PCPs and the Hardness of Generating Private Synthetic Data*

Jonathan Ullman** Salil Vadhan***

School of Engineering and Applied Sciences &
Center for Research on Computation and Society
Harvard University, Cambridge, MA
{jullman,salil}@seas.harvard.edu

Abstract. Assuming the existence of one-way functions, we show that there is no polynomial-time, differentially private algorithm $\mathcal{A}$ that takes a database $D \in (\{0,1\}^d)^n$ and outputs a "synthetic database" $\widehat{D}$ all of whose two-way marginals are approximately equal to those of D. (A two-way marginal is the fraction of database rows $x \in \{0,1\}^d$ with a given pair of values in a given pair of columns.) This answers a question of Barak et al. (PODS '07), who gave an algorithm running in time $\mathrm{poly}(n, 2^d)$.

Our proof combines a construction of hard-to-sanitize databases based on digital signatures (by Dwork et al., STOC '09) with encodings based on probabilistically checkable proofs.

We also present both negative and positive results for generating "relaxed" synthetic data, where the fraction of rows in D satisfying a predicate c are estimated by applying c to each row of $\widehat{D}$ and aggregating the results in some way.

Keywords: privacy, digital signatures, inapproximability, constraint satisfaction problems, probabilistically checkable proofs

1 Introduction

There are many settings in which it is desirable to share information about a database that contains sensitive information about individuals. For example, doctors may want to share information about health records with medical researchers, the federal government may want to release census data for public information, and a company like Netflix may want to provide its movie rental database for a public competition to develop a better recommendation system. However, it is important to do this in way that preserves the "privacy" of the individuals whose records are in the database. This privacy problem has been studied by statisticians and the database security community for a number of years (cf., [1,8,15]), and recently the theoretical computer science community has developed an appealing new approach to the problem, known as *differential privacy*. (See the surveys [10,9].).

* A full version of this paper appears on ECCC [28]

** http://seas.harvard.edu/~jullman. Supported by NSF grant CNS-0831289.

*** http://seas.harvard.edu/~salil. Supported by NSF grant CNS-0831289.

Differential Privacy. A randomized algorithm $\mathcal{A}$ is defined to be *differentially private* [11] if for every two databases $D = (x_1, \ldots, x_n)$, $D' = (x'_1, \ldots, x'_n)$ that differ on exactly one row, the distributions $\mathcal{A}(D)$ and $\mathcal{A}(D')$ are "close" to each other. Formally, we require that $\mathcal{A}(D)$ and $\mathcal{A}(D')$ assign the same probability mass to every event, up to a multiplicative factor of $e^\epsilon \approx 1 + \epsilon$, where ϵ is typically taken to be a small constant. (In addition to this multiplicative factor, it is often allowed to also let the probabilities to differ by a negligible additive term.) This captures the idea that no individual's data has a significant influence on the output of $\mathcal{A}$ (provided that data about an individual is confined to one or a few rows of the database). Differential privacy has several nice properties lacking in previous notions, such as being agnostic to the adversary's prior information and degrading smoothly under composition.

With this model of privacy, the goal becomes to design algorithms $\mathcal{A}$ that simultaneously meet the above privacy guarantee and give "useful" information about the database. For example, we may have a true query function c in which we're interested, and the goal is to design $\mathcal{A}$ that is differentially private (with ϵ as small as possible) and estimates c well (e.g. the error $|\mathcal{A}(D) - c(D)|$ is small with high probability). For example, if $c(D)$ is the fraction of database rows that satisfy some property — a *counting query* — then it is known that we can take $\mathcal{A}(D)$ to equal $c(D)$ plus random Laplacian noise with standard deviation $O(1/(\epsilon n))$, where n is the number of rows in the database and ϵ is the measure of differential privacy [5]. A sequence of works [7,13,5,11] has provided a very good understanding of differential privacy in an interactive model in which real-valued queries c are made and answered one at a time. The amount of noise that one needs when responding to a query c should be based on the sensitivity of c, as well as the total number of queries answered so far.

However, for many applications, it would be more attractive to do a noninteractive data release, where we compute and release a single, differentially private "summary" of the database that enables others to determine accurate answers to a large class of queries. What form should this summary take? The most appealing form would be a *synthetic database*, which is a new database $\widehat{D} = \mathcal{A}(D)$ whose rows are "fake", but come from the same universe as those of D and are guaranteed to share many statistics with those of D (up to some accuracy). Some advantages of synthetic data are that it can be easily understood by humans, and statistical software can be run directly on it without modification. For example, these considerations led the German Institute for Employment Research to adopt synthetic databases for the release of employment statistics [25].

Previous Results on Synthetic Data. The first result on producing differentially private synthetic data came in the work of Barak et al. [3]. Given a database D consisting of n rows from $\{0,1\}^d$, they show how to construct a differentially private synthetic database $\widehat{D}$, also of n rows from $\{0,1\}^d$, in which the full "contingency table," consisting of all conjunctive counting queries, is approximately preserved. That is, for every conjunction $c(x_1, \ldots, x_n) = x_{i_1} \wedge x_{i_2} \wedge \cdots \wedge x_{i_k}$ for $i_1, \ldots, i_k \in [d]$, the fraction of rows in $\widehat{D}$ that satisfy c equals the fraction of rows in D that satisfy c up to an additive error of $2^{O(d)}/n$. The running time of their algorithm is $\mathrm{poly}(n, 2^d)$, which is feasible for small values of d. They pose as an open problem whether the running time of their algorithm can be improved for the case where we only want to preserve the k-*way*

marginals for small k (e.g. $k = 2$). These are the counting queries corresponding to conjunctions of up to k literals. Indeed, there are only $O(d)^k$ such conjunctions, and we can produce differentially private estimates for all the corresponding counting queries in time $\mathrm{poly}(n, d^k)$ by just adding noise $O(d)^k/n$ to each one. Moreover, a version of the Barak et al. algorithm [3] can ensure that even these noisy answers are consistent with a real database.[1]

A more general and dramatic illustration of the potential expressiveness of synthetic data came in the work of Blum, Ligett, and Roth [6]. They show that for every class $\mathcal{C} = \{c : \{0,1\}^d \to \{0,1\}\}$ of predicates, there is a differentially private algorithm A that produces a synthetic database $\hat{D} = \mathcal{A}(D)$ such that all counting queries corresponding to predicates in $\mathcal{C}$ are preserved to within an accuracy of $\tilde{O}((d \log(|\mathcal{C}|)/n)^{1/3})$, with high probability. In particular, with $n = \mathrm{poly}(d)$, the synthetic data can provide simultaneous accuracy for an *exponential-sized* family of queries (e.g. $|\mathcal{C}| = 2^d$). Unfortunately, the running time of the BLR mechanism is also exponential in d.

Dwork et al. [12] gave evidence that the large running time of the BLR mechanism is inherent. Specifically, assuming the existence of one-way functions, they exhibit an efficiently computable family $\mathcal{C}$ of predicates (e.g. consisting of circuits of size d^2) for which it is infeasible to produce a differentially private synthetic database preserving the counting queries corresponding to $\mathcal{C}$ (for databases of any $n = \mathrm{poly}(d)$ number of rows). For non-synthetic data, they show a close connection between the infeasibility of producing a differentially private summarization and the existence of efficient "traitor-tracing schemes." However, these results leave open the possibility that for *natural* families of counting queries (e.g. those corresponding to conjunctions), producing a differentially private synthetic database (or non-synthetic summarization) can be done efficiently. Indeed, one may have gained optimism by analogy with the early days of computational learning theory, where one-way functions were used to show hardness of learning arbitrary efficiently computable concepts in computational learning theory but natural subclasses (like conjunctions) were found to be learnable [29].

Our Results. We prove that it is infeasible to produce synthetic databases preserving even very simple counting queries, such as 2-way marginals:

Theorem 1. *Assuming the existence of one-way functions, there is a constant $\gamma > 0$ such that for every polynomial p, there is no polynomial-time, differentially private algorithm A that takes a database $D \in (\{0,1\}^d)^{p(d)}$ and produces a synthetic database $\hat{D} \in (\{0,1\}^d)^*$ such that $|c(D) - c(\hat{D})| \leq \gamma$ for all 2-way marginals c.*

(Recall that a 2-way marginal $c(D)$ computes the fraction of database rows satisfying a conjunction of two literals, i.e. the fraction of rows $x_i \in \{0,1\}^d$ such that $x_i(j) = b$ and $x_i(j') = b'$ for some columns $j, j' \in [d]$ and values $b, b' \in \{0,1\}$.) In fact, our impossibility result extends from conjunctions of 2 literals to any family of constant arity predicates that contains a function depending on at least two variables, such as parities of 3 literals.

[1] Technically, this "real database" may assign fractional weight to some rows.

As mentioned earlier, all 2-way marginals *can* be easily summarized with non-synthetic data (by just adding noise to each of the $(2d)^2$ values). Thus, our result shows that requiring a synthetic database may severely constrain what sorts of differentially private data releases are possible. (Dwork et al. [12] also showed that there exists a $\mathrm{poly}(d)$-sized family of counting queries that are hard to summarize with synthetic data, thereby separating synthetic data from non-synthetic data. Our contribution is to show that such a separation holds for a very simple and natural family of predicates, namely 2-way marginals.)

This separation between synthetic data and non-synthetic data seems analogous to the separations between proper and improper learning in computational learning theory [24,16], where it is infeasible to learn certain concept classes if the output hypothesis is constrained to come from the same representation class as the concept, but it becomes feasible if we allow the output hypothesis to come from a different representation class. This gives hope for designing efficient, differentially private algorithms that take a database and produce a compact summary of it that is not synthetic data but somehow can be used to accurately answer exponentially many questions about the original database (e.g. all marginals). The negative results of [12] on non-synthetic data (assuming the existence of efficient traitor-tracing schemes) do not say anything about natural classes of counting queries, such as marginals.

To bypass the complexity barrier stated in Theorem 1, it may not be necessary to introduce exotic data representations; some mild generalizations of synthetic data may suffice. For example, several recent algorithms [6,27,14] produce several synthetic databases, with the guarantee that the *median* answer over these databases is approximately accurate. More generally, we can consider summarizations of a database D that that consist of a collection $\hat{D}$ of rows from the same universe as the original database, and where we estimate $c(D)$ by applying the predicate c to each row of $\hat{D}$ and then aggregating the results via some aggregation function f. With standard synthetic data, f is simply the average, but we may instead allow f to take a median of averages, or apply an affine shift to the average. For such *relaxed synthetic data*, we prove the following results:

- There is a constant k such that counting queries corresponding to k-juntas (functions depending on at most k variables) cannot be accurately and privately summarized as relaxed synthetic data with a median-of-averages aggregator, or with a symmetric and monotone aggregator (that is independent of the predicate c being queried).
- For every constant k, counting queries corresponding to k-juntas *can* be accurately and privately summarized as relaxed synthetic data with an aggregator that applies an affine shift to the average (where the shift does depend on the predicate being queried).

Techniques. Our proof of Theorem 1 and our other negative results are obtained by combining the hard-to-sanitize databases of Dwork et al. [12] with PCP reductions. They construct a database consisting of valid message-signature pairs (m_i, σ_i) under a digital signature scheme, and argue that any differentially private sanitizer that preserves accuracy for the counting query associated with the signature verification predicate can

be used to forge valid signatures. We replace each message-signature pair (m_i, σ_i) with a PCP encoding π_i that proves that (m_i, σ_i) satisfies the signature verification algorithm. We then argue that if accuracy is preserved for a large fraction of the (constant arity) constraints of the PCP verifier, then we can "decode" the PCP either to violate privacy (by recovering one of the original message-signature pairs) or to forge a signature (by producing a new message-signature pair).

We remark that error-correcting codes were already used in [12] for the purpose of producing a fixed polynomial-sized set of counting queries that can be used for all verification keys. Our observation is that by using *PCP* encodings, we can reduce not only the number of counting queries in consideration, but also their computational complexity.

Our proof has some unusual features among PCP-based hardness results:

- As far as we know, this is the first time that PCPs have been used in conjunction with cryptographic assumptions for a hardness result. (They have been used together for positive results regarding computationally sound proof systems [21,22,4].) It would be interesting to see if such a combination could be useful in, say, computational learning theory (where PCPs have been used for hardness of "proper" learning [2,17] and cryptographic assumptions for hardness of representation-independent learning [29,20]).
- While PCP-based inapproximability results are usually stated as Karp reductions, we actually need them to be *Levin* reductions — capturing that they are reductions between search problems, and not just decision problems. (Previously, this property has been used in the same results on computationally sound proofs mentioned above.)

2 Preliminaries

2.1 Sanitizers

Let a *database* $D \in (\{0,1\}^d)^n$ be a matrix of n rows, $x_1, \ldots, x_n$, corresponding to people, each of which contains d binary attributes. A *sanitizer* $\mathcal{A} : (\{0,1\}^d)^n \to \mathcal{R}$ takes a database and outputs some data structure in $\mathcal{R}$. In the case where $\mathcal{R} = (\{0,1\}^d)^{\hat{n}}$ (an $\hat{n}$-row database) we say that $\mathcal{A}$ outputs a *synthetic database*.

We would like such sanitizers to be both *private* and *accurate*. In particular, the notion of privacy we are interested in is as follows

Definition 2 (Differential Privacy). *[11] A sanitizer $\mathcal{A} : (\{0,1\}^d)^n \to \mathcal{R}$ is* (ϵ, δ)-differentially private *if for every two databases $D_1, D_2 \in (\{0,1\}^d)^n$ that differ on exactly one row, and every subset $S \subseteq \mathcal{R}$*

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(D_2) \in S] + \delta$$

In the case where $\delta = 0$ we say that $\mathcal{A}$ is ϵ-differentially private.

Since a sanitizer that always outputs 0 satisfies Definition 2, we also need to define what it means for a database to be accurate. In this paper we consider accuracy

with respect to counting queries. Consider a set $\mathcal{C}$ consisting of boolean predicates $c : \{0,1\}^d \to \{0,1\}$, which we call a *concept class*. Then each predicate c induces a *counting query* that on database $D = (x_1, \ldots, x_n) \in (\{0,1\}^d)^n$ returns

$$c(D) = \frac{1}{n} \sum_{i=1}^{n} c(x_i)$$

If the output of $\mathcal{A}$ is a synthetic database $\widehat{D} \in (\{0,1\}^d)^*$, then $c(\mathcal{A}(D))$ is simply the fraction of rows of $\widehat{D}$ that satisfy the predicate c. However, if $\mathcal{A}$ outputs a data structure that is not a synthetic database, then we require that there is an *evaluator* function $\mathcal{E} : \mathcal{R} \times \mathcal{C} \to \mathbb{R}$ that estimates $c(D)$ from the output of $\mathcal{A}(D)$ and the description of c. For example, $\mathcal{A}$ may output a vector $Z = (c(D) + Z_c)_{c \in \mathcal{C}}$ where Z_c is a random variable for each $c \in \mathcal{C}$, and $\mathcal{E}(Z, c)$ is the c-th component of $Z \in \mathcal{R} = \mathbb{R}^{|\mathcal{C}|}$. Abusing notation, we will write $c(\mathcal{A}(D))$ as shorthand for $\mathcal{E}(\mathcal{A}(D), c)$.

We will say that $\mathcal{A}$ is "accurate" for the concept class $\mathcal{C}$ if the estimates $c(\mathcal{A}(D))$ are close to the fractional counts $c(D)$. Formally

Definition 3 (Accuracy). *An output Z of sanitizer $\mathcal{A}(D)$ is* α-accurate *for a concept class $\mathcal{C}$ if*

$$\forall c \in \mathcal{C}, |c(Z) - c(D)| \leq \alpha.$$

A sanitizer $\mathcal{A}$ is (α, β)-accurate *for a concept class $\mathcal{C}$ if for every database D,*

$$\Pr_{\mathcal{A}'s \text{ coins}} [\forall c \in \mathcal{C},\ |c(\mathcal{A}(D)) - c(D)| \leq \alpha] \geq 1 - \beta$$

In this paper we say $f(n) = \mathrm{negl}(n)$ if $f(n) = o(n^{-c})$ for every $c > 0$ and say that $f(n)$ is *negligible*. We use $|s|$ to denote the length of the string s, and $s_1 \| s_2$ to denote the concatenation of s_1 and s_2.

2.2 Hardness of Sanitizing

Differential privacy is a very strong notion of privacy, so it is common to look for hardness results that also apply to weaker notions of privacy. These hardness results show that every sanitizer must be "blatantly non-private" in some sense. In this paper our notion of blatant non-privacy roughly states that there exists an efficient adversary who can find a row of the original database using only the output from any efficient sanitizer. Such definitions are also referred to as "row non-privacy." We define hardness-of-sanitization with respect to a particular concept class, and want to exhibit a distribution on databases for which it would be infeasible for any efficient sanitizer to give accurate output without revealing a row of the database. Specifically, following [12], we define the following notions

Definition 4 (Database Distribution Ensemble). *Let $\mathcal{D} = \mathcal{D}_d$ be an ensemble of distributions on d-column databases with $n+1$ rows $D \in (\{0,1\}^d)^{n+1}$. Let $(D, D', i) \leftarrow_R \tilde{\mathcal{D}}$ denote the experiment in which we choose $D_0 \leftarrow_R \mathcal{D}$ and $i \in [n]$ uniformly at random, and set D to be the first n rows of D_0 and D' to be D with the i-th row replaced by the $(n+1)$-st row of D_0.*

Definition 5 (Hard-to-sanitize Distribution). *Let $\mathcal{C}$ be a concept class, $\alpha \in [0,1]$ be a parameter, and $\mathcal{D} = \mathcal{D}_d$ be a database distribution ensemble.*

The distribution $\mathcal{D}$ is $(\alpha, \mathcal{C})$-hard-to-sanitize *if there exists an efficient adversary $\mathcal{T}$ such that for any alleged polynomial-time sanitizer $\mathcal{A}$ the following conditions hold:*

1. *Whenever $\mathcal{A}(D)$ is α-accurate, then $\mathcal{T}(\mathcal{A}(D))$ outputs a row of D:*

$$\Pr_{\substack{(D,D',i)\leftarrow_R \tilde{\mathcal{D}} \\ \mathcal{A}'s \text{ and } \mathcal{T}'s \text{ coins}}} [(\mathcal{A}(D) \text{ is } \alpha\text{-accurate for } \mathcal{C}) \wedge (\mathcal{T}(\mathcal{A}(D)) \cap D = \emptyset)] \leq \text{negl}(d).$$

2. *For every efficient sanitizer $\mathcal{A}$, $\mathcal{T}$ cannot extract x_i from the database D':*

$$\Pr_{\substack{(D,D',i)\leftarrow_R \tilde{\mathcal{D}} \\ \mathcal{A}'s \text{ and } \mathcal{T}'s \text{ coins}}} [\mathcal{T}(\mathcal{A}(D')) = x_i] \leq \text{negl}(d)$$

where x_i is the i-th row of D.

In [12], it was shown that every distribution that is $(\alpha, \mathcal{C})$-hard-to-sanitize in the sense of Definition 5, is also hard to sanitize while achieving even weak differential privacy

Claim 6. *[12] If a distribution ensemble $\mathcal{D} = \mathcal{D}_d$ on $n(d)$-row databases is $(\alpha, \mathcal{C})$-hard-to-sanitize, then for every constant $a > 0$ and every $\beta = \beta(d) \leq 1 - 1/\text{poly}(d)$, no efficient sanitizer that is (α, β)-accurate with respect to $\mathcal{C}$ can achieve $(a \log(n), (1 - 8\beta)/2n^{1+a})$-differential privacy.*

In particular, for all constants $\epsilon, \beta > 0$, no polynomial-time sanitizer can achieve (α, β)-accurateness and $(\epsilon, \text{negl}(n))$-differential privacy.

We could use a weaker definition of hard-to-sanitize distributions, which would still suffice to rule out differential privacy, that only requires that for every efficient $\mathcal{A}$, there exists an adversary $\mathcal{T}_{\mathcal{A}}$ that almost always extracts a row of D from every α-accurate output of $\mathcal{A}(D)$. In our definition we require that there exists a fixed adversary $\mathcal{T}$ that almost always extracts a row of D from every α-accurate output of any efficient $\mathcal{A}$. Reversing the quantifiers in this fashion only makes our negative results stronger.

In this paper we are concerned with sanitizers that output synthetic databases, so we will relax Definition 5 by restricting the quantification over sanitizers to only those sanitizers that output synthetic data.

Definition 7 (Hard-to-sanitize Distribution as Synthetic Data). *A database distribution ensemble $\mathcal{D}$ is* $(\alpha, \mathcal{C})$-hard-to-sanitize as synthetic data *if the conditions of Definition 5 hold for every sanitizer $\mathcal{A}$ that outputs a synthetic database.*

3 Relationship with Hardness of Approximation

The objective of a privacy-preserving sanitizer is to reveal some properties of the underlying database without giving away enough information to reconstruct that database. This requirement has different implications for sanitizers that produce synthetic databases and those with arbitrary output.

The SuLQ framework of [5] is a well-studied, efficient technique for achieving (ϵ, δ)-differential privacy, with non-synthetic output. To get accurate, private output for a family of counting queries with predicates in $\mathcal{C}$, we can release a vector of noisy counts $(c(D) + Z_c)_{c \in \mathcal{C}}$ where the random variables $(Z_c)_{c \in \mathcal{C}}$ are drawn independently from a distribution suitable for preserving privacy. (e.g. a Laplace distribution with standard deviation $O(|\mathcal{C}| / \epsilon n)$).

Consider the case of an n-row database D that contains satisfying assignments to a 3CNF formula φ, and suppose our concept class includes all disjunctions on three literals (or, equivalently, all conjunctions on three literals). Then the technique above releases a set of noisy counts that describes a database in which every clause of φ is satisfied by most of the rows of D. However, sanitizers that output accurate synthetic databases are required to produce a database that consists of rows that satisfy most of the clauses of φ.

Because of the noise added to the output, the requirement of a synthetic database does not strictly force the sanitizer to find a satisfying assignment for the given 3CNF. However, it is known to be NP-hard to find even approximate satisfying assignments for many constraint satisfaction problems. In our main result, Theorem 14, we will show that there exists a distribution over databases that is hard-to-sanitize with respect to synthetic data for any concept class that is sufficient to express a hard-to-approximate constraint satisfaction problem.

3.1 Hard to Approximate CSPs

We define a *constraint satisfaction problem* to be the following.

Definition 8 (Constraint Satisfaction Problem (CSP)). *For a function $q = q(d) \leq d$, a* family of $q(d)$-CSPs*, denoted $\Gamma = (\Gamma_d)_{d \in \mathbb{N}}$, is a sequence of sets Γ_d of boolean predicates on $q(d)$ variables. If $q(d)$ and Γ_d do not depend on d then we refer to Γ as a* fixed family of q-CSPs*.*

For every $d \geq q(d)$, let $\mathcal{C}_\Gamma^{(d)}$ be the class consisting of all predicates $c : \{0,1\}^d \to \mathbb{R}$ of the form $c(u_1, \ldots, u_d) = \gamma(u_{i_1}, \ldots, u_{i_{q(d)}})$ for some $\gamma \in \Gamma_d$ and $i_1, \ldots, i_{q(d)} \in [d]$. We call $\mathcal{C}_\Gamma = \cup_{d=0}^{\infty} \mathcal{C}_\Gamma^{(d)}$ the class of constraints of Γ*. Finally, we say a multiset $\varphi \subseteq \mathcal{C}_\Gamma^{(d)}$ is a d-*variable instance of $\mathcal{C}_\Gamma$ *and each $\varphi_i \in \varphi$ is a* constraint of φ*.*

We say that an assignment x satisfies *the constraint φ_i if $\varphi_i(u) = 1$. For $\varphi = \{\varphi_1, \ldots, \varphi_m\}$, define*

$$\mathrm{val}(\varphi, u) = \frac{\sum_{i=1}^{m} \varphi_i(u)}{m} \qquad \textit{and} \qquad \mathrm{val}(\varphi) = \max_{u \in \{0,1\}^d} \mathrm{val}(\varphi, u).$$

Our hardness results will apply to concept classes $\mathcal{C}_\Gamma^{(d)}$ for CSP families Γ with certain additional properties. Specifically we define,

Definition 9 (Nice CSP). *A family $\Gamma = (\Gamma_d)_{d \in \mathbb{N}}$ of $q(d)$-CSPs* nice *if*

1. *$q(d) = d^{1-\Omega(1)}$,*
2. *for every $d \in \mathbb{N}$, Γ_d contains a non-constant predicate $\varphi^* : \{0,1\}^{q(d)} \to \{0,1\}$. Moreover, φ^* and two assignments $u_0^*, u_1^* \in \{0,1\}^{q(d)}$ such that $\varphi^*(u_0) = 0$ and $\varphi^*(u_1) = 1$ can be found in time* $\mathrm{poly}(d)$.

We note that any fixed family of q-CSP that contains a non-constant predicate is a nice CSP. Indeed, these CSPs (e.g. conjunctions of 2 literals) are the main application of interest for our results. However it will sometimes be useful to work with generalizations to nice CSPs with predicates of non-constant arity.

For our hardness result, we will need to consider a strong notion of hard constraint satisfaction problems, which is related to probabilistically checkable proofs. First we recall the standard notion of hardness of approximation under Karp reductions. (stated for additive, rather than multiplicative approximation error)

Definition 10 (inapproximability under Karp reductions). *For functions $\alpha, \gamma : \mathbb{N} \to [0,1]$. A family of CSPs $\Gamma = (\Gamma_d)_{d\in\mathbb{N}}$ is* (α,γ)-hard-to-approximate under Karp reductions *if there exists a polynomial-time computable function R such that for every circuit C with input size $\overline{d}$, if we set $\varphi_C = R(C) \subseteq \mathcal{C}_\Gamma^{(d)}$ for some $d = \mathrm{poly}(\overline{d})$, then*

1. *if C is satisfiable, then $\mathrm{val}(\varphi_C) \geq \gamma(d)$, and*
2. *if C is unsatisfiable, then $\mathrm{val}(\varphi_C) < \gamma(d) - \alpha(d)$.*

For our hardness result, we will need a stronger notion of inapproximability, which says that we can efficiently transform satisfying assignments of C into solutions to φ_C of high value, and vice-versa.

Definition 11 (inapproximability under Levin reductions). *For functions $\alpha, \gamma : \mathbb{N} \to [0,1]$. A family of CSPs $\Gamma = (\Gamma_d)_{d\in\mathbb{N}}$ is* (α,γ)-hard-to-approximate under Levin reductions *if there exist polynomial-time computable functions R, Enc, Dec such that for every circuit C with input of size $\overline{d}$ if we set $\varphi_C = R(C) \subseteq \mathcal{C}_\Gamma^{(d)}$ for some $d = \mathrm{poly}(\overline{d})$, then*

1. *for every $u \in \{0,1\}^{\overline{d}}$ such that $C(u) = 1$, $\mathrm{val}(\varphi_C, Enc(u, C)) \geq \gamma(d)$,*
2. *and for every $\pi \in \{0,1\}^d$ such that $\mathrm{val}(\varphi_C, \pi) \geq \gamma(d) - \alpha(d)$, $C(Dec(\pi, C)) = 1$,*
3. *and for every $u \in \{0,1\}^{\overline{d}}$, $Dec(Enc(u, C)) = u$*

When we do not wish to specify the value γ we will simply say that the family Γ is α-hard-to-approximate under Levin reductions *to indicate that there exists such a $\gamma \in (\alpha, 1]$. If we drop the requirement that R is efficiently computable, then we say that Γ is* (α,γ)-hard-to-approximate under inefficient Levin reductions*.*

The notation Enc, Dec reflects the fact that we think of the set of assignments π such that $\mathrm{val}(\varphi_C, \pi) \geq \gamma$ as a sort of error-correcting code on the satisfying assignments to C. Any π' with value close to γ can be decoded to a valid satisfying assignment.

Levin reductions are a stronger notion of reduction than Karp reductions. To see this, let Γ be α-hard-to-approximate under Levin reductions, and let R, Enc, Dec be the functions described in Definition 11. We now argue that for every circuit C, the formula $\varphi_C = R(C)$ satisfies conditions 1 and 2 of Definition 10. Specifically, if there exists an assignment $u \in \{0,1\}^{\overline{d}}$ that satisfies C, then $Enc(u, C)$ satisfies at least a γ fraction of the constraints of φ_C. Conversely if any assignment $\pi \in \{0,1\}^d$ satisfies at least a $\gamma - \alpha$ fraction of the constraints of φ_C, then $Dec(\pi, C)$ is a satisfying assignment of C.

Variants of the PCP Theorem can be used to show that essentially every class of CSP is hard-to-approximate in this sense. We restrict to CSP's that are closed under complement as it suffices for our application.

Theorem 12 (variant of PCP Theorem). *For every fixed family of CSPs Γ that is closed under negation and contains a function that depends on at least two variables, there is a constant $\alpha = \alpha(\Gamma) > 0$ such that Γ is α-hard to approximate under Levin reductions.*

It seems likely that optimized PCP/inapproximability results (like [19]) are also Levin reductions, which would yield fairly large values for α for natural CSPs (e.g. $\alpha = 1/8 - \epsilon$ if Γ contains all conjunctions of 3-literals, because then $\mathcal{C}_\Gamma$ contains MAX 3-SAT.)

For some of our results we will need CSPs that are very hard to approximate (under possibly inefficient reductions), which we can obtain by "sequential repetition" of constant-error PCPs.

Theorem 13 (variant of PCP Theorem with subconstant error). *There is a constant C such that for every $\epsilon = \epsilon(d) > 2^{-\mathrm{poly}(d)}$, the constraint family $\Gamma = (\Gamma_d)_{d\in\mathbb{N}}$ of $k(d)$-clause 3-CNF formulas is $(1-\epsilon(d), 1)$-hard-to-approximate under inefficient Levin reductions, for $k(d) = C\log(1/\epsilon(d))$.*

Further discussion of these theorems can be found in the full version of this paper.

4 Hard-to-Sanitize Distributions from Hard CSPs

In this section we prove that to efficiently produce a synthetic database that is accurate for the constraints of a CSP that is hard-to-approximate under Levin reductions, we must pay constant error in the worst case. Following [12], we start with a digital signature scheme, and a database of valid message-signature pairs. There is a verifying circuit C_{vk} and valid message-signature pairs are satisfying assignments to that circuit. Now we encode each row of database using the function Enc, described in Definition 11, that maps satisfying assignments to C_{vk} to assignments of the CSP instance $\varphi_{C_{vk}} = R(C_{vk})$ with value at least γ. Then, any assignment to the CSP instance that satisfies a $\gamma - \alpha$ fraction of clauses can be decoded to a valid message-signature pair. The database of encoded message-signature pairs is what we will use as our hard-to-sanitize distribution.

4.1 Main Hardness Result

We are now ready to state and prove our hardness result. Let $\Gamma = (\Gamma_d)_{d\in\mathbb{N}}$ be a family of $q(d)$-CSPs and let $\mathcal{C}_\Gamma = \cup_{d=1}^{\infty}\mathcal{C}_\Gamma^{(d)}$ be the class of constraints of Γ, which was constructed in Definition 8. We now state our hardness result.

Theorem 14. *Let $\Gamma = (\Gamma_d)_{d\in\mathbb{N}}$ be a family of nice $q(d)$-CSPs such that $\Gamma_d \cup \neg\Gamma_d$ is $\alpha(d)$-hard-to-approximate under (possibly inefficient) Levin reductions for $\alpha = \alpha(d) \in (0, 1/2)$. Assuming the existence of one-way functions, for every polynomial $n(d)$, there exists a distribution ensemble $\mathcal{D} = \mathcal{D}_d$ on $n(d)$-row databases that is $(\alpha(d), \mathcal{C}_\Gamma^{(d)})$-hard-to-sanitize as synthetic data.*

Proof. Let $\Pi = (Gen, Sign, Ver)$ be a digital signature scheme where it is even hard to produce a new signature for a previously signed message[2]. and let Γ be a family of CSPs that is α-hard-to-approximate under Levin reductions. Let R, Enc, Dec be the polynomial-time functions and $\gamma = \gamma(d) \in (\alpha, 1]$ be the parameter from Definition 11. Let $\kappa = d^\tau$ for a constant $\tau > 0$ to be defined later.

Let $n = n(d) = \text{poly}(d)$. We define the database distribution ensemble $\mathcal{D} = \mathcal{D}_d$ to generate $n+1$ random message-signature pairs and then encode them as PCP witnesses with respect to the signature-verification algorithm. We also encode the verification key for the signature scheme using the non-constant constraint $\varphi^* : \{0,1\}^{q(d)} \to \{0,1\}$ in Γ_d and the assignments $u_0^*, u_1^* \in \{0,1\}^{q(d)}$ such that $\varphi^*(u_0^*) = 0$ and $\varphi^*(u_1^*) = 1$, as described in the definition of nice CSPs (Definition 9).

Database Distribution Ensemble $\mathcal{D} = \mathcal{D}_d$:

$(sk, vk) \leftarrow_{\text{R}} Gen(1^\kappa)$, let $vk = vk_1 vk_2 \dots vk_\ell$, where $\ell = |vk| = \text{poly}(\kappa)$
$(m_1, \dots, m_{n+1}) \leftarrow_{\text{R}} (\{0,1\}^\kappa)^{n+1}$
for $i = 1$ to $n+1$ **do**
  $x_i := Enc(m_i \| Sign_{sk}(m_i), C_{vk}) \| u^*_{vk_1} \| u^*_{vk_2} \| \dots \| u^*_{vk_\ell}$, padded with zeros to be of length exactly d
end for
return $D_0 := (x_1, \dots, x_{n+1})$

Recall that $s_1 \| s_2$ denotes the concatenation of the strings s_1 and s_2. Note that the length of x_i before padding is $\text{poly}(\kappa) + q(d)\text{poly}(\kappa) \le d^{1-\Omega(1)}\text{poly}(d^\tau)$, so we can choose the constant $\tau > 0$ to be small enough that the length of x before padding is at most d and the above is well defined.

Every valid pair $(m, Sign_{sk}(m))$ is a satisfying assignment of the circuit C_{vk}, hence every row of D_0 constructed in this way will satisfy at least a γ fraction of the clauses of the formula $\varphi_{C_{vk}} = R(C_{vk})$. Additionally, for every bit of the verification key, there is a block of $q(d)$ bits in each row that contains either a satisfying assignment or a non-satisfying assignment of φ^*, depending on whether that bit of the key is 1 or 0. Specifically, let $L = |Enc(m_i \| Sign_{sk}(m_i))|$ in the construction of D_0 and for $j = 1, 2, \dots, \ell$, let $\varphi_j^*(x) = \varphi^*(x_{L+(j-1)q+1}, x_{L+(j-1)q+2}, \dots, x_{L+jq})$. Then, by construction, $\varphi_j^*(D_0) = vk_j$, the j-th bit of the verification key. Note that $\varphi_j^* \in \mathcal{C}_\Gamma^{(d)}$ for $j = 1, 2, \dots, \ell$, by our construction of $\mathcal{C}_\Gamma^{(d)}$ (Definition 8).

We now prove the following two lemmas that will establish $\mathcal{D}$ is hard-to-sanitize:

Lemma 15. *There exists a polynomial-time adversary $\mathcal{T}$ such that for every polynomial-time sanitizer $\mathcal{A}$,*

$$\Pr_{\substack{(D, D', i) \leftarrow_R \tilde{\mathcal{D}} \\ \mathcal{A}'s \text{ and } \mathcal{T}'s \text{ coins}}} \left[(\mathcal{A}(D) \textit{ is } \alpha\textit{-accurate for } \mathcal{C}_\Gamma^{(d)}) \wedge (\mathcal{T}(\mathcal{A}(D)) \cap D = \emptyset) \right] \le \text{negl}(d) \tag{1}$$

[2] These digital signature schemes are defined formally in the full version of this paper. In [18] it is shown how to modify known constructions [23,26] to obtain a such a digital signature scheme from any one-way function.

Proof. Our privacy adversary tries to find a row of the original database by trying to PCP-decode each row of the "sanitized" database and then re-encoding it. In order to do so, the adversary needs to know the verification key used in the construction of the database, which it can discover from the answers to the queries φ_j^*, defined above. Formally, we define the privacy adversary by means of a subroutine that tries to learn the verification key and then PCP-decode each row of the input database:

Subroutine $\mathcal{K}(\widehat{D})$:

Let d be the dimension of rows in $\widehat{D}$, $\kappa = d^\tau$, $\ell = |vk| = \mathrm{poly}(\kappa)$.
for $j = 1$ to ℓ **do**
  $\widehat{vk}_j = \left[\varphi_j^*(\widehat{D}) \text{ rounded to } \{0,1\}\right]$
end for
return $\widehat{vk}_1 \| \widehat{vk}_2 \| \ldots \| \widehat{vk}_\ell$

Subroutine $\mathcal{T}_0(\widehat{D})$:

Let $\hat{n}$ be the number of rows in $\widehat{D}$, $\widehat{vk} = \mathcal{K}(\widehat{D})$
for $i = 1$ to $\hat{n}$ **do**
  if $C_{\widehat{vk}}(Dec(\hat{x}_i, C_{\widehat{vk}})) = 1$ **then**
    return $Dec(\hat{x}_i, C_{\widehat{vk}})$
  end if
end for
return $\perp$

Privacy Adversary $\mathcal{T}(\widehat{D})$:

Let $\widehat{vk} = \mathcal{K}(\widehat{D})$.
return $Enc(\mathcal{T}_0(\widehat{D}), C_{\widehat{vk}})$

Let $\mathcal{A}$ be a polynomial-time sanitizer, we will show that Inequality (1) holds.

Claim 16. *If $\widehat{D} = \mathcal{A}(D)$ is α-accurate for $\mathcal{C}_\Gamma^{(d)}$, then $\mathcal{T}_0(\widehat{D})$ outputs a pair (m, σ) s.t. $C_{vk}(m, \sigma) = 1$.*

Proof. First we argue that if $\widehat{D}$ is α-accurate for $\mathcal{C}_\Gamma^{(d)}$ for $\alpha < 1/2$, then $\mathcal{K}(\widehat{D}) = vk$, where vk is the verification key used in the construction of D_0. By construction, $\varphi_j^*(D) = vk_j$. If $vk_j = 0$ and $\widehat{D}$ is α-accurate for D then $\varphi_j^*(\widehat{D}) \leq \alpha < 1/2$, and $\widehat{vk}_j = vk_j$. Similarly, if $vk_j = 1$ then $\varphi_j^*(\widehat{D}) \geq 1 - \alpha > 1/2$, and $\widehat{vk}_j = vk_j$. Thus, for the rest of the proof we will be justified in substituting vk for $\widehat{vk}$.

Next we show that if $\widehat{D}$ is α-accurate, then $\mathcal{T}_0(\widehat{D}) \neq \perp$. It is sufficient to show there exists $\hat{x}_i \in \widehat{D}$ such that $\mathrm{val}(\varphi_{C_{vk}}, x_i) \geq \gamma - \alpha$, which implies $C_{vk}(Dec(\hat{x}_i, C_{vk})) = 1$.

Since every $(m_i, Sign_{sk}(m_i))$ pair is a satisfying assignment to C_{vk}, the definition of Enc (Definition 11) implies that each row x_i of D has $\mathrm{val}(\varphi_{C_{vk}}, x_i) \geq \gamma$. Thus if $\varphi_{C_{vk}} = \{\varphi_1, \ldots, \varphi_m\}$, then

$$\frac{1}{m}\sum_{j=1}^{m} \varphi_j(D) = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n}\varphi_j(x_i)\right) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{val}(\varphi_{C_{vk}}, x_i) \geq \gamma.$$

Since $\widehat{D}$ is α-accurate for $\mathcal{C}_\Gamma^{(d)}$, and for every constraint φ_j, either $\varphi_j \in \Gamma$ or $\neg\varphi_j \in \Gamma$, then for every constraint $\varphi_j \in \varphi_{C_{vk}}$, we have $\varphi_j(\widehat{D}) \geq \varphi_j(D) - \alpha$. Thus

$$\frac{1}{\hat{n}}\sum_{i=1}^{\hat{n}} \mathrm{val}(\varphi_{C_{vk}}, \hat{x}_i) = \frac{1}{m}\sum_{j=1}^{m} \varphi_j(\widehat{D}) \geq \frac{1}{m}\sum_{j=1}^{m} \varphi_j(D) - \alpha \geq \gamma - \alpha.$$

So for at least one row $\hat{x} \in \widehat{D}$ it must be the case that $\mathrm{val}(\varphi_{C_{vk}}, \hat{x}) \geq \gamma - \alpha$. The definition of Dec (Definition 11) implies $C_{vk}(Dec(\hat{x}, C_{vk})) = 1$.

Now notice that if $\mathcal{T}_0(\mathcal{A}(D))$ outputs a valid message-signature pair but $\mathcal{T}(\mathcal{A}(D)) \cap D = \emptyset$, then this means $\mathcal{T}_0(\mathcal{A}(D))$ is forging a new signature not among those used to generate D, violating the security of the digital signature scheme. Formally, we construct a signature forger as follows:

Forger $\mathcal{F}(vk)$ with oracle access to $Sign_{sk}$:

Use the oracle $Sign_{sk}$ to generate an n-row database D just as in the definition of $\mathcal{D}_d$ (consisting of PCP encodings of valid message-signature pairs and an encoding of vk).
Let $\widehat{D} := \mathcal{A}(D)$
return $\hat{x}^* := \mathcal{T}_0(\widehat{D})$

Notice that running $\mathcal{F}$ in a chosen-message attack is equivalent to running $\mathcal{T}$ in the experiment of inequality (1), except that $\mathcal{F}$ does not re-encode the output of $\mathcal{T}_0(\mathcal{A}(D))$. By the super-security of the signature scheme, if the $\hat{x}^*$ output by $\mathcal{F}$ is a valid message-signature pair (as holds if $\mathcal{A}(D)$ is α-accurate for $\mathcal{C}_\Gamma^{(d)}$, by Claim 16), then it must be one of the message-signature pairs used to construct D (except with probability $\mathrm{negl}(\kappa) = \mathrm{negl}(d)$). This implies that $\mathcal{T}(\mathcal{A}(D)) = Enc(\hat{x}^*, C_{vk}) \in D$ (except with negligible probability). Thus, we have

$$\Pr_{\substack{(D,D',i)\leftarrow_R \tilde{\mathcal{D}} \\ \mathcal{A}'s \text{ coins}}} [\mathcal{A}(D) \text{ is } \alpha\text{-accurate for } \mathcal{C}_\Gamma^{(d)} \Rightarrow \mathcal{T}(\mathcal{A}(D)) \in D] \geq 1 - \mathrm{negl}(d),$$

which is equivalent to the statement of the lemma.

Lemma 17.

$$\Pr_{\substack{(D,D',i)\leftarrow_R \tilde{\mathcal{D}} \\ \mathcal{A}'s \text{ and } \mathcal{T}'s \text{ coins}}} [\mathcal{T}(\mathcal{A}(D')) = x_i] \leq \mathrm{negl}(d)$$

Proof. Since the messages m_i used in D_0 are drawn independently, D' contains no information about the message m_i, thus no adversary can, on input $\mathcal{A}(D')$ output the target row x_i except with probability $2^{-\kappa} = \mathrm{negl}(d)$.

These two claims suffice to establish that $\mathcal{D}$ is $(\alpha, \mathcal{C}_\Gamma)$-hard-to-sanitize as synthetic data.

Theorem 1 in the introduction follows by combining Theorems 12 and 14.

5 Relaxed Synthetic Data

The proof of Theorem 14 requires that the sanitizer output a synthetic database. In this section we present similar hardness results for sanitizers that produce other forms of output, as long as they still produce a collection of elements from $\{0,1\}^d$, that are interpreted as the data of (possibly "fake") individuals. More specifically, we consider sanitizers that output a database $\widehat{D} \in (\{0,1\}^d)^{\hat{n}}$ but are evaluated by applying a predicate c to each row and then applying a function f to the resulting bits and the predicate c. For example, when the sanitizer outputs a synthetic database, we have $f(b_1,\ldots,b_{\hat{n}},c) = (1/\hat{n})\sum_{i=1}^{\hat{n}} b_i$, which is just the fraction of rows that get labeled with a 1 by the predicate c (independent of c).

We now give a formal definition of *relaxed synthetic data*

Definition 18 (Relaxed Synthetic Data). *A sanitizer $\mathcal{A} : (\{0,1\}^d)^n \to (\{0,1\}^d)^{\hat{n}}$ with evaluator $\mathcal{E}$ outputs* relaxed synthetic data *for a family of predicates $\mathcal{C}$ if there exists $f : \{0,1\}^{\hat{n}} \times \mathcal{C} \to [0,1]$ such that for every $c \in \mathcal{C}$*

$$\mathcal{E}(\widehat{D}, c) = f(c(\hat{x}_1), c(\hat{x}_2), \ldots, c(\hat{x}_{\hat{n}}), c),$$

and f is monotone[3] *in the first $\hat{n}$ inputs.*

This relaxed notion of synthetic data is of interest because many natural approaches to sanitizing yield outputs of this type. In particular, several previous sanitization algorithms [6,27,14] produce a *set* of synthetic databases and answer a query by taking a median over the answers given by the individual databases. We view such databases as a single synthetic database but require that f have a special form. Unfortunately, the sanitizers of [14] and [27] run in time exponential in the dimension of the data, d, and the results of the next subsection show this limitation is inherent even for simple concept classes.

We now give an informal description of our hardness results for different forms of relaxed synthetic data. Our proofs use the same construction of hard-to-sanitize databases as Theorem 14 with a modified analysis and parameters to show that the output must still contain a PCP-decodable row. Formal statements and proofs of all of the following statements can be found in the full version of this paper.

- We say that a sanitizer outputs *medians of synthetic data* if it satisfies Definition 18 with

$$\mathcal{E}(\hat{x}_1,\ldots,\hat{x}_{\hat{n}},c) = \text{median}\left\{\frac{1}{|S_1|}\sum_{i\in S_1} c(\hat{x}_i), \ldots, \frac{1}{|S_\ell|}\sum_{i\in S_\ell} c(\hat{x}_i)\right\}$$

  for some partition $[\hat{n}] = S_1 \cup S_2 \cdots \cup S_\ell$. We rule out efficient sanitizers with medians of synthetic data for CSPs that are hard-to-approximate under Levin reductions within a multiplicative factor larger than 2. By Theorem 13, these CSPs include k-clause 3-CNF formulas for some constant k.

[3] Given two vectors $a = (a_1,\ldots,a_n)$ and $b = (b_1,\ldots,b_n)$ we say $b \succeq a$ iff $b_i \geq a_i$ for every $i \in [n]$. We say a function $f : \{0,1\}^n \to [0,1]$ is *monotone* if $b \succeq a \implies f(b) \geq f(a)$.

– We say that a sanitizer outputs *symmetric relaxed synthetic data* if it satisfies Definition 18 with $\mathcal{E}(\hat{x}_1, \ldots, \hat{x}_{\hat{n}}, c) = g((1/\hat{n}) \sum_{i=1}^{\hat{n}} c(\hat{x}_i))$ for a monotone function $g : [0,1] \to [0,1]$. These evaluators are symmetric both in that g does not depend on the predicate and that g only depends on the fraction of rows that satisfy the predicate. We rule out efficient sanitizers with symmetric relaxed synthetic data for CSPs that are hard-to-approximate under Levin reductions within an additive factor larger than $1/2$. By Theorem 13, these CSPs include k-clause CNF formulas, for some constant k.
– We show that no efficient sanitizer can produce accurate relaxed synthetic data for a sequence of CSPs that is $(1 - \mathrm{negl}(d))$-hard-to-approximate under inefficient Levin reductions. By Theorem 13, these CSPs include 3-CNF formulas of $\omega(\log d)$ clauses.

5.1 Positive Results for Relaxed Synthetic Data

We also show that there exists an efficient sanitizer for the family of all constant-arity predicates. As an intermediate step, we also show there exists an efficient sanitizer as symmetric relaxed synthetic data for any family of parity predicates. Our results show that relaxed synthetic data allows for more efficient sanitization than standard synthetic data, since Theorem 14 rules out an accurate, efficient sanitizer with standard synthetic data, even for 3-literal parity predicates. Our result for parities also shows that our hardness result for symmetric relaxed synthetic data is tight with respect to the required hardness of approximation, since the class of 3-literal parity predicates is $(1/2 - \epsilon)$-hard-to-approximate [19]

A function $f : \{0,1\}^d \to \{0,1\}$ is a *k-junta* if it depends on at most k variables. Let $\mathcal{J}_{d,k}$ be the set of all k-juntas on d variables.

Theorem 19. *There exists an ϵ-differentially private sanitizer that runs in time $\mathrm{poly}(n, d)$ and produces relaxed synthetic data and is (α, β)-accurate for $\mathcal{J}_{d,k}$ when*

$$n \geq \frac{C\binom{d}{\leq k} \log\left(\binom{d}{\leq k}/\beta\right)}{\alpha\epsilon}$$

for a sufficiently large constant C, where $\binom{d}{\leq k} = \sum_{i=0}^{k} \binom{d}{i}$.

To prove Theorem 19, we start with a sanitizer for *parity predicates*. A function $\chi : \{0,1\}^d \to \{-1,1\}$ is a *parity predicate*[4] if there exists a vector $s \in \{0,1\}^d$ s.t. $\chi(x) = \chi_s(x) = (-1)^{\langle x,s \rangle}$.

Theorem 20. *Let $\mathcal{P}$ be a family of parity predicates such that $\chi_{0^d} \notin \mathcal{P}$. There exists an ϵ-differentially private sanitizer $\mathcal{A}(D, \mathcal{P})$ that runs in time $\mathrm{poly}(n, d)$ and produces symmetric relaxed synthetic data that is (α, β)-accurate for $\mathcal{P}$ when*

$$n \geq \frac{2|\mathcal{P}| \log\left(2|\mathcal{P}|/\beta\right)}{\alpha\epsilon}.$$

[4] In the preliminaries we define a predicate to be a $\{0,1\}$-valued function but our definition naturally generalizes to $\{-1,1\}$-valued functions. For $c : \{0,1\}^d \to \{-1,1\}$ and database $D = (x_1, \ldots, x_n) \in (\{0,1\}^d)^n$, we define $c(D) = \frac{1}{n} \sum_{i=1}^{n} c(x_i)$

Without relaxed synthetic data, Theorems 19 and 20 can be achieved by simply releasing a vector of noisy answers to the queries [11]. Our sanitizer begins with this vector of noisy answers and constructs relaxed synthetic data from those answers. Our technique is similar to that of Barak et. al. [3], which also begins with noisy answers and constructs a (standard) synthetic database that gives approximately the same answers. However, they construct their synthetic database by solving a linear program over a set of 2^d variables, each of which represents the frequency of one of the possible rows $x \in \{0,1\}^d$. Thus their sanitizer runs in time exponential in d.

Our sanitizer also starts with a vector of noisy answers to parity queries and *efficiently* constructs symmetric relaxed synthetic data that gives answers to each query that are close to the initial noisy answers after applying a *fixed linear scaling*. To construct each row of the synthetic database, $\widehat{D}$, we select a random parity query in $\chi \in \mathcal{P}$ and then sample a row $x \in \{0,1\}^d$ such that the expectation of $\chi(x)$ is equal to the initial noisy estimate of $\chi(D)$; it can be shown that for every $\chi' \neq \chi$ (except χ_{0^d}), the expectation of $\chi'(x)$ is zero. Thus we can estimate the value of $\chi(D)$ by taking $\chi(\widehat{D})$ and multiplying by $|\mathcal{P}|$. We then show that if we apply our sanitizer to the family $\mathcal{P}_{d,k}$ containing all parity predicates on d variables that depend on at most k variables, the result is also accurate for the family $\mathcal{J}_{d,k}$ of k-juntas after applying an affine shift that depends on the average value of the junta of interest.

A complete discussion of these results is deferred to the full version of this paper.

Acknowledgments

We thank Boaz Barak, Irit Dinur, Cynthia Dwork, Vitaly Feldman, Oded Goldreich, Johan Håstad, Valentine Kabanets, Dana Moshkovitz, Anup Rao, Guy Rothblum, and Les Valiant for helpful conversations. We also thank the anonymous reviewers for helpful comments.

References

1. Adam, N.R., Wortmann, J.: Security-control methods for statistical databases: A comparative study. ACM Computing Surveys 21, 515–556 (1989)
2. Alekhnovich, M., Braverman, M., Feldman, V., Klivans, A.R., Pitassi, T.: The complexity of properly learning simple concept classes. In: J. Comput. Syst. Sci. vol. 74, pp. 16–34 (2008)
3. Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: Privacy, accuracy, and consistency too: A holistic solution to contingency table release. In: Proceedings of the 26th Symposium on Principles of Database Systems. pp. 273–282 (2007)
4. Barak, B., Goldreich, O.: Universal arguments and their applications. In: SIAM J. Comput. vol. 38, pp. 1661–1694 (2008)
5. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: The SuLQ framework. In: Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (June 2005)
6. Blum, A., Ligett, K., Roth, A.: A learning theory approach to non-interactive database privacy. In: Proceedings of the 40th ACM SIGACT Symposium on Thoery of Computing (2008)

7. Dinur, I., Nissim, K.: Revealing information while preserving privacy. In: Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. pp. 202–210 (2003)
8. Duncan, G.: International Encyclopedia of the Social and Behavioral Sciences, chap. Confidentiality and statistical disclosure limitation. Elsevier (2001)
9. Dwork, C.: A firm foundation for private data analysis. Communications of the ACM (to appear)
10. Dwork, C.: Differential privacy. In: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)(2). pp. 1–12 (2006)
11. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Proceedings of the 3rd Theory of Cryptography Conference. pp. 265–284 (2006)
12. Dwork, C., Naor, M., Reingold, O., Rothblum, G., Vadhan, S.: When and how can privacy-preserving data release be done efficiently? In: Proceedings of the 2009 International ACM Symposium on Theory of Computing (STOC) (2009)
13. Dwork, C., Nissim, K.: Privacy-preserving datamining on vertically partitioned databases. In: Proceedings of CRYPTO 2004. vol. 3152, pp. 528–544 (2004)
14. Dwork, C., Rothblum, G., Vadhan, S.P.: Boosting and differential privacy. In: Proceedings of FOCS 2010 (2010)
15. Evfimievski, A., Grandison, T.: Encyclopedia of Database Technologies and Applications, chap. Privacy Preserving Data Mining (a short survey). Information Science Reference (2006)
16. Feldman, V.: Hardness of proper learning. In: The Encyclopedia of Algorithms. Springer-Verlag (2008)
17. Feldman, V.: Hardness of approximate two-level logic minimization and PAC learning with membership queries. Journal of Computer and System Sciences 75(1), 13–26 (2009), http://dx.doi.org/10.1016/j.jcss.2008.07.007
18. Goldreich, O.: Foundations of Cryptography, vol. 2. Cambridge University Press (2004)
19. Håstad, J.: Some optimal inapproximability results. In: J. ACM. vol. 48, pp. 798–859 (2001)
20. Kearns, M.J., Valiant, L.G.: Cryptographic limitations on learning boolean formulae and finite automata. In: J. ACM. vol. 41, pp. 67–95 (1994)
21. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: STOC (1992)
22. Micali, S.: Computationally sound proofs. In: SIAM J. Comput. vol. 30, pp. 1253–1298 (2000)
23. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: STOC. pp. 33–43 (1989)
24. Pitt, L., Valiant, L.G.: Computational limitations on learning from examples. In: J. ACM. vol. 35, pp. 965–984 (1988)
25. Reiter, J.P., Drechsler, J.: Releasing multiply-imputed synthetic data generated in two stages to protect confidentiality. Iab discussion paper, Intitut für Arbeitsmarkt und Berufsforschung (IAB), Nürnberg (Institute for Employment Research, Nuremberg, Germany) (2007), http://ideas.repec.org/p/iab/iabdpa/200720.html
26. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: STOC. pp. 387–394 (1990)
27. Roth, A., Roughgarden, T.: Interactive privacy via the median mechanism. In: STOC 2010 (2010)
28. Ullman, J., Vadhan, S.P.: PCPs and the hardness of generating synthetic data. Electronic Colloquium on Computational Complexity (ECCC) 17, 17 (2010)
29. Valiant, L.G.: A theory of the learnable. Communications of the ACM 27(11), 1134–1142 (1984)