Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 17 (2010)

PCPs and the Hardness of Generating 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

January 6, 2011

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" $\hat{D}$ 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 the PCP Theorem.

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 $\hat{D}$ and aggregating the results in some way.

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

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

ISSN 1433-8092

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, 14, 21]), and recently the theoretical computer science community has developed an appealing new approach to the problem, known as *differential privacy*. (See the surveys [16, 15].).

Differential Privacy. A randomized algorithm $\mathcal{A}$ is defined to be *differentially private* [17] 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 [8]. A sequence of works [11, 19, 8, 17] 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 [33].

Previous Results on Synthetic Data. The first result on producing differentially private synthetic data came in the work of Barak et al. [5]. 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 [5] 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 [9]. 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. [18] 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 [36].

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

Theorem 1.1. *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.

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

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

severely constrain what sorts of differentially private data releases are possible. (Dwork et al. [18] 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 [32, 22], 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 [18] 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 1.1, it may not be necessary to introduce exotic data representations; some mild generalizations of synthetic data may suffice. For example, several recent algorithms [9, 35, 20] 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 1.1 and our other negative results are obtained by combining the hard-to-sanitize databases of Dwork et al. [18] 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 [18] 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 [28, 29, 6].) 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, 23] and cryptographic assumptions for hardness of representation-independent learning [36, 26]).
- 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 2.1 (Differential Privacy)**.** [17] 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 2.1, 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 $\mathcal{A}$ that outputs a synthetic database is "accurate" for the concept class $\mathcal{C}$ if the fractional counts $c(\mathcal{A}(D))$ are close to the fractional counts $c(D)$. Formally

Definition 2.2 (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 $f(n) = \text{negl}(n)$ 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 rule out 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 [18], we define the following notions

Definition 2.3 (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 $(D, D', i) \leftarrow_{\text{R}} \tilde{\mathcal{D}}$ denote the experiment in which we choose $D_0 \leftarrow_{\text{R}} \mathcal{D}$ 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 2.4 (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 two conditions hold:

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

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

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

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

   where x_i is the i-th row of D.

In [18], it was shown that every distribution that is $(\alpha, \mathcal{C})$-hard-to-sanitize in the sense of Definition 2.4, is also hard to sanitize while achieving even weak differential privacy

Claim 2.5. *[18] 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 2.4 by restricting the quantification over sanitizers to only those sanitizers that output synthetic data.

Definition 2.6 (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 2.4 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 [8] 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 with synthetic-database output 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 4.4, 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 3.1 (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

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

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

Definition 3.2 (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 3.3 (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 3.4 (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 3.4. We now argue that for every circuit C, the formula $\varphi_C = R(C)$ satisfies conditions 1 and 2 of Definition 3.3. Specifically, if there exists an assignment $u \in \{0,1\}^{\bar{d}}$ 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 3.5 (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.*

Proof sketch. Hardness under Karp reductions follows directly from the classification theorems of Creignou [10] and Khanna et al. [27]. These theorems show that all CSPs are either α-hard under Karp reductions for some constant $\alpha > 0$ or can be solved optimally in polynomial time. By inspection, the only CSPs that fall into the polynomial-time cases (0-valid, 1-valid, and 2-monotone) and are closed under negation are those containing only dictatorships and constant functions.

The fact that standard PCPs actually yield Levin reductions has been explicitly discussed and formalized by Barak and Goldreich [6] in the terminology of PCPs rather than reductions (the function Enc is called "relatively efficient oracle-construction" and the function Dec is called "a proof-of-knowledge property"). They verify that these properties hold for the PCP construction of Babai et al. [4], whereas we need it for PCPs of constant query complexity. While the properties probably hold for most (if not all) existing PCP constructions, the existence of the efficient "decoding" function g requires some verification. We observe that it follows as a black box from the PCPs of Proximity of [7, 12]. There, a prefix of the PCP (the "implicit input oracle") can be taken to be the encoding of a satisfying assignment of the circuit C in an efficiently decodable error-correcting code. If the PCP verifier accepts with higher probability than the soundness error s, then it is guaranteed that the prefix is close to a valid codeword, which in turn can be decoded to a satisfying assignment. By the correspondence between PCPs and CSPs [3], this yields a CSP (with constraints of constant arity) that is α-hard to approximate under Levin reductions for some constant $\alpha > 0$ (and $\gamma = 1$). The sequence of approximation-preserving reductions from arbitrary CSPs to MAX-CUT [31] can be verified to preserve efficiency of decoding (indeed, the correctness of the reductions is proven by specifying how to encode and decode). Finally, the reductions of [27] from MAX-CUT to any other CSP all involve constant-sized "gadgets" that allow encoding and decoding to be done locally and very efficiently. □

It seems likely that optimized PCP/inapproximability results (like [25]) 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 3.6. *There is a constant C such that for every $\epsilon = \epsilon(d) > 0$, 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 $k(d) = C\log(1/\epsilon(d))$.*

Proof sketch. As in the proof of Theorem 3.5, disjunctions of 3 literals are $(1-\delta, 1)$-hard-to-approximate under Levin reductions for some constant $\delta > 0$. By taking $k(d) = \log_\delta(1/\epsilon(d))$ sequential repetitions of this PCP, we get a PCP with completeness 1 and soundness $\epsilon(d)$ whose constraints are 3-CNF formulas with $k(d) = \log_\delta(1/\epsilon(d))$ clauses.

We have to check that this resulting PCP preserves the properties of inefficient Levin reductions. The encoder for the k-fold sequential repetition is unchanged. If the initial reduction is $R(C) = \varphi_C = \{\varphi_1, \ldots, \varphi_m\}$ (a set of 3-literal disjunctions), then the reduction $R^k(C)$ for the k-fold sequential repetition will produce m^k, k-clause 3-CNF formulae by taking every subcollection of k clauses in φ_C. Specifically, for every $i_1, i_2, \ldots, i_k \in [m]$, $R^k(C)$ will contain a k-clause 3-CNF formula $\varphi_{i_1} \wedge \varphi_{i_2} \wedge \cdots \wedge \varphi_{i_k}$.

The decoder also remains unchanged. If the value of an assignment π is at least δ^k with respect to $R^k(C)$ then it must have value at least δ with respect to $R(C)$ and thus $Dec(\pi, C)$ will return a satisfying assignment to C, that is $C(Dec(\pi, C)) = 1$.

Notice that when $k = k(d) = \omega(1)$, the reduction will produce $m^{\omega(1)}$ clauses and be inefficient. Thus we will have an inefficient Levin reduction if we want to obtain $\epsilon(d) = o(1)$ from this construction. □

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 [18], 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 3.4, 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 Super-Secure Digital Signature Schemes

Before proving our main result, we will formally define a *super-secure digital signature scheme*. These digital signature schemes have the property that it is infeasible under chosen-message attack to find a new message-signature pair that is different from all obtained during the attack, even a new signature for an old message. First we formally define digital signature schemes

Definition 4.1 (Digital signature scheme)**.** A *digital signature scheme* is a tuple of three probabilistic polynomial time algorithms $\Pi = (Gen, Sign, Ver)$ such that

1. Gen takes as input the security parameter 1^κ and outputs a key pair $(sk, vk) \leftarrow_R Gen(1^\kappa)$.
2. $Sign$ takes sk and a message $m \in \{0,1\}^*$ as input and outputs $\sigma \leftarrow_R Sign_{sk}(m)$.
3. Ver takes vk and pair (m, σ) and deterministically outputs a bit $b \in \{0,1\}$, such that for every (sk, vk) in the range of Gen, and every message m, we have $Ver_{vk}(m, Sign_{sk}(m)) = 1$.

We define the security of a digital signature scheme with respect to the following game.

Definition 4.2 (Weak forgery game)**.** For any signature scheme $\Pi = (Gen, Sign, Ver)$ and probabilistic polynomial time adversary $\mathcal{F}$, $\texttt{WeakForge}(\mathcal{F}, \Pi, \kappa)$ is the following probabilistic experiment.

1. $(sk, vk) \leftarrow_{\text{R}} Gen(1^\kappa)$.
2. $\mathcal{F}$ is given vk and oracle access to $Sign_{sk}$. The adversary adaptively queries $Sign_{sk}$ on a set of messages $Q \subset \{0,1\}^*$, receives a set of message-signature pairs $A \subset \{0,1\}^*$ and outputs (m^*, σ^*).
3. The output of the game is 1 if and only if (1) $Ver_{vk}(m^*, \sigma^*) = 1$, and (2) $(m^*, \sigma^*) \notin A$.

The weak forgery game is easier for the adversary to win than the standard forgery game because the final condition requires that the signature output by $\mathcal{F}$ be different from all pairs $(m, \sigma) \in A$, but allows for the possibility that $m^* \in Q$. In the standard definition, the final condition would be replaced by $m^* \notin Q$. Thus the adversary has more possible outputs that would result in a "win" under this definition than under the standard definition.

Definition 4.3 (Super-secure digital signature scheme)**.** A digital signature scheme $\Pi = (Gen, Sign, Ver)$ is *super-secure under adaptive chosen-message attack* if for every probabilistic polynomial time adversary, $\mathcal{F}$, $\Pr[\texttt{WeakForge}(\mathcal{F}, \Pi, \kappa) = 1] \leq \text{negl}(\kappa)$.

Although the above definition is stronger than the usual definition of existentially unforgeable digital signatures, in [24] it is shown how to modify known constructions [30, 34] to obtain a super-secure digital signature scheme from any one-way function.

4.2 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 3.1. We now state our hardness result.

Theorem 4.4. *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 super-secure digital signature scheme 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 3.4. 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 3.2).

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 $\text{poly}(\kappa) + q(d)\text{poly}(\kappa) \leq d^{1-\Omega(1)}\text{poly}(d^\tau)$, 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.

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

$(sk, vk) \leftarrow_{\mathrm{R}} Gen(1^\kappa)$, let $vk = vk_1 vk_2 \dots vk_\ell$, where $\ell = |vk| = \mathrm{poly}(\kappa)$
$(m_1, \dots, m_{n+1}) \leftarrow_{\mathrm{R}} (\{0,1\}^\kappa)^{n+1}$
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 $D_0 := (x_1, \dots, x_{n+1})$

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 $\varphi^*_j \in \mathcal{C}^{(d)}_\Gamma$ for $j = 1, 2, \dots, \ell$, by our construction of $\mathcal{C}^{(d)}_\Gamma$ (Definition 3.1).

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

Lemma 4.5. *There exists a polynomial-time adversary $\mathcal{T}$ such that for every polynomial-time sanitizer $\mathcal{A}$,*

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

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:

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

Claim 4.6. *If $\widehat{D} = \mathcal{A}(D)$ is α-accurate for $\mathcal{C}^{(d)}_\Gamma$, 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 $\mathcal{C}^{(d)}_\Gamma$ 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 3.4) implies that each row x_i of D has $\mathrm{val}(\varphi_{C_{vk}}, x_i) \geq \gamma$. Thus if $\varphi_{C_{vk}} = \{\varphi_1, \dots, \varphi_m\}$, then

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

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

Let d be the dimension of rows in $\widehat{D}$, $\kappa = d^\tau$, $\ell = |vk| = \text{poly}(\kappa)$.
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 $Enc(\mathcal{T}_0(\widehat{D}), C_{\widehat{vk}})$

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

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

So for at least one row $\hat{x} \in \widehat{D}$ it must be the case that $\text{val}(\varphi_{C_{vk}}, \hat{x}) \geq \gamma - \alpha$. The definition of Dec (Definition 3.4) 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 4.6), then it must be one of the message-signature pairs used to construct D (except with probability $\text{negl}(\kappa) = \text{negl}(d)$). This implies that $\mathcal{T}(\mathcal{A}(D)) = Enc(\hat{x}^*, C_{vk}) \in D$ (except with negligible

probability). Thus, we have

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

which is equivalent to the statement of the lemma. □

Lemma 4.7.

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

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 $2^{-\kappa} = \text{negl}(d)$. □

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

Theorem 1.1 in the introduction follows by combining Theorems 3.5 and 4.4.

5 Relaxed Synthetic Data

The proof of Theorem 4.4 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 interpreted using an evaluation function of the following form: To evaluate predicate $c \in \mathcal{C}$ on $\widehat{D}$, apply c to each row $\hat{x}_i$ of $\widehat{D}$ to get a string of $\hat{n}$ bits, and then apply a function $f : \{0,1\}^{\hat{n}} \times \mathcal{C} \to [0,1]$ to determine the answer. 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 5.1 (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[2] 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 [20, 35, 9] produce a *set* of synthetic databases and answer a query by taking a median over the answers given by the individual databases. Sanitizers that use medians of synthetic databases no longer have the advantage that they are "interchangeable" with the original data, but are still be desirable for data releases because they retain the property that a small data structure can give accurate answers to a large number of queries. We view such databases

[2]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 $b \succeq a \Longrightarrow f(b) \geq f(a)$.

as a single synthetic database but require that f have a special form. Unfortunately, the sanitizers of [20] and [35] 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.

Throughout this section we will continue to use $c(\widehat{D})$ to refer to the answer given by $\widehat{D}$ *when interpreted as a synthetic database*.

We now present our hardness results for relaxed synthetic data where the function f takes the median over synthetic database (Section 5.1), where f is an arbitrary monotone, symmetric function (Section 5.2), or when the family of concepts contains CSPs that are very hard to approximate (Section 5.3). Our proofs use the same construction of hard-to-sanitize databases as Theorem 4.4 with a modified analysis and parameters to show that the output must still contain a PCP-decodable row.

5.1 Hardness of Sanitizing as Medians

In this section we establish that the distribution $\mathcal{D}$ used in the proof of Theorem 4.4 is hard-to-sanitize as medians of synthetic data, formally defined as:

Definition 5.2 (medians of synthetic data)**.** A sanitizer $\mathcal{A} : (\{0,1\}^d)^n \to (\{0,1\}^d)^{\hat{n}}$ with evaluator $\mathcal{E}$ outputs *medians of synthetic data* if there is a partition $[\hat{n}] = S_1 \cup S_2 \cdots \cup S_\ell$ such that

$$\mathcal{E}(\hat{x}_1, \ldots, \hat{x}_{\hat{n}}, c) = \text{median}\left\{ \frac{1}{|S_1|}\sum_{i \in S_1} c(\hat{x}_i), \frac{1}{|S_2|}\sum_{i \in S_2} c(\hat{x}_i), \ldots, \frac{1}{|S_\ell|}\sum_{i \in S_\ell} c(\hat{x}_i) \right\}.$$

Note that medians of synthetic data are a special case of relaxed synthetic data. In the following, we rule out efficient sanitizers with medians of synthetic data for CSPs that are hard to approximate within a multiplicative factor larger than 2. By Theorem 3.6, these CSPs include k-clause 3-CNF formulas for some constant k.

Theorem 5.3. *Let $\Gamma = (\Gamma_d)_{d \in \mathbb{N}}$ be a family of nice (Definition 3.2) $q(d)$-CSPs such that $\Gamma_d \cup \neg\Gamma_d$ is $((\alpha(d)+\gamma(d))/2, \gamma(d))$-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 medians of synthetic data.*

Proof. Let Γ be a family of CSPs that is $((\alpha+\gamma)/2, \gamma)$-hard-to-approximate under Levin reductions. Let $\mathcal{D} = \mathcal{D}_d$ be the database distribution ensemble described in the proof of Theorem 4.4. Let $\mathcal{A}(D) = \widehat{D}$ and let $\left\{\widehat{D}_1, \widehat{D}_2, \ldots, \widehat{D}_\ell\right\}$ be the partition of the rows of $\widehat{D}$ corresponding to $S_1, \ldots, S_\ell$, i.e. $\widehat{D}_i = (\hat{x}_j)_{j \in S_i}$.

Assuming that $\widehat{D}$ is α-accurate as medians of synthetic data, we will show that there must exist a row $\hat{x} \in \widehat{D}$ such that $\text{val}(\varphi_{C_{vk}}, \hat{x}) \geq \gamma - (\alpha+\gamma)/2 = (\gamma-\alpha)/2$. To do so, we observe that if $\widehat{D}$ is accurate as medians of synthetic databases, then for each predicate, half of $\widehat{D}$'s synthetic databases must give an answer that is "close to D's answer". Thus one of these synthetic datsbases must be "close" to D for half of the predicates in $\varphi_{C_{vk}}$. By our construction of D, we conclude that each of these predicates is satisfied by many rows of this synthetic database and thus some row satisfies enough of the predicates to decode a message-signature pair.

We need to show that there exists an adversary $\mathcal{T}$ such that for every polynomial time sanitizer $\mathcal{A}$,

$$\Pr_{\substack{(D,D',i) \leftarrow_R \tilde{\mathcal{D}} \\ \mathcal{A}'s \text{ and } \mathcal{T}'s \text{ coins}}} \left[(\mathcal{A}(D) \text{ is } \alpha\text{-accurate for } \mathcal{C}_\Gamma^{(d)}) \wedge (\mathcal{T}(\mathcal{A}(D)) \cap D = \emptyset) \right] \leq \text{negl}(d) \tag{2}$$

To do so, we will use the same subroutine $\mathcal{T}_0(\widehat{D})$ we used for the proof of Lemma 4.5. That is, we consider a subroutine that looks for rows satisfying sufficiently many clauses of $\varphi_{C_{vk}}$ and returns the PCP-decoding of that row. It will suffice to establish the following claim, analogous to Claim 4.6:

Claim 5.4. *If $\widehat{D}$ is α-accurate for $\mathcal{C}_\Gamma^{(d)}$ as medians of synthetic data, then $\mathcal{T}_0(\widehat{D})$ outputs a pair (m, σ) s.t. $C_{vk}(m, \sigma) = 1$.*

Proof. As in the proof of Claim 4.6, if $\widehat{D}$ is α-accurate for $\mathcal{C}_\Gamma^{(d)}$ for $\alpha < 1/2$, then $\mathcal{K}(\widehat{D}) = vk$, the verification key used in the construction of D_0. For the rest of the proof we will be justified in substituting vk for $\widehat{vk}$.

If $\varphi_{C_{vk}} = \{\varphi_1, \ldots, \varphi_m\}$, then $\frac{1}{m}\sum_{j=1}^m \varphi_j(D) \geq \gamma$. We say that $\widehat{D}_k$ is *good for* φ_j if $\varphi_j(\widehat{D}_k) \geq \varphi_j(D) - \alpha$. Since the median over $\left\{\widehat{D}_1, \widehat{D}_2, \ldots, \widehat{D}_\ell\right\}$ is α-accurate for every constraint $\varphi_j \in \varphi_{C_{vk}}$ we have

$$\Pr_{k \leftarrow_R [\ell]}\left[\widehat{D}_k \text{ is good for } \varphi_j\right] \geq \frac{1}{2}$$

Then

$$\begin{aligned}
\mathop{\mathbb{E}}_{k \leftarrow_R [\ell]}\left[\frac{1}{|S_k|}\sum_{i \in |S_k|} \mathrm{val}(\varphi_{C_{vk}}, \hat{x}_i)\right] &= \mathop{\mathbb{E}}_{k \leftarrow_R [\ell]}\left[\frac{1}{m}\sum_{j=1}^m \varphi_j(\widehat{D}_k)\right] \\
&= \frac{1}{m}\sum_{j=1}^m \mathop{\mathbb{E}}_{k \leftarrow_R [\ell]}\left[\varphi_j(\widehat{D}_k)\right] \\
&\geq \frac{1}{m}\sum_{j=1}^m \left(\Pr_{k \leftarrow_R [\ell]}\left[\widehat{D}_k \text{ is good for } \varphi_j\right] \cdot ((\varphi_j(D) - \alpha)\right) \\
&\geq \frac{1}{m}\sum_{j=1}^m \left(\frac{1}{2} \cdot (\varphi_j(D) - \alpha)\right) \\
&\geq \frac{\gamma - \alpha}{2}
\end{aligned}$$

□

So for at least one row $\hat{x} \in \widehat{D}$ it must be the case that $\mathrm{val}(\varphi_{C_{vk}}, \hat{x}) \geq (\gamma - \alpha)/2$. Since the distribution $\mathcal{D}$ is unchanged, Lemma 4.7 still holds in this setting. Thus we have established that $\mathcal{D}$ is $(\alpha, \mathcal{C}_\Gamma)$-hard-to-sanitize as medians of synthetic data. □

5.2 Hardness of Sanitizing with Symmetric Evaluation Functions

In this section we establish the hardness of sanitization for relaxed synthetic data where the evaluator function is symmetric.

Definition 5.5 (symmetric relaxed synthetic data)**.** A sanitizer $\mathcal{A} : (\{0,1\}^d)^n \to (\{0,1\}^d)^{\hat{n}}$ with evaluator $\mathcal{E}$ outputs *symmetric relaxed synthetic data* if there exists a monotone function $g : [0,1] \to [0,1]$ such that

$$\mathcal{E}(\hat{x}_1, \ldots, \hat{x}_{\hat{n}}, c) = g\left(\frac{1}{\hat{n}}\sum_{i=1}^{\hat{n}} c(\hat{x}_i)\right).$$

Note that symmetric relaxed synthetic data is also a special case of relaxed synthetic data. Our definition of symmetric relaxed synthetic data is actually symmetric in two respects, because we require that g does not depend on the predicate c and also that g only depends on the fraction of rows that satisfy c. Similar to medians of synthetic data,we show that it is intractable to produce a sanitization as symmetric relaxed synthetic data that is accurate when the queries come from a CSP that is hard to approximate.

Theorem 5.6. *Let $\Gamma = (\Gamma_d)_{d\in\mathbb{N}}$ be a family of nice (Definition 3.2) $q(d)$-CSPs that is closed under complement ($\Gamma_d = \neg\Gamma_d$) and is $(\alpha(d) + 1/2, \gamma(d))$-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 symmetric relaxed synthetic data.*

By Theorem 3.6, the family of k-clause CNF formulas, for some constant k, is $(1/2 + \alpha)$-hard-to-approximate under Levin reductions for $\alpha > 0$.

Proof. Let Γ be a family of CSPs that is $(\alpha+1/2)$-hard-to-approximate under Levin reductions. Let $\mathcal{D} = \mathcal{D}_d$ be the database distribution ensemble described in the proof of Theorem 4.4, $D \leftarrow_{\mathrm{R}} \mathcal{D}$, and $\widehat{D} = \mathcal{A}(D)$. We will use the idea from Theorem 4.4, which shows that the underlying synthetic database cannot contain a row that satisfies too many clauses of $\varphi_{C_{vk}}$, to show that g must map a small input to a large output and a large input to a small answer, contradicting the monotonicity of g.

Let $\varphi_{C_{vk}} = \{\varphi_1, \ldots, \varphi_m\}$, then $\frac{1}{m}\sum_{j=1}^m \varphi_j(D) \geq \gamma$. It must also be that $\frac{1}{m}\sum_{j=1}^m \varphi_j(\widehat{D}) \leq \gamma - \alpha - 1/2$. Otherwise there would exist a row $\hat{x} \in \widehat{D} = \mathcal{A}(D)$ such that $\mathrm{val}(\varphi_{C_{vk}}) \geq \gamma - \alpha - 1/2$. But if this were the case we could PCP-decode $\hat{x}$ as in the proof of Theorem 4.4. Thus

$$\frac{1}{m}\sum_{j=1}^m \left(\varphi_j(D) - \varphi_j(\widehat{D})\right) \geq \alpha + 1/2$$

so there must exist $J \in [m]$ s.t. $\varphi_J(D) - \varphi_J(\widehat{D}) \geq \alpha + 1/2$. Since $\varphi_J(\widehat{D}) \geq 0$ we also have $\varphi_J(D) \geq \alpha + 1/2 > 1/2$, and since $\varphi_J(D) \leq 1$ we also have $\varphi_J(\widehat{D}) \leq 1/2 - \alpha < 1/2$.

By monotonicity of g and α-accuracy of $\widehat{D}$ as symmetric relaxed synthetic data we have

$$g(1/2 - \alpha) \geq g(\varphi_J(\widehat{D})) \geq \varphi_J(D) - \alpha \geq \frac{1}{2}.$$

Consider the negation of φ_J. Since $\neg\varphi_J(D) = 1 - \varphi_J(D)$ we can conclude that $\neg\varphi_J(D) \leq 1/2 - \alpha$ and $\neg\varphi_J(\widehat{D}) \geq 1/2 + \alpha$. Thus we have

$$g(1/2 + \alpha) \leq g(\neg\varphi_J(\widehat{D})) \leq \neg\varphi_J(D) + \alpha \leq \frac{1}{2}.$$

But $g(1/2 - \alpha) \geq 1/2 \geq g(1/2 + \alpha)$ and $\alpha > 0$ contradicts the monotonicity of g. □

5.3 Hardness of Sanitizing Very Hard CSPs with Relaxed Synthetic Data

In this section 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 3.6, these CSPs include 3-CNF formulas of $\omega(\log d)$ clauses.

Intuitively, an efficient sanitizer must produce a synthetic database of $\hat{n}(d) = \text{poly}(d)$ rows, and thus as d grows, an efficient sanitizer cannot produce a synthetic database that contains a row satisfying a non-negligible fraction of clauses from a particular CSP instance (the signature-verification CSP from our earlier results). Thus using evaluators of the type in Definition 5.1 there can only be one answer to most queries, and thus we cannot get an accurate sanitizer.

Theorem 5.7. *Let $\Gamma = (\Gamma_d)_{d\in\mathbb{N}}$ be a family of nice (Definition 3.2) $q(d)$-CSPs such that $\Gamma_d \cup \neg\Gamma_d$ is $(1-\epsilon(d), 1)$-hard-to-approximate under (possibly inefficient) Levin reductions for a negligible function $\epsilon(d)$. 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 $(1/3, \mathcal{C}_\Gamma^{(d)})$-hard-to-sanitize as relaxed synthetic data.*

Proof. Let $\Gamma = \Gamma(d)$ be a family of CSPs that is $(1-\epsilon(d), 1)$-hard-to-approximate under Levin reductions. Let $\mathcal{D} = \mathcal{D}_d$ be the database distribution ensemble described in the proof of Theorem 4.4. Note that since $\Gamma(d)$ depends on d there will be a sequence of triples (Enc_d, Dec_d, R_d) and the construction of $\mathcal{D}_d$ should use the appropriate encoder for each choice of d. Let $D \leftarrow_{\text{R}} \mathcal{D}_d$, and $\mathcal{A}(D) = \widehat{D}$.

Let $\varphi_{C_{vk}} = \{\varphi_1, \ldots, \varphi_m\}$. By the construction of $\mathcal{D}_d$ we have

$$\varphi_j(D) = 1$$

for every $j \in [m]$. As in the proof of Theorem 5.6 it must be that

$$\frac{1}{m}\sum_{j=1}^{m} \varphi_j(\widehat{D}) \leq \epsilon(d).$$

Otherwise there would exist a row $\hat{x} \in \widehat{D} = \mathcal{A}(D)$ such that $\text{val}(\varphi_{C_{vk}}) \geq \epsilon(d)$. But if this were the case we could PCP-decode $\hat{x}$ as in the proof of Theorem 4.4.

Since $\mathbb{E}_{j\leftarrow_{\text{R}}[m]}[\varphi_j(\widehat{D})] \leq \epsilon(d)$, there must exist a subset $J \subseteq [m]$ of size $|J| \geq 2m/3$ such that for all $j \in J$, $\varphi_j(\widehat{D}) \leq 3\epsilon(d) \leq \text{negl}(d)$.

Since $\widehat{D} \in (\{0,1\}^d)^{\hat{n}}$ for $\hat{n} = \hat{n}(d) = \text{poly}(d)$ (by the efficiency of $\mathcal{A}$),

$$\frac{1}{\hat{n}}\sum_{i=1}^{\hat{n}} \varphi_j(\hat{x}_i) = \varphi_j(\widehat{D}) \in \{0, 1/\hat{n}(d), 2/\hat{n}(d), \ldots, 1\}$$

and thus $\varphi_j(\widehat{D}) \leq \text{negl}(d)$ implies $\varphi_j(\widehat{D}) = 0$ for large n. Let $\mathcal{E}(\widehat{D}, c) = f(c(\hat{x}_1), \ldots, c(\hat{x}_{\hat{n}}), c)$. If we assume $\widehat{D}$ is $1/3$-accurate as relaxed synthetic data then $f(0^{\hat{n}}, \varphi_j) \geq 2/3$ for every $j \in J$.

Now consider the execution of $\mathcal{A}$ on the database $D' = (0^d)^n$. With probability $1-\text{negl}(d)$, $Dec(0^d, C_{vk})$ is not a valid message-signature pair, thus by Definition 3.4 Part 2, we have

$$\varphi_{C_{vk}}(D') = \varphi_{C_{vk}}(0^d) \leq \epsilon(d).$$

Since the rows of D' are idenitcal, $\varphi_j(D') \in \{0,1\}$ for every $j \in [m]$. So for at least a $1-\epsilon(d)$ fraction of $j \in [m]$, we have $\varphi_j(D') = 0$.

Let $\widehat{D}' = \mathcal{A}(D')$. By repeating the signature-forging argument, we see that with probability $1 - \text{negl}(d)$

$$\frac{1}{m}\sum_{j=1}^{m} \varphi_j(\widehat{D}') \leq \epsilon(d)$$

and thus there must exist a subset $J' \subseteq [m]$ of size $|J'| \geq 2m/3$ such that $j \in J' \implies \varphi_j(\widehat{D}') \leq 3\epsilon(d) \leq \mathrm{negl}(d)$. So $\varphi_j(\widehat{D}') = 0$ for every $j \in J'$ as well. There must also exist a set $J'' \subseteq J'$ of size $|J''| = (2/3 - \epsilon(d))m$, such that for every $j \in J''$, $\varphi_j(D') = \varphi_j(\widehat{D}') = 0$. So if $\widehat{D}'$ is $1/3$-accurate for $\mathcal{C}$ as relaxed synthetic data it must be that $f(0^{\hat{n}}, \varphi_j) \leq 1/3$ for every $j \in J''$.

By our choice of J and J'' there must exist $j \in J \cap J''$ such that:

1. $f(0^{\hat{n}}, \varphi_j) \geq 2/3$, and
2. $f(0^{\hat{n}}, \varphi_j) \leq 1/3$,

which is a contradiction.

□

5.4 Positive Results for Relaxed Synthetic Data

In this section we present an efficient, accurate sanitizer for small (e.g. polynomial in d) families of parity queries that outputs symmetric relaxed synthetic data and show that this sanitizer also yields accurate answers for any family of constant-arity predicates when evaluated as a relaxed synthetic data. Our result for parities shows that relaxed synthetic data (and even symmetric relaxed synthetic data) allows for more efficient sanitization than standard synthetic data, since Theorem 4.4 rules out an accurate, efficient sanitizer that produces a standard synthetic database, even for the class of 3-literal parity predicates. Our result for parities also shows that our hardness result for symmetric relaxed synthetic data (Theorem 5.6) 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 [25]

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 5.8. *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 $\binom{d}{\leq k} = \sum_{i=0}^{k} \binom{d}{i}$.

The privacy, accuracy, and efficiency guarantees of our theorem can be achieved without relaxed synthetic data simply by releasing a vector of noisy answers to the queries [17]. Our sanitizer will, in fact, begin with this vector of noisy answers and construct relaxed synthetic data from those answers. Our technique is similar to that of Barak et. al. [5], which begins with a vector of noisy answers to *parity queries* (defined in Section 5.4.1) and constructs a (standard) synthetic database that gives answers to each query that are close to the initial noisy answers. They construct their synthetic database by solving a linear program over 2^d variables that correspond to the frequency of each possible row $x \in \{0,1\}^d$, and 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*. We then show that the database our sanitizer constructs is also accurate for the family of k-juntas using an affine shift that depends on the junta.

5.4.1 Efficient Sanitizer for Parities

To prove Theorem 5.8, we start with a sanitizer for *parity predicates*.

Definition 5.9 (Parity Predicate)**.** A function $\chi : \{0,1\}^d \to \{-1,1\}$ is a *parity predicate* [3] if there exists a vector $s \in \{0,1\}^d$ s.t.

$$\chi(x) = \chi_s(x) = (-1)^{\langle x,s \rangle}.$$

We will use $wt(s) = \sum_{i=1}^{d} s_i$ to denote the number of non-zero entries in s.

Theorem 5.10. *Let $\mathcal{P}$ be a family of parity predicates on d variables 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*

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

The analysis of our sanitizer will make use of the following standard fact about parity predicates

Fact 5.11. *Two parity predicates $\chi_s, \chi_t : \{0,1\}^d \to \{-1,1\}$ are either identical or orthogonal. Specifically, for $s \neq t$, $s \neq 0^d$ and $b \in \{-1,1\}$,*

$$\mathop{\mathbb{E}}_{x \leftarrow_R \{0,1\}^d}[\chi_s(x) | \chi_t(x) = b] = \mathop{\mathbb{E}}_{x \leftarrow_R \{0,1\}^d}[\chi_s(x)] = 0.$$

Our sanitizer will start with noisy answers to the predicate queries $\chi_s(D)$. Each noisy answer will be the true answer perturbed with noise from a *Laplace distribution*. The Laplace distribution $Lap(\sigma)$ is a continuous distribution on $\mathbb{R}$ with probability density function $p_\sigma(y) \propto \exp(-|y|/\sigma)$. The following theorem of Dwork, et. al. [17] shows that these queries are differentially private for an appropriate choice of σ.

Theorem 5.12 ([17])**.** *Let $(c_1, c_2, \ldots, c_k)$ be a set of predicates and let $\sigma = k/n\epsilon$ and let $D \in (\{0,1\}^d)^n$ be a database. Then the mechanism $\mathcal{A}(D) = (c_1(D) + Z_1, c_2(D) + Z_2, \ldots, c_k(D) + Z_k)$, where $(Z_1, \ldots, Z_k)$ are independent samples from $Lap(\sigma)$ is ϵ-differentially private.*

In order to argue about the accuracy of our mechanism we need to know how much error is introduced by noise from the Laplace distribution. The following fact gives a bound on the tail of a Laplace random variable.

Fact 5.13. *The tail of the Laplace distribution decays exponentially. Specifically,*

$$\Pr[|Lap(\sigma)| \geq t] = \exp(-t/\sigma).$$

Now we present our sanitizer for queries that are parity functions. We will not consider the query χ_{0^d} as $\chi_{0^d}(x) = 1$ for every $x \in \{0,1\}^d$. Let $\mathcal{P}$ be a set of parity functions that does not contain χ_{0^d}. We now present a $\mathrm{poly}(n, d, |\mathcal{P}|)$-time sanitizer for $\mathcal{P}$.

Our sanitizer starts by getting noisy estimates of the quantities $\chi(D)$ for each predicate $\chi \in \mathcal{P}$ by adding Laplace noise. Then it builds the relaxed synthetic data $\widehat{D}$ in blocks of rows. Each block of rows is "assigned" to contain an answer to a query χ. In that block we randomly choose rows such that the expected

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

value of χ on each row equals the noisy estimate of $\chi(D)$. By Fact 5.11, the expected value of every other predicate χ' is 0 for rows in this block. The sanitizer is accurate so long as the total number of rows is sufficient for the value of $\chi(\widehat{D})$ to be concentrated around its expectation.

Sanitizer $\mathcal{A}(D, \mathcal{P})$, where $P = \{\chi^{(1)}, \ldots, \chi^{(t)}\}$:
  Let $\sigma := |\mathcal{P}|/n\epsilon \quad T := (2|\mathcal{P}|/\alpha^2)\log(4|\mathcal{P}|/\beta)$
  for all $j = 1, \ldots, t$ **do**
    Let $a_j := \chi^{(j)}(D) + Lap(\sigma)$
    for $i = jT + 1$ to $(j+1)T$ **do**
      With probability $(a_j + 1)/2$: Let $\hat{x}_i \leftarrow_R \{x \in \{0,1\}^d \mid \chi^{(j)}(x) = 1\}$
      Otherwise: Let $\hat{x}_i \leftarrow_R \{x \in \{0,1\}^d \mid \chi^{(j)}(x) = -1\}$
    end for
  end for
  return $\widehat{D} = (\hat{x}_1, \ldots, \hat{x}_{tT})$

Evaluator $\mathcal{E}_{\mathcal{P}}(\widehat{D}, \chi)$:
  return $|\mathcal{P}| \cdot \chi(\widehat{D})$

The following claims will suffice to establish Theorem 5.10

Claim 5.14. *$\mathcal{A}$ is ϵ-differentially private.*

Proof. The output of $\mathcal{A}$ only depends on the answer to $|\mathcal{P}|$ predicate queries. By Theorem 5.12 the answers to $|\mathcal{P}|$ predicate queries perturbed by independent samples from $Lap(|\mathcal{P}|/n\epsilon)$ is ϵ-differentially private. □

Claim 5.15. *$\mathcal{A}$ is (α, β)-accurate for $\mathcal{P}$ when*

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

Proof. We want to show that for every $\chi^{(j)} \in \mathcal{P}$

$$\left||\mathcal{P}| \cdot \chi^{(j)}(\widehat{D}) - \chi^{(j)}(D)\right| \leq \alpha$$

except with probability β. To do so we consider separately the error introduced in going from $\chi^{(j)}(D)$ to a_j using Laplacian noise and the error introduced in going from noisy answers a_j to $\chi^{(j)}(\widehat{D})$ by sampling rows at random.

First we bound the error introduced by the noisy queries to D. Specifically, we want to show that for every $\chi^{(j)} \in \mathcal{P}$

$$\left|\chi^{(j)}(D) - a_j\right| \leq \alpha/2$$

except with probability $\beta/2$. For each $\chi^{(j)}$ we have

$$\Pr[|\chi^{(j)}(D) - a_j| \geq \alpha/2] \leq \exp(-n\epsilon\alpha/2|\mathcal{P}|)$$

by Fact 5.13. So by a union bound we have

$$\Pr[\exists\chi^{(j)}\ |\chi^{(j)}(D) - a_j| \geq \alpha/2] \leq |\mathcal{P}|\exp(-\alpha/2\sigma) \leq |\mathcal{P}|\exp(-n\epsilon\alpha/2|\mathcal{P}|) < \beta/2,$$

so long as

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

We also want to show that for every $\chi^{(j)} \in \mathcal{P}$

$$\left||\mathcal{P}| \cdot \chi^{(j)}(\widehat{D}) - a_j\right| \leq \alpha/2$$

except with probability $\beta/2$, where a_j is the noisy answer for $\chi^{(j)}(D)$ computed in $\mathcal{A}(D)$. To do so, we will show that the expectation of $|\mathcal{P}|\chi^{(j)}(\widehat{D})$ is indeed a_j, then we will use a Chernoff-Hoeffding bound to show that the random rows generated by $\mathcal{A}(D)$ are close to their expectation. Finally we take a union bound over all $\chi \in \mathcal{P}$.

Fix $\chi^{(j)} \in \mathcal{P}$ and consider $\chi^{(j)}(\widehat{D})$. $\chi^{(j)}(\widehat{D})$ is the sum of T independent biased coin flips. In rows $jT+1, jT+2, \ldots, (j+1)T$ (the rows where we focus on $\chi^{(j)}$) the expectation of each coin flip is a_j, and in all other rows the expectation of each coin flip is 0 by Fact 5.11. Thus

$$\mathbb{E}\left[\chi^{(j)}(\widehat{D})\right] = \mathbb{E}\left[\frac{1}{\hat{n}}\sum_{i=1}^{\hat{n}} \chi^{(j)}(\hat{x}_i)\right] = a_j/|\mathcal{P}|$$

for every $\chi^{(j)} \in \mathcal{P}$.

By a Chernoff-Hoeffding Bound[4] we conclude

$$\Pr\left[\left||\mathcal{P}| \cdot \chi^{(j)}(\widehat{D}) - a_j\right| \geq \alpha/2\right] < 2\exp(-T\alpha^2/2|\mathcal{P}|).$$

By taking a union bound over $\mathcal{P}$ we conclude

$$\Pr\left[\exists \chi^{(j)} \ \left||\mathcal{P} \cdot |\chi^{(j)}(\widehat{D}) - a_j\right| \geq \alpha/2\right] < 2|\mathcal{P}|\exp(-T\alpha^2/2|\mathcal{P}|) \leq \beta/2.$$

Combining the two bounds suffices to prove the claim. □

5.4.2 Efficient Sanitizer for k-Juntas

We now show that our sanitizer for parity queries can also be used to give accurate answers for any family of k-juntas, for constant k. We start with the observation that k-juntas only have Fourier mass on coefficients of weight at most k. Alternatively, this says that any k-junta can be written as a linear combination of parity functions on at most k variables. (In the language of our previous construction, χ_s such that $wt(s) \leq k$.) Thus we start by running our sanitizer for parity predicates on the set $\mathcal{P}_k$ containing all parity predicates on at most k variables. We have to modify the evaluator function to take into account that not every k-junta predicate has the same bias. Indeed, we cannot control $\chi_{0^d}(\widehat{D})$ in our output, as $\chi_{0^d}(D) = 1$ for any database. Thus our evaluator will apply an affine shift to each result that depends on the junta. Because the evaluator depends on the predicate, the resulting sanitizer no longer outputs symmetric relaxed synthetic data.

The use of a sanitizer for parity queries as a building block to construct a sanitizer for arbitrary k-juntas is inspired by [5], which uses a noisy vector of answers to parity queries as a building block to construct

[4]One form of the Chernoff-Hoeffding Bound states if $X_1, \ldots, X_n$ are independent random variables over $[0,1]$ and $X = (1/n)\sum_{i=1}^{n}$ then $\Pr[|X - \mathbb{E}[X]| \geq t] < 2\exp(-2nt^2)$ [13]

synthetic data for a particular class of k-juntas (conjunctions on k-literals). However, while their sanitizer constructs a standard synthetic database and is inefficient, our construction of symmetric relaxed synthetic data for parity predicates is efficient, and thus our eventual sanitizer for k-juntas will also be efficient.

Consider a predicate $c : \{0,1\}^d \to \{0,1\}$. Then we can take the Fourier expansion

$$c(x) = \sum_{s \in \{0,1\}^d} \widehat{c}(s)\chi_s(x)$$

where

$$\widehat{c}(s) = \mathop{\mathbb{E}}_{x \leftarrow_R \{0,1\}^d}[c(x)\chi_s(x)].$$

The accuracy of our sanitizer relies on the following fact about the Fourier coefficients of k-juntas

Fact 5.16. *If $c : \{0,1\}^d \to \{0,1\}$ is a k-junta, then $wt(s) > k \implies \widehat{c}(s) = 0$*

Let $\mathcal{P}_k = \left\{\chi_s \mid s \in \{0,1\}^d, 1 \le wt(s) \le k\right\}$. Our sanitizer for k-juntas is just the sanitizer for parities applied to the set $\mathcal{P}_k$. We now define the evaluator that computes the answer to conjunction queries from the output of $\mathcal{A}(D, \mathcal{P}_k)$.

Evaluator $\mathcal{E}(\widehat{D}, c)$ for a k-junta c:

return $|\mathcal{P}_k| c(\widehat{D}) - (|\mathcal{P}_k| - 1)\,\widehat{c}(\emptyset)$

Efficiency and privacy follow from the analysis of $\mathcal{A}$. Let $\mathcal{J}_{d,k}$ be a family of all k-juntas on d variables.

Theorem 5.17. *$\mathcal{A}(D, \mathcal{P}_k)$ is (α, β)-accurate for $\mathcal{J}_{d,k}$ using $\mathcal{E}$ when*

$$n \ge \frac{2|\mathcal{P}_k| \log(2|\mathcal{P}_k|/\beta)}{\alpha\epsilon}.$$

Proof. Let $c \in \mathcal{J}_{d,k}$ be a fixed predicate. Assume that $\widehat{D}$ is α-accurate for $\mathcal{P}_k$ using $\mathcal{E}_{\mathcal{P}}$. This event occurs with probability at least $1 - \beta$ by Theorem 5.10 and our assumption on n. We now analyze the quantity $c(\widehat{D})$.

$$\begin{aligned}
c(\widehat{D}) &= \frac{1}{\hat{n}} \sum_{i=1}^{\hat{n}} c(\hat{x}_i) \\
&= \frac{1}{\hat{n}} \sum_{i=1}^{\hat{n}} \sum_{s \in \{0,1\}^d} \widehat{c}(s)\chi_s(\hat{x}_i) \\
&= \sum_{s \in \{0,1\}^d : |s| \le k} \widehat{c}(s)\chi_s(\widehat{D}) && (3) \\
&= \widehat{c}(\emptyset) + \sum_{s \in \{0,1\}^d : 1 \le |s| \le k} \widehat{c}(s)\chi_s(\widehat{D}) && (4) \\
&\le \widehat{c}(\emptyset) + \sum_{s \in \{0,1\}^d : 1 \le |s| \le k} \widehat{c}(s)\left(\frac{\chi_s(D) + \alpha}{|\mathcal{P}_k|}\right) && (5) \\
&\le \frac{1}{|\mathcal{P}_k|}\left((|\mathcal{P}_k| - 1)\,\widehat{c}(\emptyset) + \sum_{s \in \{0,1\}^d : |s| \le k} (\widehat{c}(s)\chi_s(D) + \alpha)\right) \\
&= \frac{1}{|\mathcal{P}_k|}\left((|\mathcal{P}_k - 1)\,\widehat{c}(\emptyset) + c(D)\right) + \alpha
\end{aligned}$$

where step 3 uses Fact 5.16, step 4 uses the fact that $\chi_{0^d}(x) = 1$ everywhere, and step 5 uses the fact that $\widehat{D}$ is α-accurate for $\mathcal{P}_k$ when evaluated by $\mathcal{E}_{\mathcal{P}}$. A similar argument shows that

$$c(\widehat{D}) \geq \frac{1}{|\mathcal{P}_k|}\left(\left(|\mathcal{P}_k| - 1\right)\widehat{c}(\emptyset) + c(D)\right) - \alpha$$

Thus $\widehat{D} = \mathcal{A}(D)$ is α-accurate for $\mathcal{J}_{d,k}$ using $\mathcal{E}$ with probability at least $1 - \beta$.

□

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.

References

[1] ADAM, N. R., AND WORTMANN, J. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys 21* (1989), 515–556.

[2] ALEKHNOVICH, M., BRAVERMAN, M., FELDMAN, V., KLIVANS, A. R., AND PITASSI, T. The complexity of properly learning simple concept classes. In *J. Comput. Syst. Sci.* (2008), vol. 74, pp. 16–34.

[3] ARORA, S., LUND, C., MOTWANI, R., SUDAN, M., AND SZEGEDY, M. Proof verification and the hardness of approximation problems. *J. ACM 45*, 3 (1998), 501–555.

[4] BABAI, L., FORTNOW, L., LEVIN, L. A., AND SZEGEDY, M. Checking computations in polylogarithmic time. In *STOC* (1991), pp. 21–31.

[5] BARAK, B., CHAUDHURI, K., DWORK, C., KALE, S., MCSHERRY, F., AND 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* (2007), pp. 273–282.

[6] BARAK, B., AND GOLDREICH, O. Universal arguments and their applications. In *SIAM J. Comput.* (2008), vol. 38, pp. 1661–1694.

[7] BEN-SASSON, E., GOLDREICH, O., HARSHA, P., SUDAN, M., AND VADHAN, S. P. Robust pcps of proximity, shorter pcps, and applications to coding. In *SIAM J. Comput.* (2006), vol. 36, pp. 889–974.

[8] BLUM, A., DWORK, C., MCSHERRY, F., AND NISSIM, K. Practical privacy: The SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (June 2005).

[9] BLUM, A., LIGETT, K., AND ROTH, A. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th ACM SIGACT Symposium on Thoery of Computing* (2008).

[10] CREIGNOU, N. A dichotomy theorem for maximum generalized satisfiability problems. In *J. Comput. Syst. Sci.* (1995), vol. 51, pp. 511–522.

[11] DINUR, I., AND NISSIM, K. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (2003), pp. 202–210.

[12] DINUR, I., AND REINGOLD, O. Assignment testers: Towards a combinatorial proof of the pcp theorem. In *SIAM J. Comput.* (2006), vol. 36, pp. 975–1024.

[13] DUBHASHI, D. P., AND SEN, S. Concentration of measure for randomized algorithms: techniques and applications. In Handbook of Randomized Algorithms, 2001.

[14] DUNCAN, G. *International Encyclopedia of the Social and Behavioral Sciences*. Elsevier, 2001, ch. Confidentiality and statistical disclosure limitation.

[15] DWORK, C. A firm foundation for private data analysis. *Communications of the ACM (to appear)*.

[16] DWORK, C. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)(2)* (2006), pp. 1–12.

[17] DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference* (2006), pp. 265–284.

[18] DWORK, C., NAOR, M., REINGOLD, O., ROTHBLUM, G., AND 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).

[19] DWORK, C., AND NISSIM, K. Privacy-preserving datamining on vertically partitioned databases. In *Proceedings of CRYPTO 2004* (2004), vol. 3152, pp. 528–544.

[20] DWORK, C., ROTHBLUM, G., AND VADHAN, S. P. Boosting and differential privacy. In *Proceedings of FOCS 2010* (2010).

[21] EVFIMIEVSKI, A., AND GRANDISON, T. *Encyclopedia of Database Technologies and Applications*. Information Science Reference, 2006, ch. Privacy Preserving Data Mining (a short survey).

[22] FELDMAN, V. Hardness of proper learning. In *The Encyclopedia of Algorithms*. Springer-Verlag, 2008.

[23] FELDMAN, V. Hardness of approximate two-level logic minimization and PAC learning with membership queries. *Journal of Computer and System Sciences 75*, 1 (2009), 13–26.

[24] GOLDREICH, O. *Foundations of Cryptography*, vol. 2. Cambridge University Press, 2004.

[25] HÅSTAD, J. Some optimal inapproximability results. In *J. ACM* (2001), vol. 48, pp. 798–859.

[26] KEARNS, M. J., AND VALIANT, L. G. Cryptographic limitations on learning boolean formulae and finite automata. In *J. ACM* (1994), vol. 41, pp. 67–95.

[27] KHANNA, S., SUDAN, M., TREVISAN, L., AND WILLIAMSON, D. P. The approximability of constraint satisfaction problems. In *SIAM J. Comput.* (2000), vol. 30, pp. 1863–1920.

[28] KILIAN, J. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC* (1992).

[29] MICALI, S. Computationally sound proofs. In *SIAM J. Comput.* (2000), vol. 30, pp. 1253–1298.

[30] NAOR, M., AND YUNG, M. Universal one-way hash functions and their cryptographic applications. In *STOC* (1989), pp. 33–43.

[31] PAPADIMITRIOU, C. H., AND YANNAKAKIS, M. Optimization, approximation, and complexity classes. In *J. Comput. Syst. Sci.* (1991), vol. 43, pp. 425–440.

[32] PITT, L., AND VALIANT, L. G. Computational limitations on learning from examples. In *J. ACM* (1988), vol. 35, pp. 965–984.

[33] REITER, J. P., AND 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.

[34] ROMPEL, J. One-way functions are necessary and sufficient for secure signatures. In *STOC* (1990), pp. 387–394.

[35] ROTH, A., AND ROUGHGARDEN, T. Interactive privacy via the median mechanism. In *STOC 2010* (2010).

[36] VALIANT, L. G. A theory of the learnable. *Communications of the ACM 27*, 11 (1984), 1134–1142.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de