIM$^+$-CDP requires efficient protocols and perfect security, meaning that it cannot be satisfied for a task that cannot be solved in strict polynomial time. SIM-CDP on the other hand makes no such requirements, meaning that it can be fulfilled for inefficient protocols. That is, the only thing required is to find a task that can be solved in, say, exponential time but not in polynomial time. We find such a task in the shape of computing the XOR gate to within a given probability of failure, which is suitable because it is equivalent to sampling a Bernoulli trial with a given parameter, and this parameter can easily be chosen such that the sampling can be done exactly only in super-polynomial time.

Proposition C.3 (ME-separating SIM-CDP and SIM$^+$-CDP). *There exist $\varepsilon_\kappa > 0$ for which $(\varepsilon_\kappa, 0)$-SIM-CDP $\underset{ME}{\not\Longrightarrow}$ $(\varepsilon_\kappa, 0)$-SIM$^+$-CDP.*

Proof. We consider the two-party case and passive corruptions. Let $\varepsilon_\kappa = ln(2^{2^\kappa} - 1)$, $D = \{0, 1\}^2$, set the utility function to $u((D_1, D_2), (\eta, y)) := \mathbb{1}\{y = D_1 \oplus D_2\}$ and $\alpha = 1 - 2^{-2^\kappa}$. That is, the task is to have party 2 output the XOR of the inputs of both the parties' inputs and to be incorrect with a probability of at most 2^{-2^κ}.

To see that there is no $(\varepsilon_\kappa, 0)$-SIM$^+$-CDP protocol solving the task, assume towards a contradiction that there is such a protocol π'. Since SIM$^+$-CDP demands perfect correctness, we know that in an honest execution, the output distributions of π' and the ideal functionality it realises, $\mathcal{M}'$, have identical output distributions. Since π' runs in strict polynomial time, this implies that there is a strict PPT mechanism $\mathcal{M}'$ that is $(ln(2^{2^\kappa} - 1), 0)$-SDP and outputs a bit that is $D_1 \oplus D_2$ with probability at least α and its negation with probability at most $1 - \alpha$. It is easy to see that no binary $(\varepsilon_\kappa, 0)$-SDP mechanism can have accuracy above $e^{\varepsilon_\kappa}/(e^{\varepsilon_\kappa} + 1)$, which means that the task above is the same as outputting the correct function evaluation with probability $1 - 2^{-2^\kappa}$. Sampling a Bernoulli trial with such a parameter is impossible in strict polynomial time (as it requires exponentially many fair coins) and thus we have reached the contradiction.

We now give a protocol π that solves the task and simultaneously is $(\varepsilon_\kappa, 0)$-SIM-CDP. The key here is that π need not be efficient, since there is no such requirement in SIM-CDP. Let π simply be that party 1 runs randomised response on its input with parameter ε_κ. That is, first P_1 samples a Bernoulli trial with parameter $1 - 2^{-2^\kappa}$ using 2^κ uniform coins, which can trivially be done in exponential time. Call the sample outcome b. Then P_1 sends $c \leftarrow D_1 \oplus b$ to P_2 and outputs $\perp$. Then P_2 outputs $D_2 \oplus c$. This protocol obviously solves the task and is $(\varepsilon_\kappa, 0)$-SIM-CDP via the simulator that samples b as P_1 does and then outputs $\perp$ to P_1 and $D_1 \oplus D_2 \oplus b$ to P_2. □

Note that the task in the proof above is quite contrived and is chosen as to simplify the proof rather than being practically interesting or general. In fact, any task that has an optimal SDP mechanism that can be run in exponential but not polynomial time suffices for the proof idea, and thus is it probable that the proposition extends into quite general parameter regimes. Further (as noted in MPRV), if one does restrict the SIM-CDP protocol to be efficient, it seems likely that there should be ME-separations between the two notions, since SIM-CDP allows the simulator (there also the mechanism) to have access to the inputs of both parties, whereas in SIM$^+$-CDP the simulator only has access to the outputs of the ideal functionality and from them it has to construct the adversarial view, which seems much more restrictive.