Separating Computational and Statistical Differential Privacy in the Client-Server Model

Mark Bun(✉), Yi-Hsiu Chen, and Salil Vadhan

John A. Paulson School of Engineering and Applied Sciences, Center for Research on Computation and Society, Harvard University, Cambridge, MA, USA
{mbun,yhchen,salil}@seas.harvard.edu

Abstract. Differential privacy is a mathematical definition of privacy for statistical data analysis. It guarantees that any (possibly adversarial) data analyst is unable to learn too much information that is specific to an individual. Mironov et al. (CRYPTO 2009) proposed several computational relaxations of differential privacy (CDP), which relax this guarantee to hold only against computationally bounded adversaries. Their work and subsequent work showed that CDP can yield substantial accuracy improvements in various multiparty privacy problems. However, these works left open whether such improvements are possible in the traditional client-server model of data analysis. In fact, Groce, Katz and Yerukhimovich (TCC 2011) showed that, in this setting, it is impossible to take advantage of CDP for many natural statistical tasks.

Our main result shows that, assuming the existence of sub-exponentially secure one-way functions and 2-message witness indistinguishable proofs (zaps) for **NP**, that there is in fact a computational task in the client-server model that can be efficiently performed with CDP, but is infeasible to perform with information-theoretic differential privacy.

1 Introduction

Differential privacy is a formal mathematical definition of privacy for the analysis of statistical datasets. It promises that a data analyst (treated as an adversary) cannot learn too much individual-level information from the outcome of an analysis. The traditional definition of differential privacy makes this promise information-theoretically: Even a computationally unbounded adversary is limited in the amount of information she can learn that is specific to an individual.

M. Bun—Supported by an NDSEG Fellowship and NSF grant CNS-1237235. Part of this work was done while the author was visiting Yale University.

Y.-H. Chen—Supported by NSF grant CCF-1420938.

S. Vadhan—Supported by NSF grant CNS-1237235 and a Simons Investigator Award. Part of this work was done while the author was visiting the Shing-Tung Yau Center and the Department of Applied Mathematics at National Chiao-Tung University in Hsinchu, Taiwan.

M. Hirt and A. Smith (Eds.): TCC 2016-B, Part I, LNCS 9985, pp. 607–634, 2016.
DOI: 10.1007/978-3-662-53641-4_23

On one hand, there are now numerous techniques that actually achieve this strong guarantee of privacy for a rich body of computational tasks. On the other hand, the information-theoretic definition of differential privacy does not itself permit the use of basic cryptographic primitives that naturally arise in the practice of differential privacy (such as the use of cryptographically secure pseudorandom generators in place of perfect randomness). More importantly, computationally secure relaxations of differential privacy open the door to designing improved mechanisms: ones that either achieve better *utility* (accuracy) or *computational efficiency* over their information-theoretically secure counterparts.

Motivated by these observations, and building on ideas suggested in [BNO08], Mironov et al. [MPRV09] proposed several definitions of *computational differential privacy* (CDP). All of these definitions formalize what it means for the output of a mechanism to "look" differentially private to a computationally bounded (i.e. probabilistic polynomial-time) adversary. The sequence of works [DKM+06, BNO08, MPRV09] introduced a paradigm that enables *two or more parties* to take advantage of CDP, either to achieve better utility or reduced round complexity, when computing a joint function of their private inputs: The parties use a secure multi-party computation protocol to simulate having a trusted third party perform a differentially private computation on the union of their inputs. Subsequent work [MMP+10] showed that such a CDP protocol for approximating the Hamming distance between two private bit vectors is in fact more accurate than any (information-theoretically secure) differentially private protocol for the same task. A number of works [CSS12, GMPS13, HOZ13, KMS14, GKM+16] have since sought to characterize the extent to which CDP yields accuracy improvements for two-party privacy problems.

Despite the success of CDP in the design of improved algorithms in the multiparty setting, much less is known about what can be achieved in the traditional client-server model, in which a trusted curator holds all of the sensitive data and mediates access to it. Beyond just the absence of any techniques for taking advantage of CDP in this setting, results of Groce, Katz, and Yerukhimovich [GKY11] (discussed in more detail below) show that CDP yields no additional power in the client-server model for many basic statistical tasks. An additional barrier stems from the fact that all known lower bounds against computationally efficient differentially private algorithms [DNR+09, UV11, Ull13, BZ14, BZ16] in the client-server model are proved by exhibiting computationally efficient adversaries. Thus, these lower bounds rule out the existence of CDP mechanisms just as well as they rule out differentially private ones.

In this work, we give the first example of a computational problem in the client-server model which can be solved in polynomial-time with CDP, but (under plausible assumptions) is computationally infeasible to solve with (information-theoretic) differential privacy. Our problem is specified by an efficiently computable *utility function* u, which takes as input a dataset $D \in \mathcal{X}^n$ and an answer $r \in \mathcal{R}$, and outputs 1 if the answer r is "good" for the dataset D, and 0 otherwise.

Theorem 1 (Main (Informal)). *Assuming the existence of sub-exponentially secure one-way functions and "exponentially extractable" 2-message witness indistinguishable proofs (zaps) for* **NP***, there exists an efficiently computable utility function* $u : \mathcal{X}^n \times \mathcal{R} \to \{0,1\}$ *such that*

1. *There exists a polynomial time CDP mechanism* M^{CDP} *such that for every dataset* $D \in \mathcal{X}^n$*, we have* $\Pr[u(D, M^{\mathrm{CDP}}(D)) = 1] \geq 2/3$.
2. *There exists a computationally unbounded differentially private mechanism* M^{unb} *such that* $\Pr[u(D, M^{\mathrm{unb}}(D)) = 1] \geq 2/3$.
3. *For every polynomial time differentially private* M*, there exists a dataset* $D \in \mathcal{X}^n$*, such that* $\Pr[u(D, M(D)) = 1] \leq 1/3$.

Note that the theorem provides a task where achieving differential privacy is infeasible – not impossible. This is inherent because the CDP mechanism we exhibit (for item 1) satisfies a simulation-based form of CDP ("SIM-CDP"), which implies the existence of a (possibly inefficient) differentially private mechanism, provided the utility function u is efficiently computable as we require. It remains an intriguing open problem to exhibit a task that can be achieved with a weaker indistinguishably-based notion of CDP ("IND-CDP") but is *impossible* to achieve (even inefficiently) with differential privacy. Such a task would also separate IND-CDP and SIM-CDP, which is an interesting open problem in its own right.

Circumventing the impossibility results of [GKY11]. Groce et al. showed that in many natural circumstances, computational differential privacy cannot yield any additional power over differential privacy in the client-server model. In particular, they showed two impossibility results:

1. If a CDP mechanism accesses a one-way function (or more generally, any cryptographic primitive that can be instantiated with a random function) in a black-box way, then it can be simulated just as well (in terms of both utility and computationally efficiency) by a differentially private mechanism.
2. If the output of a CDP mechanism is in $\mathbb{R}^d$ (for some constant d) and its utility is measured via an L_p-norm, then the mechanism can be simulated by a differentially private one, again without significant loss of utility or efficiency.

(In Sect. 4, we revisit the techniques [GKY11] to strengthen the second result in some circumstances. In general, we show that when error is measured in any metric with doubling dimension $O(\log k)$, CDP cannot improve utility by more than a constant factor. Specifically, respect to L_p-error, CDP cannot do much better than DP mechanisms even when d is logarithmic in the security parameter.)

We get around both of these impossibility results by (1) making non-black-box use of one-way functions via the machinery of zap proofs and (2) relying on a utility function that is far from the form in which the second result of [GKY11] applies. Indeed, our utility function is cryptographic and unnatural

from a data analysis point view. Roughly speaking, it asks whether the answer r is a valid zap proof of the statement "there exists a row of the dataset D that is a valid message-signature pair" for a secure digital signature scheme. It remains an intriguing problem for future work whether a separation can be obtained from a more natural task (such as answering a polynomial number of counting queries with differential privacy).

Our Construction and Techniques. Our construction is based on the existence of two cryptographic primitives: an existentially unforgeable digital signature scheme $(\mathrm{Gen}, \mathrm{Sign}, \mathrm{Ver})$, and a 2-message witness indistinguishable proof system (zap) (P, V) for **NP**. We make use of complexity leveraging [CGGM00] and thus require a complexity gap between the two primitives: namely, a sub-exponential time algorithm should be able to break the security of the zap proof system, but should not be able to forge a valid message-signature pair for the digital signature scheme.

We now describe (eliding technical complications) the computational task which allows us to separate computational and information-theoretic differential privacy in the client-server model. Inspired by prior differential privacy lower bounds [DNR+09, UV11], we consider a dataset D that consists of many valid message-signature pairs $(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ for the digital signature scheme. We say that a mechanism M gives a useful answer on D, i.e. the utility function $u(D, M(D))$ evaluates to 1, if it produces a proof π in the zap proof system that there exists a message-signature pair (m, σ) for which $\mathrm{Ver}(m, \sigma) = 1$.

First, let us see how the above task can be performed *inefficiently* with differential privacy. Consider the mechanism M^{unb} that first confirms (in a standard differentially private way) that its input dataset indeed contains "many" valid message-signature pairs. Then M^{unb} uses its unbounded computational resources to forge a canonical valid message-signature pair (m, σ) and uses the zap prover on witness (m, σ) to produce a proof π. Since the choice of the forged pair does not depend on the input dataset at all, the procedure as a whole is differentially private.

Now let us see how a CDP mechanism can perform the same task efficiently. Our mechanism M^{CDP} again first checks that it possesses many valid message-signature pairs, but this time it simply outputs a proof π using an arbitrary valid pair $(m_i, \sigma_i) \in D$ as its witness. Since the proof system is witness indistinguishable, a computationally bounded observer cannot distinguish π from the canonical proof output by the differentially private mechanism M^{unb}. Thus, the mechanism M^{CDP} is in fact CDP in the strongest (simulation-based) sense.

Despite the existence of the inefficient differentially private mechanism M^{unb}, we show that the existence of an efficient mechanism M for this task would violate the sub-exponential security of the digital signature scheme. Suppose there were such a mechanism M. Now consider a sub-exponential time adversary A that completely breaks the security of the zap proof system, in the sense that given a valid proof π, it is always able to recover a corresponding witness (m, σ). Since M is differentially private, the (m, σ) extracted by A cannot be in the dataset D given to M. Thus, (m, σ) constitutes a forgery of a valid

message-signature pair, and hence the composed algorithm $A \circ M$ violates the security of the signature scheme.

2 Preliminaries

2.1 (Computational) Differential Privacy

We first set notations that will be used throughout this paper, and recall the notions of (ε, δ)-differential privacy and computational differential privacy. The abbreviation "PPT" stands for "probabilistic polynomial-time Turing machine."

Security Parameter k. Let $k \in \mathbb{N}$ denote a security parameter. In this work, datasets, privacy-preserving mechanisms, and privacy parameters ε, δ will all be sequences parameterized in terms of k. Adversaries will also have their computational power parameterized by k; in particular, efficient adversaries have circuit size polynomial in k. A function is said to be *negligible* if it vanishes faster than any inverse polynomial in k.

Dataset D. A dataset D is an ordered tuple of n elements from some data universe $\mathcal{X}$. Two datasets D, D' are said to be *adjacent* (written $D \sim D'$) if they differ in at most one row. We use $\{D_k\}_{k\in\mathbb{N}}$ to denote a sequence of datasets, each over a data universe $\mathcal{X}_k$, with sizes growing with the parameter k. The size in bits of a dataset D_k, and in particular the number of rows n, will always be $\mathrm{poly}(k)$.

Mechanism M. A mechanism $M : \mathcal{X}^* \to \mathcal{R}$ is a randomized function taking a dataset $D \in \mathcal{X}^*$ to an output in a range space $\mathcal{R}$. We will be especially interested in ensembles of *efficient* mechanisms $\{M_k\}_{k\in\mathbb{N}}$ where each $M_k : \mathcal{X}_k^* \to \mathcal{R}_k$, when run on an input dataset $D \in \mathcal{X}_k^n$, runs in time $\mathrm{poly}(k, n)$.

Adversary A. Given an ensemble of mechanisms $\{M_k\}_{k\in\mathbb{N}}$ with $M_k : X_k^* \to \mathcal{R}_k$, we model an adversary $\{A_k\}_{k\in\mathbb{N}}$ as a sequence of polynomial-size circuits $A_k : \mathcal{R}_k \to \{0, 1\}$. Equivalently, $\{A_k\}_{k\in\mathbb{N}}$ can be thought of as a probabilistic polynomial time Turing machine with non-uniform advice.

Definition 1 (Differential Privacy [DMNS06, DKM+06]**).** *A mechanism M is (ε, δ)-differentially private if for all adjacent datasets $D \sim D'$ and every set $S \subseteq \mathrm{Range}(M)$,*

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S] + \delta$$

Equivalently, for all adjacent datasets $D \sim D'$ and every (computationally unbounded) algorithm A, we have

$$\Pr[A(M(D)) = 1] \leq e^{\varepsilon} \Pr[A(M(D')) = 1] + \delta \tag{1}$$

For consistency with the definition of SIM-CDP, we also make the following definitions for sequences of mechanisms:

– *An ensemble of mechanisms* $\{M_k\}_{k\in\mathbb{N}}$ *is* ε_k*-DP if for all* k*,* M_k *is* $(\varepsilon_k, \mathrm{negl}(k))$*-differentially private.*
– *An ensemble of mechanisms* $\{M_k\}_{k\in\mathbb{N}}$ *is* ε_k*-PURE-DP if for all* k*,* M_k *is* $(\varepsilon_k, 0)$*-differentially private.*

The above definitions are completely information-theoretic. Several computational relaxations of this definition are proposed by Mironov et al. [MPRV09]. The first "indistinguishability-based" definition, denoted IND-CDP, relaxes Condition (1) to hold against computationally-bounded adversaries:

Definition 2 (IND-CDP). *A sequence of mechanisms* $\{M_k\}_{k\in\mathbb{N}}$ *is* ε_k*-IND-CDP if there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all sequences of pairs of* $\mathrm{poly}(k)$*-size adjacent datasets* $\{(D_k, D'_k)\}_{k\in\mathbb{N}}$*, and all non-uniform polynomial time adversaries* A*,*

$$\Pr[A(M_k(D_k)) = 1] \leq e^{\varepsilon_k} \Pr[A(M_k(D'_k)) = 1] + \mathrm{negl}(k).$$

Mironov et al. [MPRV09] also proposed a stronger "simulation-based" definition of computational differential privacy. A mechanism is said to be ε-SIM-CDP if its output is computationally indistinguishable from that of an ε-differentially private mechanism:

Definition 3 (SIM-CDP). *A sequence of mechanisms* $\{M_k\}_{k\in\mathbb{N}}$ *is* ε_k*-SIM-CDP if there exists a negligible function* $\mathrm{negl}(\cdot)$ *and a family of mechanisms* $\{M'_k\}_{k\in\mathbb{N}}$ *that is* ε_k*-differentially private such that for all* $\mathrm{poly}(k)$*-size datasets* D*, and all non-uniform polynomial time adversaries* A*,*

$$|\Pr[A(M_k(D)) = 1] - \Pr[A(M'_k(D)) = 1]| \leq \mathrm{negl}(k).$$

If M'_k *is in fact* ε_k*-pure differentially private, then we say that* $\{M_k\}_{k\in\mathbb{N}}$ *is* ε_k*-PURE-SIM-CDP.*

Writing $A \preceq B$ to denote that a mechanism satisfying definition A also satisfies definition B (that is, A is a stricter privacy definition than B). We have the following relationships between the various notions of (computational) differential privacy:

$$\text{DP} \preceq \text{SIM-CDP} \preceq \text{IND-CDP}.$$

We will state and prove our separation between CDP and differential privacy for the simulation-based definition SIM-CDP. Since SIM-CDP is a stronger privacy notion than IND-CDP, this implies a separation between IND-CDP and differential privacy as well.

2.2 Utility

We describe an abstract notion of what it means for a mechanism to "succeed" at performing a computational task. We define a computational task implicitly

in terms of an efficiently computable *utility function*, which takes as input a dataset $D \in \mathcal{X}^*$ and an answer $r \in \mathcal{R}$ and outputs a score describing how well r solves a given problem on instance D. For our purposes, it suffices to consider binary-valued utility functions u, which output 1 iff the answer r is "good" for the dataset D.

Definition 4 (Utility). *A* utility function *is an efficiently computable (deterministic) function $u : \mathcal{X}^* \times \mathcal{R} \to \{0,1\}$. A mechanism M is* α-useful *for a utility function $u : \mathcal{X}^* \times \mathcal{R} \to \{0,1\}$ if for all datasets D,*

$$\Pr_{r \leftarrow M(D)}[u(D,r) = 1] \geq \alpha.$$

Restricting our attention to efficiently computable utility functions is necessary to rule out pathological separations between computational and statistical notions of differential privacy. For instance, let $\{G_k\}_{k \in \mathbb{N}}$ be a pseudorandom generator with $G_k : \{0,1\}^k \to \{0,1\}^{2k}$, and consider the (hard-to-compute) function $u(0,r) = 1$ iff r is in the image of G_k, and $u(1,r) = 1$ iff r is *not* in the image of G_k. Then the mechanism $M(b)$ that samples from G_k if $b = 0$ and samples a random string if $b = 1$ is useful with overwhelming probability. Moreover, M is computationally indistinguishable from the mechanism that always outputs a random string, and hence SIM-CDP. On the other hand, the supports of $u(0,\cdot)$ and $u(1,\cdot)$ are disjoint, so no differentially private mechanism can achieve high utility with respect to u.

2.3 Zaps (2-Message WI Proofs)

The first cryptographic tool we need in our construction is 2-message witness indistinguishable proofs for **NP** ("zaps") [FS90,DN07] in the plain model (with no common reference string). Consider a language $L \in \mathbf{NP}$. A *witness relation* for L is a polynomial-time decidable binary relation $R_L = \{(x,w)\}$ such that $|w| \leq \text{poly}(|x|)$ whenever $(x,w) \in R_L$, and

$$x \in L \iff \exists w \text{ s.t. } (x,w) \in R_L.$$

Definition 5 (Zap). *Let $R_L = \{(x,w)\}$ be a witness-relation corresponding to a language $L \in \mathbf{NP}$. A* zap proof system *for R_L consists of a pair of algorithms (P,V) where:*

- *In the first round, the* verifier *sends a message $\rho \leftarrow \{0,1\}^{\ell(k,|x|)}$ ("public coins"), where $\ell(\cdot,\cdot)$ is a fixed polynomial.*
- *In the second round, the* prover *runs a PPT P that takes as input a pair (x,w) and verifier's first message ρ and outputs a proof π.*
- *The verifier runs an efficient, deterministic algorithm V that takes as input an instance x, a first-round message ρ, and proof π, and outputs a bit in $\{0,1\}$.*

The security requirements of the proof system are:

1. Perfect completeness. *An honest prover who possesses a valid witness can always convince an honest verifier. Formally, for all* $x \in \{0,1\}^{\mathrm{poly}(k)}$, $(x,w) \in R_L$, *and* $\rho \in \{0,1\}^{\ell(k,|x|)}$,

$$\Pr_{\pi \leftarrow P(1^k,x,w,\rho)}[V(1^k,x,\rho,\pi)=1]=1.$$

2. Statistical soundness. *With overwhelming probability over the choice of* ρ, *it is impossible to convince an honest verifier of the validity of a false statement. Formally, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all sufficiently large* k *and* $t=\mathrm{poly}(k)$, *we have*

$$\Pr_{\rho \leftarrow \{0,1\}^{\ell(k,t)}}[\exists x \notin L \cap \{0,1\}^t, \pi \in \{0,1\}^* : V(1^k,x,\rho,\pi)=1] \le \mathrm{negl}(k).$$

3. Witness indistinguishability. *For every sequence* $\{x_k\}_{k\in\mathbb{N}}$ *with* $|x_k| = \mathrm{poly}(k)$, *every two sequences* $\{w_k^1\}_{k\in\mathbb{N}}$, $\{w_k^2\}_{k\in\mathbb{N}}$ *such that* $(x_k,w_k^1),(x_k,w_k^2) \in R_L$, *and every choice of the verifier's first message* ρ, *we have*

$$\{P(1^k,x_k,w_k^1,\rho)\}_{k\in\mathbb{N}} \overset{c}{\approx} \{P(1^k,x_k,w_k^2,\rho)\}_{k\in\mathbb{N}}.$$

Namely, for every such pair of sequences, there exists a negligible function $\mathrm{negl}(\cdot)$ *such that for all polynomial-time adversaries* A *and all sufficiently large* k, *we have*

$$|\Pr[A(1^k,P(1^k,x_k,w_k^1,\rho))=1] - \Pr[A(1^k,P(1^k,x_k,w_k^2,\rho))=1]| \le \mathrm{negl}(k).$$

In our construction, we will need more fine-grained control over the security of our zap proof system. In particular, we need the proof system to be *extractable* by an adversary running in time $2^{O(k)}$, in that such an adversary can always reverse-engineer a valid proof π to find a witness w such that $(x,w) \in R_L$. It is important to note that we require the running time of the adversary to be exponential in the security parameter k, but otherwise independent of the statement size $|x|$.

Definition 6 (Extractable Zap). *The algorithm triple* (P,V,E) *is an* extractable zap *proof system if* (P,V) *is a zap proof system and there exists an algorithm* E *running in time* $2^{O(k)}$ *with the following property:*

4. (Exponential Statistical) Extractability. *There exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all* $x \in \{0,1\}^{\mathrm{poly}(k)}$*:*

$$\Pr_{\rho \leftarrow \{0,1\}^{\ell(k,|x|)}}[\exists \pi \in \{0,1\}^*, w \in E(1^k,x,\rho,\pi) : (x,w) \notin R_L \ \wedge \ V(1^k,x,\rho,\pi)=1] \le \mathrm{negl}(k).$$

While we do not know whether extractability is a generic property of zaps, it is preserved under Dwork and Naor's reduction to NIZKs in the common random string model. Namely, if we plug an extractable NIZK into Dwork and Naor's construction, we obtain an extractable zap.

Theorem 2. *Every language in* **NP** *has an extractable zap proof system* (P, V, E)*, as defined in Definition 6, if there exists non-interactive zero-knowledge proofs of knowledge for* **NP** [DN07].

For completeness, we sketch Dwork and Naor's construction in Appendix B and argue its extractability.

2.4 Digital Signatures

The other ingredient we need in our construction is sub-exponentially strongly unforgeable digital signature schemes. Here "strong unforgeability" [ADR02] means that the adversary in the existential unforgeability game is allowed to forge a signature for a message it has queried before, as long as the signature is different than the one it received.

Definition 7 (Sub-exponentially Strongly Unforgeable Digital Signature Scheme). *Let* $c \in (0,1)$ *be a constant. A* c-strongly unforgeable digital signature *is a triple of PPT algorithms* $(\mathrm{Gen}, \mathrm{Sign}, \mathrm{Ver})$ *where*

- $(sk, vk) \leftarrow \mathrm{Gen}(1^k)$*: The* generation algorithm *takes as input a security parameter* k *and generates a secret key and a verification key.*
- $\sigma \leftarrow \mathrm{Sign}(sk, m)$*: The* signing algorithm *signs a message* $m \in \{0,1\}^*$ *to produce a signature* $\sigma \in \{0,1\}^*$*.*
- $b \leftarrow \mathrm{Ver}(vk, m, \sigma)$*: The (deterministic)* verification algorithm *outputs a bit to indicate whether the signature* σ *is a valid signature of* m*.*

The algorithms have the following properties:

1. Correctness. *For every message* $m \in \{0,1\}^*$*,*

$$\Pr_{\substack{(sk,vk)\leftarrow\mathrm{Gen}(1^k)\\ \sigma\leftarrow\mathrm{Sign}(sk,m)}}[\mathrm{Ver}(vk, m, \sigma) = 1] = 1.$$

2. Existential unforgeability. *There exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all adversaries* A *running in time* 2^{k^c}*,*

$$\Pr_{\substack{(sk,vk)\leftarrow\mathrm{Gen}(1^k)\\ (m,\sigma)\leftarrow A^{\mathrm{Sign}(sk,\cdot)}(vk)}}[\mathrm{Ver}(m, \sigma) = 1 \ \textit{and}\ (m, \sigma) \notin Q] < \mathrm{negl}(k)$$

where Q *is the set of messages-signature pairs obtained through* A*'s use of the signing oracle.*

Theorem 3. *If sub-exponentially secure one-way functions exist, then there is a constant* $c \in (0,1)$ *such that a* c*-strongly unforgeable digital signature scheme exists.*

The reduction from a one-way function to digital signature [NY89, Rom90, KK05, Gol04] can be applied when both schemes are secure against sub-exponential time adversaries.

3 Separating CDP and Differential Privacy

In this section, we define a computational problem in the client-server model that can be efficiently solved with CDP, but not with statistical differential privacy. That is, we define a utility function u for which there exists a CDP mechanism achieving high utility. On the other hand, any efficient differentially private algorithm can only have negligible utility.

Theorem 4 (Main). *Assume the existence of sub-exponentially secure one-way functions and extractable zaps for* **NP**. *Then there exists a sequence of data universes* $\{\mathcal{X}_k\}_{k\in\mathbb{N}}$, *range spaces* $\{\mathcal{R}_k\}_{k\in\mathbb{N}}$ *and an (efficiently computable) utility function* $u_k : \mathcal{X}_k^* \times \mathcal{R}_k \to \{0,1\}$ *such that*

1. *There exists a polynomial* p *such that for any* $\varepsilon_k, \beta_k > 0$ *there exists a polynomial-time* ε_k-PURE-SIM-CDP *mechanism* $\{M_k^{\mathrm{CDP}}\}_{k\in\mathbb{N}}$ *and an (inefficient)* ε_k-PURE-DP *mechanism* $\{M_k^{\mathrm{unb}}\}_{k\in\mathbb{N}}$ *such that for every* $n \geq p(k, 1/\varepsilon_k, \log(1/\beta_k))$ *and dataset* $D \in \mathcal{X}_k^n$, *we have*

$$\Pr[u_k(D, M^{\mathrm{CDP}}(D)) = 1] \geq 1 - \beta_k \quad \textit{and} \quad \Pr[u_k(D, M^{\mathrm{unb}}(D)) = 1] \geq 1 - \beta_k$$

2. *For every* $\varepsilon_k \leq O(\log k)$, $\alpha_k = 1/\operatorname{poly}(k)$, $n = \operatorname{poly}(k)$, *and efficient* $(\varepsilon_k, \delta = 1/n^2)$*-differentially private mechanism* $\{M'_k\}_{k\in\mathbb{N}}$, *there exists a dataset* $D \in \mathcal{X}_k^n$ *such that*

$$\Pr[u(D, M'(D)) = 1] \leq \alpha_k \quad \textit{for sufficient large } k.$$

Remark 1 We can only hope to separate SIM-CDP and differential privacy by designing a task that is *infeasible* with differential privacy but not *impossible*. By the definition of (PURE-)SIM-CDP for a mechanism $\{M_k\}_{k\in\mathbb{N}}$, there exists an ε_k-(PURE-)DP mechanism $\{M'_k\}_{k\in\mathbb{N}}$ that is computationally indistinguishable from $\{M_k\}_{k\in\mathbb{N}}$. But if for every differentially private $\{M'_k\}_{k\in\mathbb{N}}$, there were a dataset $D_k \in \mathcal{X}_k^n$ such that $\Pr[u_k(D_k, M'_k(D_k)) = 1] \leq \Pr[u_k(D_k, M_k(D_k)) = 1] - 1/\operatorname{poly}(k)$, then the utility function $u_k(D_k, \cdot)$ would itself serve as a distinguisher between $\{M'_k\}_{k\in\mathbb{N}}$ and $\{M_k\}_{k\in\mathbb{N}}$.

3.1 Construction

Let $(\mathrm{Gen}, \mathrm{Sign}, \mathrm{Ver})$ be a c-strongly unforgeable secure digital signature scheme with parameter $c > 0$ as in Definition 7. After fixing c, we define for each $k \in \mathbb{N}$ a reduced security parameter $k_c = k^{c/2}$. We will use k_c as the security parameter for an extractable zap proof system (P, V, E). Since k and k_c are polynomially related, a negligible function in k is negligible in k_c and vice versa.

Given a security parameter $k \in \mathbb{N}$, define the following sets of bit strings:

Verification Key Space: $\mathcal{K}_k = \{0,1\}^{\ell_1}$ where $\ell_1 = |vk|$ for $(sk, vk) \leftarrow \mathrm{Gen}(1^k)$,
Message Space: $\mathcal{M}_k = \{0,1\}^k$,

Signature Space: $\mathcal{S}_k = \{0,1\}^{\ell_2}$ where $\ell_2 = |\sigma|$ for $\sigma \leftarrow \text{Sign}(sk, m)$ with $m \in \mathcal{M}_k$,
Public Coins Space: $\mathcal{P}_k = \{0,1\}^{\ell_3}$ where $\ell_3 = \text{poly}(\ell_1)$ is the length of first-round zap messages used to prove statements from $\mathcal{K}_k$ under security parameter k_c,
Data Universe: $\mathcal{X}_k = \mathcal{K}_k \times \mathcal{M}_k \times \mathcal{S}_k \times \mathcal{P}_k$.

That is, similarly to one the hardness results of [DNR+09], we consider datasets D that contain n rows of the form $x_1 = (vk_1, m_1, \sigma_1, \rho_1), \ldots, x_n = (vk_n, m_n, \sigma_n, \rho_n)$ each corresponding to a verification key, message, and signature from the digital signature scheme, and to a zap verifier's public coin tosses.

Let $L \in \mathbf{NP}$ be the language

$$vk \in (L \cap \mathcal{K}_k) \iff \exists (m, \sigma) \in \mathcal{M}_k \times \mathcal{S}_k \text{ s.t. } \text{Ver}(vk, m, \sigma) = 1$$

which has the natural witness relation

$$R_L = \bigcup_k \{(vk, (m, \sigma)) \in \mathcal{K}_k \times (\mathcal{M}_k \times \mathcal{S}_k) \ : \ \text{Ver}(vk, m, \sigma) = 1\}.$$

Define

Proof Space: $\Pi_k = \{0,1\}^{\ell_4}$ where $\ell_4 = |\pi|$ for $\pi \leftarrow P(1^{k_c}, vk, (m, \sigma), \rho)$ for $vk \in (L \cap \mathcal{K}_k)$ with witness $(m, \sigma) \in \mathcal{M}_k \times \mathcal{S}_k$ and public coins $\rho \in \mathcal{P}_k$, and
Output Space: $\mathcal{R}_k = \mathcal{K}_k \times \mathcal{P}_k \times \Pi_k$.

Definition of Utility Function u. We now specify our computational task of interest via a utility function $u : \mathcal{X}_k^n \times \mathcal{R}_k \to \{0,1\}$. For any string $vk \in \mathcal{K}_k$ and $D = ((vk_1, m_1, \sigma_1, \rho_1), \cdots, (vk_n, m_n, \sigma_n, \rho_n)) \in \mathcal{X}_k^n$ define an auxiliary function

$$f_{vk,\rho}(D) = \#\{i \in [n] : vk_i = vk \wedge \rho_i = \rho \wedge \text{Ver}(vk, m_i, \sigma_i) = 1\}.$$

That is, $f_{vk,\rho}$ is the number of elements of the dataset D with verification key equal to vk and public coin string equal to ρ for which (m_i, σ_i) is a valid message-signature pair under vk. We now define $u(D, (vk, \rho, \pi)) = 1$ iff

$$\begin{aligned} f_{vk,\rho}(D) \geq 9n/10 \quad &\wedge \quad V(1^{k_c}, vk, \rho, \pi) = 1 \\ &\text{or} \\ f_{vk',\rho'}(D) < 9n/10 \quad &\text{for all } vk' \in \mathcal{K}_k \text{ and } \rho' \in \mathcal{P}_k. \end{aligned}$$

That is, the utility function u is satisfied if either (1) many entries of D contain valid message-signature pairs under the same verification key vk with the same public coin string ρ and π is a valid proof for statement vk using ρ, or (2) it is not the case that many entries of D contain valid message-signature pairs under the same verification key, with the same public coin string (in which case any response (vk, ρ, π) is acceptable).

3.2 An Inefficient Differentially Private Algorithm

We begin by showing that there is an inefficient differentially private mechanism that achieves high utility under u.

Proposition 1. *Let $k \in \mathbb{N}$. For every $\varepsilon > 0$, there exists an $(\varepsilon, 0)$-differentially private algorithm $M_k^{\mathrm{unb}} : \mathcal{X}_k^n \to \mathcal{R}_k$ such that, for every $\beta > 0$, every $n \geq \frac{10}{\varepsilon} \log(2 \cdot |\mathcal{K}_k| \cdot |\mathcal{P}_k|/\beta)) = \mathrm{poly}(1/\varepsilon, \log(1/\beta), k)$ and $D \in (\mathcal{K}_k \times \mathcal{M}_k \times \mathcal{S}_k \times \mathcal{P}_k)^n$,*

$$\Pr_{(vk,\rho,\pi) \leftarrow M_k^{\mathrm{unb}}(D)} [u(D, (vk, \rho, \pi)) = 1] \geq 1 - \beta$$

Remark 2. While the mechanism M^{unb} considered here is only accurate for $n \geq \Omega(\log |\mathcal{P}_k|)$, it is also possible to use "stability techniques" [DL09, TS13] to design an (ε, δ)-differentially private mechanism that achieves high utility for $n \geq O(\log(1/\delta)/\varepsilon)$ for $\delta > 0$. We choose to provide a "pure" ε-differentially private algorithm here to make our separation more dramatic: Both the inefficient differentially private mechanism and the efficient SIM-CDP mechanism achieve pure $(\varepsilon, 0)$-privacy, whereas no efficient mechanism can even achieve (ε, δ)-differential privacy with $\delta > 0$.

Our algorithm relies on standard differentially private techniques for identifying frequently occurring elements in a dataset.

Report Noisy Max. Consider a data universe $\mathcal{X}$. A predicate $q : \mathcal{X} \to \{0, 1\}$ defines a *counting query* over the set of datasets $\mathcal{X}^n$ as follows: For $D = (x_1, \ldots, x_n) \in \mathcal{X}^n$, we abuse notation by defining $q(D) = \sum_{i=1}^n q(x_i)$. We further say that a collection of counting queries Q is *disjoint* if, whenever $q(x) = 1$ for some $q \in Q$ and $x \in \mathcal{X}$, we have $q'(x) = 0$ for every other $q' \neq q$ in Q. (Thus, disjoint counting queries slightly generalize *point functions*, which are each supported on exactly one element of the domain $\mathcal{X}$.)

The "Report Noisy Max" algorithm [DR14], combined with observations of [BV16], can efficiently and privately identify which of a set of disjoint counting queries is (approximately) the largest on a dataset D, and release its identity along with the corresponding noisy count. We sketch the proof of the following proposition in Appendix A.

Proposition 2 (Report Noisy Max). *Let Q be a set of efficiently computable and sampleable disjoint counting queries over a domain $\mathcal{X}$. Further suppose that for every $x \in \mathcal{X}$, the query $q \in Q$ for which $q(x) = 1$ (if one exists) can be identified efficiently. For every $n \in \mathbb{N}$ and $\varepsilon > 0$ there is an mechanism $F : \mathcal{X}^n \to \mathcal{X} \times \mathbb{R}$ such that*

1. *F runs in time $\mathrm{poly}(n, \log |\mathcal{X}|, \log |Q|, 1/\varepsilon)$.*
2. *F is ε-differentially private.*
3. *For every dataset $D \in \mathcal{X}^n$, let $q_{\mathrm{OPT}} = \mathrm{argmax}_{q \in Q} q(D)$ and $\mathrm{OPT} = q_{\mathrm{OPT}}(D)$. Let $\beta > 0$. Then with probability at least $1 - \beta$, the algorithm F outputs a solution $(\hat{q}, a)$ such that $a \geq \hat{q}(D) - \gamma/2$ where $\gamma = \frac{8}{\varepsilon} \cdot (\log |Q| + \log(1/\beta))$. Moreover, if $\mathrm{OPT} - \gamma > \max_{q \neq q_{\mathrm{OPT}}} q(D)$, then $\hat{q} = \mathrm{argmax}_{q \in Q} q(D)$.*

We are now ready to describe our unbounded algorithm M_k^{unb} as Algorithm 1. We prove Proposition 1 via the following two claims, capturing the privacy and utility guarantees of M_k^{unb}, respectively.

Algorithm 1. M_k^{unb}

Input: Dataset $D \in (\mathcal{K}_k \times \mathcal{M}_k \times \mathcal{S}_k \times \mathcal{P}_k)^n$
Output: Triple $(vk, \rho, \pi) \in \mathcal{K}_k \times \mathcal{P}_k \times \Pi_k$

1. Run the Report Noisy Max algorithm on D with privacy parameter ε using the set of disjoint counting queries $\{f_{vk,\rho} : vk \in \mathcal{K}_k, \rho \in \mathcal{P}_k\}$, obtaining an answer $((vk, \rho), a)$.
2. If $a < 7n/10$, output $(\perp, \perp, \perp)$ and halt. Otherwise:
3. Choose the lexicographically first $(m^*, \sigma^*) \in \mathcal{M}_k \times \mathcal{S}_k$ such that $\text{Ver}(vk, m^*, \sigma^*) = 1$ (If no such pair exists, output $(\perp, \perp, \perp)$ and halt)
4. Let $\pi = P(1^{k_c}, vk, (m^*, \sigma^*), \rho)$, and output (vk, ρ, π).

Lemma 1. *The algorithm M_k^{unb} is ε-differentially private.*

Proof. The algorithm M_k^{unb} accesses its input dataset D only through the ε-differentially private Report Noisy Max algorithm (Proposition 2). Hence, by the closure of differential privacy under post-processing, M_k^{unb} is also ε-differentially private.

Lemma 2. *The algorithm M_k^{unb} is $(1-\beta)$-useful for any number of rows $n \geq \frac{20}{\varepsilon}(\log(|\mathcal{K}_k| \cdot |\mathcal{P}_k|/\beta))$.*

Proof. If $f_{vk,\rho}(D) < 9n/10$ for every vk and ρ, then the utility of the mechanism is always 1. Therefore, it suffices to consider the case when there exist vk, ρ for which $f_{vk,\rho}(D) \geq 9n/10$. When such vk and ρ exist, observe that we have $f_{vk',\rho'}(D) \leq n/10$ for every other pair $(vk', \rho') \neq (vk, \rho)$. Thus, as long as

$$\frac{9n}{10} - \frac{n}{10} > \frac{8}{\varepsilon} \cdot (\log(|\mathcal{K}_k| \cdot |\mathcal{P}_k|) + \log(1/\beta)),$$

the Report Noisy Max algorithm successfully identifies the correct vk, ρ in Step 1 with probability all but β (Proposition 2). Moreover, the reported value a is at least $7n/10$. By the perfect completeness of the zap proof system, the algorithm produces a useful triple (vk, ρ, π) in Step 4. Thus, the mechanism as a whole is $(1-\beta)$-useful.

3.3 A SIM-CDP Algorithm

We define a PPT algorithm M_k^{CDP} in Algorithm 2, which we argue is an efficient, SIM-CDP algorithm achieving high utility with respect to u.

Algorithm 2. M_k^{CDP}

Input: Dataset $D \in (\mathcal{K}_k \times \mathcal{M}_k \times \mathcal{S}_k \times \mathcal{P}_k)^n$
Output: Triple $(vk, \rho, \pi) \in \mathcal{K}_k \times \mathcal{P}_k \times \Pi_k$

1. Run the Report Noisy Max algorithm on D with privacy parameter ε using the set of disjoint counting queries $\{f_{vk,\rho} : vk \in \mathcal{K}_k, \rho \in \mathcal{P}_k\}$, obtaining an answer $((vk, \rho), a)$.
2. If $a < 7n/10$, output $(\perp, \perp, \perp)$ and halt. Otherwise:
3. Select the first $(vk_i = vk, m_i, \sigma_i) \in D$ such that $\mathrm{Ver}(vk, m_i, \sigma_i) = 1$ (If there is no such pair in the dataset, output $(\perp, \perp, \perp)$ and halt).
4. Let $\pi = P(1^{k_c}, vk, (m_i, \sigma_i), \rho)$, and output (vk, ρ, π).

The only difference between M_k^{CDP} and the inefficient algorithm M_k^{unb} occurs in Step 3, where we have replaced the inefficient process of finding a canonical message-signature pair (m^*, σ^*) with selecting a message-signature pair (m_i, σ_i) in the dataset. Since all the other steps (Report Noisy Max and the zap prover's algorithm) are efficient, M_k^{CDP} runs in polynomial time. However, this change renders M_k^{CDP} statistically non-differentially private, since a (computationally unbounded) adversary could reverse engineer the proof π produced in Step 4 to recover the pair (m_i, σ_i) contained in the dataset. On the other hand, the witness indistinguishability of the proof system implies that M_k^{CDP} is nevertheless computationally differentially private:

Lemma 3. *The algorithm* M_k^{CDP} *is* ε-SIM-CDP *provided that* $n \geq (20/\varepsilon) \cdot (k + \log|\mathcal{K}_k| + \log|\mathcal{P}_k|) = \mathrm{poly}(k, 1/\varepsilon)$.

Proof. Indeed, we will show that $M'_k = M_k^{\mathrm{unb}}$ is secure as the simulator for $M_k = M_k^{\mathrm{CDP}}$. That is, we will show that for any $\mathrm{poly}(k)$-size adversary A, that

$$\Pr[A(M_k^{\mathrm{CDP}}(D)) = 1] - \Pr[A(M_k^{\mathrm{unb}}(D)) = 1] \leq \mathrm{negl}(k).$$

First observe that by definition, the first two steps of the mechanisms are identical. Now define, for either mechanism M_k^{unb} or M_k^{CDP}, a "bad" event B where the mechanism in Step 1 produces a pair $((vk, \rho), a)$ for which $f_{vk,\rho}(D) = 0$, but does *not* output $(\perp, \perp, \perp)$ in Step 2. For either mechanism, the probability of the bad event B is $\mathrm{negl}(k)$, as long as $n \geq (20/\varepsilon) \cdot (k + \log(|\mathcal{K}_k| \cdot |\mathcal{P}_k|))$. This follows from the utility guarantee of the Report Noisy Max algorithm (Proposition 2), setting $\beta = 2^{-k}$.

Thus, it suffices to show that for any fixing of the coins of both mechanisms in Steps 1 and 2 in which B does not occur, that the mechanisms $M_k^{\mathrm{CDP}}(D)$ and $M_k^{\mathrm{unb}}(D)$ are indistinguishable. There are now two cases to consider based on the coin tosses in Steps 1 and 2:

Case 1: Both Mechanisms Output $(\perp, \perp, \perp)$ *in Step 2.* In this case,

$$\Pr[A(M_k^{\mathrm{CDP}}(D)) = 1] = \Pr[A(\perp, \perp, \perp) = 1] = \Pr[A(M_k^{\mathrm{unb}}(D)) = 1],$$

and the mechanisms are perfectly indistinguishable.

Case 2: Step 1 Produced a Pair $((vk, \rho), a)$ *for which* $f_{vk,\rho}(D) > 0$. In this case, we reduce to the indistinguishability of the zap proof system. Let $(vk_i = vk, m_i, \sigma_i)$ be the first entry of D for which $\mathrm{Ver}(vk, m_i, \sigma_i) = 1$, and let (m^*, σ^*) be the lexicographically first message-signature pair with $\mathrm{Ver}(vk, m^*, \sigma^*) = 1$. The proofs we are going to distinguish are $\pi_{\mathrm{CDP}} \leftarrow P(1^{k_c}, vk, (m_i, \sigma_i), \rho)$ and $\pi_{\mathrm{unb}} \leftarrow P(1^{k_c}, vk, (m^*, \sigma^*), \rho)$. Let $A^{\mathrm{zap}}(1^{k_c}, \rho, \pi) = A(vk, \rho, \pi)$. Then we have

$$\Pr[A(M_k^{\mathrm{CDP}}(D)) = 1] = \Pr[A^{\mathrm{zap}}(1^{k_c}, \rho, \pi_{\mathrm{CDP}}) = 1]$$

and

$$\Pr[A(M_k^{\mathrm{unb}}(D)) = 1] = \Pr[A^{\mathrm{zap}}(1^{k_c}, \rho, \pi_{\mathrm{unb}}) = 1].$$

Thus, indistinguishability of $M_k^{\mathrm{CDP}}(D)$ and $M_k^{\mathrm{unb}}(D)$ follows from the witness indistinguishability of the zap proof system.

The proof of Lemma 2 also shows that M_k is useful for u.

Lemma 4. *The algorithm* M_k^{CDP} *is* $(1-\beta)$*-useful for any number of rows* $n \geq \frac{20}{\varepsilon}(\log(2 \cdot |\mathcal{K}_k| \cdot |\mathcal{P}_k|/\beta))$.

3.4 Infeasibility of Differential Privacy

We now show that any efficient algorithm achieving high utility cannot be differentially private. In fact, like many prior hardness results, we provide an attack A that does more than violate differential privacy. Specifically we exhibit a distribution on datasets such that, given any useful answer produced by an efficient mechanism, A can with high probability recover a row of the input dataset. Following [DNR+09], we work with the following notion of a re-identifiable dataset distribution.

Definition 8 (Re-identifiable Dataset Distribution). *Let* $u : \mathcal{X}^n \times \mathcal{R} \to \{0,1\}$ *be a utility function. Let* $\{\mathcal{D}_k\}_{k\in\mathbb{N}}$ *be an ensemble of distributions over* $(D_0, z) \in \mathcal{X}^{n(k)+1} \times \{0,1\}^{\mathrm{poly}(k)}$ *for* $n(k) = \mathrm{poly}(k)$. *(Think of* D_0 *as a dataset on* $n+1$ *rows, and* z *as a string of auxiliary information about* D_0*). Let* $(D, D', i, z) \leftarrow \tilde{\mathcal{D}}_k$ *denote a sample from the following experiment: Sample* $(D_0 = (x_1, \ldots, x_{n+1}), z) \leftarrow \mathcal{D}_k$ *and* $i \in [n]$ *uniformly at random. Let* $D \in \mathcal{X}^n$ *consist of the first* n *rows of* D_0*, and let* D' *be the dataset obtained by replacing* x_i *in* D *with* x_{n+1}.

We say the ensemble $\{\mathcal{D}_k\}_{k\in\mathbb{N}}$ *is a* re-identifiable dataset distribution with respect to u *if there exists a (possibly inefficient) adversary* A *and a negligible function* $\mathrm{negl}(\cdot)$ *such that for all polynomial-time mechanisms* M_k*,*

1. *Whenever* M_k *is useful,* A *recovers a row of* D *from* $M_k(D)$*. That is, for any PPT* M_k*:*

$$\Pr_{\substack{(D,D',i,z)\leftarrow\tilde{\mathcal{D}}_k \\ r\leftarrow M_k(D)}}[u(D,r) = 1 \ \wedge \ A(r,z) \notin D] \leq \mathrm{negl}(k).$$

2. *A cannot recover the row x_i not contained in D' from $M_k(D')$. That is, for any algorithm M_k:*

$$\Pr_{\substack{(D,D',i,z)\leftarrow\tilde{\mathcal{D}}_k \\ r\leftarrow M_k(D')}}[A(r,z)=x_i]\leq \mathrm{negl}(k),$$

where x_i is the i-th row of D.

Proposition 3 ([DNR+09]). *If a distribution ensemble $\{\mathcal{D}_k\}_{k\in\mathbb{N}}$ on datasets of size $n(k)$ is re-identifiable with respect to a utility function u, then for every $\gamma>0$ and $\alpha(k)$ with $\min\{\alpha,(1-8\alpha)/8n^{1+\gamma}\}\geq\mathrm{negl}(k)$, there is no polynomial-time $(\varepsilon=\gamma\log(n),\delta=(1-8\alpha)/2n^{1+\gamma})$-differentially private mechanism $\{M_k\}_{k\in\mathbb{N}}$ that is α-useful for u.*

In particular, for every $\varepsilon=O(\log k),\alpha=1/\operatorname{poly}(k)$, there is no polynomial-time $(\varepsilon,1/n^2)$-differentially private and α-useful mechanism for u.

Construction of a Re-identifiable Dataset Distribution. For $k\in\mathbb{N}$, recall that the digital signature scheme induces a choice of verification key space $\mathcal{K}_k$, message space $\mathcal{M}_k$, and signature space $\mathcal{S}_k$, each on $\operatorname{poly}(k)$-bit strings. Let $n=\operatorname{poly}(k)$. Define a distribution $\{\mathcal{D}_k\}_{k\in\mathbb{N}}$ as follows. To sample (D_0,z) from $\mathcal{D}_k$, first sample a key pair $(sk,vk)\leftarrow\mathrm{Gen}(1^k)$. Sample messages $m_1,\ldots,m_{n+1}\leftarrow\mathcal{M}_k$ uniformly at random. Then let $\sigma_i\leftarrow\mathrm{Sign}(sk,m_i)$ for each $i=1,\ldots,n+1$. Let the dataset $D_0=(x_1,\ldots,x_{n+1})$ where $x_i=(vk,m_i,\sigma_i,\rho)$, and set the auxiliary string $z=(vk,\rho)$.

Proposition 4. *The distribution $\{\mathcal{D}_k\}_{k\in\mathbb{N}}$ defined above is re-identifiable with respect to the utility function u.*

Proof. We define an adversary $A:\mathcal{R}_k\times\mathcal{K}_k\to\mathcal{X}_k$. Consider an input to A of the form $(r,z)=((vk',\rho',\pi),(vk,\rho))$. If $vk'\neq vk$ or $\rho'\neq\rho$ or $\pi=\perp$, then output $(vk,\perp,\perp,\rho)$. Otherwise, run the zap extraction algorithm $E(1^{k_c},vk,\rho,\pi)$ to extract a witness (m,σ), and output the resulting (vk,m,σ,ρ). Note that the running time of A is $2^{O(k_c)}$.

We break the proof of re-identifiability into two lemmas. First, we show that A can successfully recover a row in D from any useful answer:

Lemma 5. *Let $M_k:\mathcal{X}_k^n\to\mathcal{R}_k$ be a PPT algorithm. Then*

$$\Pr_{\substack{(D,D',i,z)\leftarrow\tilde{\mathcal{D}}_k \\ r\leftarrow M_k(D)}}[u(D,r)=1\ \wedge\ A(r,z)\notin D]\leq\mathrm{negl}(k).$$

Proof. First, if $u(D,r)=u(D,(vk',\rho',\pi))=1$, then $vk'=vk$, $\rho'=\rho$, and $V(1^k,vk,\rho,\pi)=1$. In other words, π is a valid proof that $vk\in(L\cup\mathcal{K}_k)$. Hence, by the extractability of the zap proof system, we have that $(m,\sigma)=E(1^{k_c},vk,\rho,\pi)$ satisfies $(vk,(m,\sigma))\in R_L$; namely $\mathrm{Ver}(vk,m,\sigma)=1$ with overwhelming probability over the choice of ρ.

Algorithm 3. Forgery algorithm $A_{\text{forge}}^{\text{Sign}(sk,\cdot)}$

Input: Verification key vk
Output: Message-signature pair (m, σ)

1. Sample public coins $\rho \leftarrow \mathcal{P}_k$.
2. Invoke the signing oracle n times on random messages $m_i \in \mathcal{M}_k$ to get message-signature pairs $(m_1, \sigma_1), \cdots, (m_n, \sigma_n)$, and construct the dataset $D = \{(vk, m_i, \sigma_i, \rho)\}_{i \in [n]}$.
3. Obtain the result $r = (vk, \rho, \pi)$ from $M_k(D)$.
4. Output (m, σ) where $(vk, m, \sigma, \rho) \leftarrow A(r, (vk, \rho))$.

Next, we use the exponential security of the digital signature scheme to show that the extracted pair (m, σ) must indeed appear in the dataset D. Consider the following forgery adversary for the digital signature scheme.

The dataset built by the forgery algorithm $A_{\text{forge}}^{\text{Sign}(sk,\cdot)}$ is identically distributed to a sample D from the experiment $(D, D', i, z) \leftarrow \tilde{D}_k$. Since a message-signature pair (m, σ) appears in D if and only if the signing oracle was queried on m to produce σ, we have

$$\Pr_{\substack{(sk,vk)\leftarrow \text{Gen}(1^k) \\ (m,\sigma)\leftarrow A_{\text{forge}}^{\text{Sign}(sk,\cdot)}(vk)}} [\text{Ver}(m, \sigma) = 1 \wedge (m, \sigma) \notin Q] = \Pr_{\substack{(D,D',i,z)\leftarrow \tilde{\mathcal{D}}_k \\ r \leftarrow M_k(D)}} [u(D, r) = 1 \ \wedge \ (vk, m, \sigma, \rho) = A(r, z) \notin D].$$

The running time of the algorithm A, and hence the algorithm $A_{\text{forge}}^{\text{Sign}(sk,\cdot)}$, is $2^{O(k_c)} = 2^{o(k^c)}$. Thus, by the existential unforgeability of the digital signature scheme against 2^{k^c}-time adversaries, this probability is negligible in k.

We next argue that A cannot recover row $x_i = (vk, m_i, \sigma_i, \rho)$ from $M_k(D')$, where we recall that D' is the dataset obtained by replacing row x_i in D with row x_{n+1}.

Lemma 6. *For every algorithm M_k:*

$$\Pr_{\substack{(D,D',i,z)\leftarrow \tilde{\mathcal{D}}_k \\ r \leftarrow M_k(D')}} [A(r, z) = x_i] \leq \text{negl}(k),$$

where x_i is the i-th row of D.

Proof. Since in $D_0 = ((vk, m_1, \text{Sign}_{vk}(m_1), \rho) \cdots, (vk, m_{n+1}, \text{Sign}_{vk}(m_{n+1}), \rho))$, the messages $m_1, \cdots, m_{n+1}$ are drawn independently, the dataset $D' = (D_0 - \{(vk, m_i, \sigma_i, \rho)\}) \cup \{(vk, m_{n+1}, \sigma_{n+1}, \rho)\}$ contains no information about message m_i. Since m_i is drawn uniformly at random from the space $\mathcal{M}_k = \{0,1\}^k$, the probability that $A(r, z) = A(M_k(D'), (vk, \rho))$ outputs row x_i is at most $2^{-k} = \text{negl}(k)$.

Re-identifiability of the distribution $\tilde{\mathcal{D}}_k$ follows by combining Lemmas 5 and 6.

4 Limits of CDP in the Client-Server Model

We revisit the techniques of [GKY11] to exhibit a setting in which efficient CDP mechanisms cannot do much better than information-theoretically differentially private mechanisms. In particular, we consider computational tasks with output in some discrete space (or which can be reduced to some discrete space) $\mathcal{R}_k$, and with utility measured via functions of the form $g : \mathcal{R}_k \times \mathcal{R}_k \to \mathbb{R}$. We show that if $(\mathcal{R}_k, g)$ forms a metric space with $O(\log k)$-doubling dimension (and other properties described in detail later), then CDP mechanisms can be efficiently transformed into differentially private ones. In particular, when $\mathcal{R}_k = \mathbb{R}^d$ for $d = O(\log k)$ and utility is measured by an L_p-norm, we can transform a CDP mechanism into a differentially private one.

The result in this section is incomparable to that of [GKY11]. We incur a constant-factor blowup in error, rather than a negligible additive increase as in [GKY11]. However, in the case that utility is measured by an L_p norm, our result applies to output spaces of dimension that grow logarithmically in the security parameter k, whereas the result of [GKY11] only applies to outputs of constant dimension. In addition, we handle IND-CDP directly, while [GKY11] prove their results for SIM-CDP, and then extend them to IND-CDP by applying a reduction of [MPRV09].

4.1 Task and Utility

Consider a computational task with discrete output space $\mathcal{R}_k$. Let $g : \mathcal{R}_k \times \mathcal{R}_k \to \mathbb{R}$ be a metric on $\mathcal{R}_k$. We impose the following additional technical conditions on the metric space $(\mathcal{R}_k, g)$:

Definition 9 (Property $\mathcal{L}$). *A metric space formed by a discrete set $\mathcal{R}_k$ and a metric g has* property $\mathcal{L}$ *if*

1. *The doubling dimension of $(\mathcal{R}_k, g)$ is $O(\log k)$. That is, for every $a \in \mathcal{R}_k$ and radius $r > 0$, the ball $B(a, r)$ centered at a with radius r is contained in a union of* $\text{poly}(k)$ *balls of radius $r/2$.*
2. *The metric space is uniform. Namely, for any fixed radius r, the size of a ball of radius r is independent of its center.*
3. *Given a center $a \in \mathcal{R}_k$ and a radius $r > 0$, the membership in the ball $B(a, r)$ can be checked in time* $\text{poly}(k)$.
4. *Given a center $a \in \mathcal{R}_k$ and a radius $r > 0$, a uniformly random point in $B(a, r)$ can be sampled in time* $\text{poly}(k)$.

Given a metric g, we can define a utility function measuring the accuracy of a mechanism with respect to g:

Definition 10 (α-accuracy). *Consider a dataset space $\mathcal{X}_k$. Let $q_k : \mathcal{X}_k^n \to \mathcal{R}_k$ be any function on datasets of size n. Let $M_k : \mathcal{X}_k^n \to \mathbb{N}_k^d$ be a mechanism for approximating q_k. We say that M_k is α_k-*accurate for q_k with respect to g *if with overwhelming probability, the error of M_k as measured by g is at most α_k. Namely, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that*

$$\Pr[g(q_k(D), M_k(D)) \leq \alpha_k] \geq 1 - \mathrm{negl}(k).$$

We take the failure probability here to be negligible primarily for aesthetic reasons. In general, taking the failure probability to be β_k will yield in our result below a mechanism that is $(\varepsilon_k, \beta_k + \mathrm{negl}(k))$-differentially private.

Moreover, for reasonable queries q_k, taking the failure probability to be negligible is essentially without loss of generality. We can reduce the failure probability of a mechanism M_k from constant to negligible by repeating the mechanism $O(\log^2 k)$ times and taking a median. By composition theorems for differential privacy, this incurs a cost of at most $O(\log^2 k)$ in the privacy parameters. But we can compensate for this loss in privacy by first increasing the sample size n by a factor of $O(\log^2 k)$, and then applying a "secrecy-of-the-sample" argument [KLN+11] – running the original mechanism on a random subsample of the larger dataset. This step maintains accuracy as long as the query q_k generalizes from random subsamples.

4.2 Result and Proof

Theorem 5. *Let $(\mathcal{R}_k, g)$ be a metric space with property $\mathcal{L}$. Suppose $M_k : \mathcal{X}_k^n \to \mathcal{R}_k$ is an efficient ε_k-*IND-CDP *mechanism that is α_k-accurate for some function q_k with respect to g. Then there exists an efficient $(\varepsilon, \mathrm{negl}(k))$-differentially private mechanism $\hat{M}_k$ that is $O(\alpha_k)$-accurate for q_k with respect to g.*

Proof. We denote a ball centered at a with radius r in the metric space $(\mathcal{R}_k, g)$ by

$$B(a, r) = \{x \in \mathcal{R}_k \ : \ g(a, x) \leq r\}.$$

We also let $V(r) \stackrel{\mathrm{def}}{=} |B(a, r)|$ for any $a \in \mathcal{R}_k$, which is well-defined due to the uniformity of the metric space. Now we define a mechanism $\hat{M}_k$ which outputs a uniformly random point from $B(M_k(x), c_k)$, where $c_k > 0$ is a parameter be determined later. Note that $\hat{M}_k$ can be implemented efficiently due to the efficient sampling condition of property $\mathcal{L}$. Since g satisfies the triangle inequality, $\hat{M}_k$ is $(\alpha_k + c_k)$-accurate. Thus it remains to prove that $\hat{M}_k$ is $(\varepsilon, \mathrm{negl}(k))$-DP.

The key observation is that, for every $D \in \mathcal{X}_k^n$ and $s \in \mathcal{R}_k$,

$$\Pr[\hat{M}_k(D) = s] = \frac{1}{V(c_k)} \Pr[M_k(D) \in B(s, c_k)]$$

For all sets $S \subseteq \mathcal{R}_k$, we thus have

$$
\begin{aligned}
&\Pr[\hat{M}_k(D) \in S] \\
&\leq \left(\sum_{s \in S \cap B(q_k(D), \alpha_k + c_k)} \Pr[\hat{M}_k(D) = s] \right) + \Pr[\hat{M}_k(D) \notin B(q_k(D), \alpha_k + c_k)] \\
&\leq \left(\sum_{s \in S \cap B(q_k(D), \alpha_k + c_k)} \frac{1}{V(c_k)} \Pr[M_k(D) \in B(s, c_k)] \right) + \mathrm{negl}(k) \\
&\quad \text{(by the above observation and } \alpha_k\text{-accuracy of } M_k\text{)} \\
&\leq \left(\sum_{s \in S \cap B(q_k(D), \alpha_k + c_k)} \frac{1}{V(c_k)} \left(e^{\varepsilon} \Pr[M_k(D') \in B(s, c_k)] + \mathrm{negl}'(k) \right) \right) + \mathrm{negl}(k) \\
&\quad \text{(since } M_k \text{ is IND-CDP, and testing containment in } B(s, c_k) \text{ is efficient)} \\
&\leq \sum_{s \in S \cap B(q_k(D), \alpha_k + c_k)} \left[e^{\varepsilon_k} \Pr[\hat{M}_k(D') = s] + \frac{1}{V(c_k)} \mathrm{negl}'(k) \right] + \mathrm{negl}(k) \\
&\leq e^{\varepsilon_k} \Pr[M_k(D') \in S] + \frac{V(\alpha_k + c_k)}{V(c_k)} \cdot \mathrm{negl}'(k) + \mathrm{negl}(k).
\end{aligned}
$$

By the bounded doubling dimension of $(\mathcal{R}_k, g)$, we can set $c_k = O(\alpha_k)$ to make $V(\alpha_k + c_k)/V(c_k) = \mathrm{poly}(k)$. Hence $\hat{M}_k$ is a $(\varepsilon_k, \mathrm{negl}(k))$-differentially private algorithm.

L_p-norm Case. Many natural tasks can be captured by outputs in $\mathbb{R}^d$ with utility measured by an L_p norm (e.g. counting queries). Since we work with efficient mechanisms, we may assume that our mechanisms always have outputs represented by $\mathrm{poly}(k)$ bits of precision. The level of precision is unimportant, so we may assume an output space represented by k bits of precision for simplicity. By rescaling, we may assume all responses are integers and take values in $\mathbb{N}_k \stackrel{\text{def}}{=} \mathbb{N} \cap [0, 2^k]$. When $d = O(\log k)$, the doubling dimension of the new discrete metric space induced by the L_p-norm on integral points is $O(\log k)$ ([GKL03] shows that the subspace of $\mathbb{R}^d$ equipped with L_p norm has doubling dimension $O(d)$). Now the metric space almost satisfies property $\mathcal{L}$, with the exception of the uniformity condition. This is because the sizes of balls close the boundary of $\mathbb{N}_k$ are smaller than those in the interior. However, we can apply Theorem 5 to first construct a statistically DP mechanism with outputs in the larger uniform metric space $\mathbb{N}^d$. Then we may construct the final statistical mechanism $\hat{M}_k$, by projecting answers that are not in $\mathbb{N}_k^d$ to the closest point in $\mathbb{N}_k^d$. By post-processing, the modified mechanism $\hat{M}_k$ is still differentially private. Moreover, its utility is only improved since $\hat{M}_k$ can only get closer to the true query answer in every coordinate. Therefore, we have the following corollary.

Corollary 1. *Let $M_k : \mathcal{X}_k^n \to \mathbb{R}^d$ with $d = O(\log k)$ be an efficient ε_k-*IND-CDP *mechanism that is α_k-accurate for some function q_k when error is measured*

by an L_p-norm. Then there exists an efficient $(\varepsilon, \mathrm{negl}(k))$-differentially private mechanism $\hat{M}_k$ that is $O(\alpha_k)$-accurate for q_k.

Acknowledgements. We are grateful to an anonymous reviewer for pointing out that our original construction based on non-interactive witness indistinguishable proofs could be modified to accommodate 2-message proofs (zaps).

A Missing Proofs

A.1 Proof of Proposition 2

Proposition 2 (Report Noisy Max). *Let Q be a set of efficiently computable and sampleable disjoint counting queries over a domain $\mathcal{X}$. Further suppose that for every $x \in \mathcal{X}$, the query $q \in Q$ for which $q(x) = 1$ (if one exists) can be identified efficiently. For every $n \in \mathbb{N}$ and $\varepsilon > 0$ there is an mechanism $F : \mathcal{X}^n \to \mathcal{X} \times \mathbb{R}$ such that*

1. *F runs in time $\mathrm{poly}(n, \log|\mathcal{X}|, \log|Q|, 1/\varepsilon)$.*
2. *F is ε-differentially private.*
3. *For every dataset $D \in \mathcal{X}^n$, let $q_{\mathrm{OPT}} = \mathrm{argmax}_{q \in Q} q(D)$ and $\mathrm{OPT} = q_{\mathrm{OPT}}(D)$. Let $\beta > 0$. Then with probability at least $1 - \beta$, the algorithm F outputs a solution $(\hat{q}, a)$ such that $a \geq \hat{q}(D) - \gamma/2$ where $\gamma = \frac{8}{\varepsilon} \cdot (\log|Q| + \log(1/\beta))$. Moreover, if $\mathrm{OPT} - \gamma > \max_{q \neq q_{\mathrm{OPT}}} q(D)$, then $\hat{q} = \mathrm{argmax}_{q \in Q} q(D)$.*

The proof of Proposition 2 relies on the existence of an efficient sanitizer for the disjoint query class Q. Such a sanitizer appears in [Vad16], and is based on ideas of [BV16]. (There, it is stated for the specific class of point functions, but immediately extends to disjoint counting queries).

Proposition 3 ([Vad16, Theorem 7.1]). *Let Q be a set of efficiently computable and sampleable disjoint counting queries over a domain $\mathcal{X}$. Suppose that for every element $x \in \mathcal{X}$, the query $q \in Q$ for which $q(x) = 1$ (if one exists) can be identified in time $\mathrm{polylog}(|X|)$. Let $\beta > 0$. Then there exists an algorithm San running in time $\mathrm{poly}(n, \log|X|, 1/\varepsilon)$ for which the following holds. For any database $D \in \mathcal{X}^n$, with probability at least $1 - \beta$, the algorithm San produces a "synthetic database" $\hat{D} \in \mathcal{X}^m$ such that*

$$|q(D) - \frac{n}{m} q(\hat{D})| < \frac{4(\log|Q| + \log(1/\beta))}{\varepsilon}$$

for every $q \in Q$.

Proof (Proof of Proposition 2). Consider the algorithm F which first runs the algorithm San on its input dataset to obtain a synthetic dataset $\hat{D}$, and then outputs the pair $(\hat{q}, \frac{n}{m}\hat{q}(\hat{D}))$ where $\hat{q} = \mathrm{argmax}_{q \in Q} q(\hat{D})$. The algorithm F inherits its efficiency and differential privacy from San. To see that it useful, suppose San indeed produces a database $\hat{D} \in \mathcal{X}^m$ for which

$$|q(D) - \frac{n}{m} q(\hat{D})| < \frac{4(\log|Q| + \log(1/\beta))}{\varepsilon}$$

for every $q \in Q$. Let $q_{\mathrm{OPT}} = \mathrm{argmax}_{q \in Q}\, q(D)$, and $\gamma = 8(\log|Q| + \log(1/\beta))/\varepsilon$. Then $\frac{n}{m}\hat{q}(\hat{D}) \geq \frac{n}{m} q_{\mathrm{OPT}}(\hat{D}) \geq q_{\mathrm{OPT}}(D) - \gamma/2$. Moreover, suppose $q_{\mathrm{OPT}}(D) - \gamma > \max_{q \neq q_{\mathrm{OPT}}} q(D)$. Then for any $q' \neq q_{\mathrm{OPT}}$, we have

$$\frac{n}{m} q'(\hat{D}) < q'(D) + \gamma/2 < q_{\mathrm{OPT}}(D) - \gamma/2 \leq \frac{n}{m}\hat{q}(\hat{D}).$$

Hence $q'(\hat{D}) < \hat{q}(\hat{D})$ for every $q' \neq q_{\mathrm{OPT}}$, and hence $\hat{q} = q_{\mathrm{OPT}}$.

B Extractability for Zap Proof Systems

B.1 Non-interactive Zero Knowledge Proofs

Most known constructions of zaps, as defined in Definition 5, are based on constructions of non-interactive zero knowledge proofs or arguments in the common reference string model. We review the requirements of such proof systems below.

Definition 11 (NIZK Proofs and Arguments). *Let $R_L = \{(x, w)\}$ be a witness-relation corresponding to a language $L \in \mathbf{NP}$. A* non-interactive zero-knowledge proof (or argument) system *for R_L consists of a triple of algorithms (Gen, P, V) where:*

- *The generator* Gen *is a PPT that takes as input a security parameter k and statement length $t = \mathrm{poly}(k)$, and produces a common reference string* crs*. An important special case is where $\mathrm{Gen}(1^k, 1^t)$ outputs a uniformly random string, in which case we say the proof (or argument) system operates in the common* random *string model.*
- *The prover P is a PPT that takes as input a* crs *and a pair (x, w) and outputs a proof π.*
- *The verifier V is an efficient, deterministic algorithm that takes as input a* crs*, an instance x and proof π, and outputs a bit in $\{0, 1\}$.*

Various security requirements we can impose on the proof system are:

Perfect completeness. *An honest prover who possesses a valid witness can always convince an honest verifier. Formally, for all $(x, w) \in R_L$,*

$$\Pr_{\substack{\mathrm{crs} \leftarrow \mathrm{Gen}(1^k, 1^{|x|}) \\ \pi \leftarrow P(\mathrm{crs}, x, w)}} [V(\mathrm{crs}, x, \pi) = 1] = 1.$$

Statistical soundness. *It is statistically impossible to convince an honest verifier of the validity of a false statement. There exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for every sequence $\{x_k\}_{k \in \mathbb{N}}$ of $\mathrm{poly}(k)$-size statements $x_k \notin L$,*

$$\Pr_{\mathrm{crs} \leftarrow \mathrm{Gen}(1^k, 1^{|x_k|})} [\exists \pi \in \{0,1\}^* \text{ s.t. } V(\mathrm{crs}, x_k, \pi) = 1] \leq \mathrm{negl}(k).$$

Computational zero-knowledge. *Proofs do not reveal anything to the verifier beyond their validity. Formally, a proof system is* computational zero-knowledge *if there exists a PPT simulator* (S_1, S_2) *where* S_1 *produces a simulated common reference string* crs *with associated trapdoor* τ*. The pair* (crs, τ) *allows* S_2 *to simulate accepting proofs without knowledge of a witness* w*. That is, there exists a negligible function* negl *such that for all (possibly cheating) PPT verifiers* V^* *and sequences* $\{(x_k, w_k)\}_{k\in\mathbb{N}}$ *of* $\text{poly}(k)$*-size statement-witness pairs* $(x_k, w_k) \in R_L$*,*

$$\left| \Pr_{\substack{\text{crs}\leftarrow\text{Gen}(1^k,1^{|x_k|})\\ \pi\leftarrow P(\text{crs},x_k,w_k)}} [V^*(\text{crs}, x_k, \pi) = 1] - \Pr_{\substack{(\text{crs},\tau)\leftarrow S_1(1^k,1^{|x_k|})\\ \pi\leftarrow S_2(\text{crs},\tau,x_k)}} [V^*(\text{crs}, x_k, \pi) = 1] \right| \leq \text{negl}(k).$$

Statistical knowledge extraction. *A proof system is additionally a* proof of knowledge *if a witness can be extracted from a valid proof. That is, there exists a polynomial-time knowledge extractor* $E = (E_1, E_2)$ *such that* E_1 *produces a simulated common reference string* crs *with associated extraction key* ξ*, which we assume to have length* $O(k)$*.*[1] *The pair* (crs, ξ) *allows the deterministic algorithm* E_2 *to extract a witness from a proof. Formally, the first component of* $(\text{crs}, \xi) \leftarrow E_1(1^k, 1^{|x|})$ *is identically distributed to* $\text{crs} \leftarrow \text{Gen}(1^k, 1^{|x|})$*. Moreover, there exists a negligible function* negl *such that for every* $x \in \{0,1\}^{\text{poly}(k)}$*,*

$$\Pr_{\text{crs}\leftarrow\text{Gen}(1^k,1^{|x|})} \Big[\exists \xi \in \{0,1\}^*, \pi \in \{0,1\}^*, w \in E_2(\text{crs}, \xi, x, \pi) : (\text{crs}, \xi) \in E_1(1^k, 1^{|x|}) \ \wedge \ (x, w) \notin R_L \ \wedge \ V(1^k, x, \pi) = 1 \Big] \leq \text{negl}(k).$$

For technical reasons, we also require that the relation $\{(\text{crs}, \xi) \in E_1(1^k, 1^{|x|})\}$ *be recognizable in polynomial-time, which will always be the case for our constructions.*

B.2 Extractability of Zaps Based on Exponentially Extractable NIZKs

We next describe Dwork and Naor's original construction of zaps [DN07]. Here, we show that extractable zaps can be based on the existence of NIZK proofs of knowledge in the common *random* string model, which can in turn be built from various number theoretic assumptions [DP92, DDP00, GOS12]. (Recall that in

[1] Such a constraint which depends only on the security parameter k will be important for meeting our definition of exponentially extractable zaps.

the common random string model for NIZK proofs, the crs generation algorithm simply outputs a uniformly random string.) The discussion in this section can be summarized by the following theorem.

Theorem 6. *Let R_L be a witness relation for a language $L \in \mathbf{NP}$. Then R_L has an extractable zap proof system if:*

> *There exists a non-interactive zero-knowledge proof of knowledge for R_L (in the common random string model) with perfect completeness, statistical soundness, computational zero-knowledge, and statistical extractability.*

The existence of such proofs of knowledge for **NP** can be based on any of the following assumptions:

1. The existence of NIZK proofs of membership for **NP** and "dense secure public-key encryption schemes" [DP92]. NIZK proofs of membership can in turn be constructed from trapdoor permutations [FLS99] or indistinguishability obfuscation and one-way functions [BP15]. Dense secure public-key encryption schemes can be constructed under the hardness of factoring Blum integers [DDP00] or the Diffie-Hellman assumption [DP92].
2. The decisional linear assumption for groups equipped with a bilinear map [GOS12].

The remainder of this section is devoted to the proof of Theorem 6. Let R_L be a witness relation for a language $L \in \mathbf{NP}$. Let $(P_{\text{NIZK}}, V_{\text{NIZK}})$ be a NIZK proof system in the common random string model. We now describe Dwork and Naor's [DN07] zap proof system for R_L based on $(P_{\text{NIZK}}, V_{\text{NIZK}})$.

For simplicity, assume we are interested in proving statements x having length which is a fixed polynomial in k. Let $\ell = \ell(k)$ be a fixed polynomial. (This depends on the length of x and on the soundness error of the NIZK proof system. We defer discussion of its value to the proof of Proposition 6, where it will also depend on the knowledge error of the NIZK knowledge extractor E_2.) The verifier's first message is a string $\rho \in \{0,1\}^{\ell \cdot m}$, which should be interpreted as a sequence of random strings $\rho_1, \ldots, \rho_\ell$ each in $\{0,1\}^m$. Here, $m = \text{poly}(k)$ is the length of the crs used in the proof system $(P_{\text{NIZK}}, V_{\text{NIZK}})$. The prover and verifier algorithms appear as Algorithms 4 and 5 respectively.

Algorithm 4. Zap Prover $P(1^k, x, w, \rho)$

Input: Security parameter k, instance x, witness w such that $(x, w) \in R_L$, first message ρ

Output: Proof π

1. Choose a random m-bit string $b \in \{0,1\}^m$. For each $j = 1, \ldots, \ell$, let $\text{crs}_j = b \oplus \rho_j$ be the bitwise exclusive-OR of b with ρ_j
2. For each $j = 1, \ldots, \ell$, let $\pi_j \leftarrow P_{\text{NIZK}}(\text{crs}_j, x, w)$
3. Send the verifier $\pi = (b, \pi_1, \ldots, \pi_\ell)$

Algorithm 5. Zap Verifier $V(1^k, x, \rho, \pi)$

Input: Security parameter k, instance x, first message ρ, proof $\pi = (b, \pi_1, \ldots, \pi_\ell)$
Output: Accept or reject decision

1. Let $\text{crs}_j = b \oplus \rho_j$ for each $j = 1, \ldots, \ell$
2. Accept iff $V_{\text{NIZK}}(\text{crs}_j, x, \pi) = 1$ for all $j = 1, \ldots, \ell$

Theorem 7 ([DN07]). *Suppose $(P_{\text{NIZK}}, V_{\text{NIZK}})$ is a perfectly complete and statistically sound NIZK proof system for R_L in the common random string model. Then (P, V) is a perfectly complete, statistically sound zap proof system for R_L.*

Our goal now is to show that if $(P_{\text{NIZK}}, V_{\text{NIZK}})$ is also a statistically sound *proof of knowledge*, then the zap proof system (P, V) is extractable in the sense of Definition 6.

Proposition 6. *If, in addition, $(P_{\text{NIZK}}, V_{\text{NIZK}})$ is statistically knowledge extractable, then (P, V) is also an extractable zap for R_L.*

Proof (Proof). Consider the extraction Algorithm 6.

Algorithm 6. Zap Extractor $E(1^k, x, \rho, \pi)$

Input: Security parameter k, instance x, first message ρ, proof $\pi = (b, \pi_1, \ldots, \pi_\ell)$
Output: Witness w

For each $j = 1, \ldots, \ell$:

1. Via brute force, identify (and verify) an extraction key ξ_j corresponding to a common random string $\text{crs}_j = b \oplus \rho_j$
2. Run the NIZK knowledge extractor $E_2(\text{crs}_j, \xi_j, x, \pi_j)$ to obtain a witness w
3. If $(x, w) \in R_L$, halt and output w

Let $x \in \{0,1\}^*$. We say a common random string $\text{crs} \in \{0,1\}^k$ is *knowledge-sound* for x if there does *not* exist a pair (π, ξ) such that

1. $V_{\text{NIZK}}(\text{crs}, x, \pi) = 1$,
2. (crs, ξ) is in the support of $E_1(1^k, 1^{|x|})$, and
3. $(x, w) \notin R_L$ for $w \leftarrow E_2(\text{crs}, \xi, x, \pi)$.

Lemma 7. *There exists a polynomial $\ell(k)$ for which the following holds. Let $x \in \{0,1\}^{\text{poly}(k)}$ and let $\rho_1, \ldots, \rho_\ell$ be random m-bit strings. Then with overwhelming probability over the choice of ρ, for every $b \in \{0,1\}^m$, there exists an index j for which $\text{crs}_j = b \oplus \rho_j$ is knowledge-sound for x.*

Proof. Let $q(k)$ denote the knowledge error of the NIZK proof system, i.e.

$$q(k) = \Pr_{\mathrm{crs} \leftarrow \mathrm{Gen}(1^k, 1^{|x|})} [\exists \xi, \pi : (\mathrm{crs}, \xi) \in E_1(1^k, 1^{|x|}) \\ \wedge (x, E_2(\mathrm{crs}, \xi, x, \pi)) \notin R_L \wedge V_{\mathrm{NIZK}}(\mathrm{crs}, x, \pi) = 1].$$

Statistical extractability of the NIZK proof system requires that $q(k) = \mathrm{negl}(k)$ for any $|x| = \mathrm{poly}(k)$. For any fixed b, the strings $\mathrm{crs}_j = b \oplus \rho_j$ are independent and uniformly random. Therefore, the probability that all ℓ copies fail to be knowledge-sound for x is at most q^ℓ. The number of possible assignments to $b \in \{0,1\}^m$ is 2^m. Therefore, it suffices to take $\ell = 2m$ to make $2^m q^\ell < \mathrm{negl}(k)$.

We may now complete the proof of Proposition 6.

By Lemma 7, with overwhelming probability over the choice of ρ, there exists an index j for which $\mathrm{crs}_j = b \oplus \rho_j$ is knowledge-sound for x. If the zap verifier V accepts, then in particular, $V_{\mathrm{NIZK}}(\mathrm{crs}_j, x, \pi) = 1$. Thus, the zap knowledge extractor $E_2(\mathrm{crs}_j, \xi_j, x, \pi_j)$ recovers a valid witness w for x. Since the number of strings crs_j that need to be checked is polynomial in k, and each extraction key has length $O(k)$, the extractor runs in time $2^{O(k)}$.

References

[ADR02] An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_6

[BNO08] Beimel, A., Nissim, K., Omri, E.: Distributed private data analysis: simultaneously solving how and what. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 451–468. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_25

[BP15] Bitansky, N., Paneth, O.: ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 401–427. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46497-7_16

[BV16] Balcer, V., Vadhan, S.: Efficient algorithms for differentially private histograms with worst-case accuracy over large domains (2016). Manuscript

[BZ14] Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 480–499. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44371-2_27

[BZ16] Bun, M., Zhandry, M.: Order-revealing encryption and the hardness of private learning. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A. LNCS, vol. 9562, pp. 176–206. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49096-9_8

[CGGM00] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge. In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, pp. 235–244. ACM (2000)

[CSS12] Chan, T.-H.H., Shi, E., Song, D.: Privacy-preserving stream aggregation with fault tolerance. In: Keromytis, A.D. (ed.) FC 2012. LNCS, vol. 7397, pp. 200–214. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32946-3_15

[DDP00] Santis, A., Crescenzo, G., Persiano, G.: Necessary and sufficient assumptions for non-interactive zero-knowledge proofs of knowledge for all NP relations. In: Montanari, U., Rolim, J.D.P., Welzl, E. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 451–462. Springer, Heidelberg (2000). doi:10.1007/3-540-45022-X_38

[DKM+06] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006). doi:10.1007/11761679_29

[DL09] Dwork, C., Lei, J.: Differential privacy and robust statistics. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, 31 May–2 June 2009, pp. 371–380 (2009)

[DMNS06] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). doi:10.1007/11681878_14

[DN07] Dwork, C., Naor, M.: Zaps, their applications. SIAM J. Comput. **36**(6), 1513–1543 (2007). Preliminary version in FOCS 2000

[DNR+09] Dwork, C., Naor, M., Reingold, O., Rothblum, G.N., Vadhan, S.P.: On the complexity of differentially private data release: efficient algorithms and hardness results. In: STOC, pp. 381–390 (2009)

[DP92] De Santis, A., Persiano, G.: Zero-knowledge proofs of knowledge without interaction (extended abstract). In: 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, 24–27 October 1992, pp. 427–436 (1992)

[DR14] Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)

[FLS99] Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. SIAM J. Comput. **29**(1), 1–28 (1999)

[FS90] Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, STOC 1990, pp. 416–426. ACM, New York (1990)

[GKL03] Gupta, A., Krauthgamer, R., Lee, J.R.: Bounded geometries, fractals, and low-distortion embeddings. In: Proceedings of 44th Symposium on Foundations of Computer Science (FOCS 2003), 11–14 October 2003, Cambridge, pp. 534–543 (2003)

[GKM+16] Goyal, V., Khurana, D., Mironov, I., Pandey, O., Sahai, A.: Do distributed differentially-private protocols require oblivious transfer? In: 43rd International Colloquium Automata, Languages, and Programming, ICALp 2016, Rome, 12–15 July 2016, Proceedings, Part I (2016, to appear)

[GKY11] Groce, A., Katz, J., Yerukhimovich, A.: Limits of computational differential privacy in the client/server setting. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 417–431. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19571-6_25

[GMPS13] Goyal, V., Mironov, I., Pandey, O., Sahai, A.: Accuracy-privacy tradeoffs for two-party differentially private protocols. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 298–315. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40041-4_17

[Gol04] Goldreich, O.: Foundations of Cryptography: Basic Applications. Cambridge University Press, Cambridge (2004)

[GOS12] Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. J. ACM (JACM) **59**(3), 11 (2012)

[HOZ13] Haitner, I., Omri, E., Zarosim, H.: Limits on the usefulness of random oracles. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 437–456. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36594-2_25

[KK05] Katz, J., Koo, C.-Y.: On constructing universal one-way hash functions from arbitrary one-way functions. IACR Cryptology ePrint Archive 2005:328 (2005)

[KLN+11] Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.D.: What can we learn privately? SIAM J. Comput. **40**(3), 793–826 (2011)

[KMS14] Khurana, D., Maji, H.K., Sahai, A.: Black-box separations for differentially private protocols. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 386–405. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45608-8_21

[MMP+10] McGregor, A., Mironov, I., Pitassi, T., Reingold, O., Talwar, K., Vadhan, S.: The limits of two-party differential privacy. In: 2010 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 81–90. IEEE (2010)

[MPRV09] Mironov, I., Pandey, O., Reingold, O., Vadhan, S.: Computational differential privacy. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 126–142. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03356-8_8

[NY89] Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, STOC 1989, pp. 33–43. ACM, New York (1989)

[Rom90] Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, STOC 1990, pp. 387–394. ACM, New York (1990)

[TS13] Thakurta, A., Smith, A.D.: Differentially private feature selection via stability arguments, and the robustness of the Lasso. In: The 26th Annual Conference on Learning Theory. COLT 2013, 12–14 June 2013, Princeton University, pp. 819–850 (2013)

[Ull13] Ullman, J.: Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, pp. 361–370. ACM (2013)

[UV11] Ullman, J., Vadhan, S.: PCPs and the hardness of generating private synthetic data. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 400–416. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19571-6_24

[Vad16] Vadhan, S.: The complexity of differential privacy (2016). http://privacytools.seas.harvard.edu/publications/complexity-differential-privacy