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

February 10, 2010

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 PCP-based Levin-reductions from NP search problems to finding approximate solutions to CSPs.

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

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

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, 13, 19]), and recently the theoretical computer science community has developed an appealing new approach to the problem, known as *differential privacy*. (See the surveys [15, 14].[1]).

Differential Privacy. A randomized algorithm $\mathcal{A}$ is defined to be *differentially private* [16] 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 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]. The papers [11, 18, 8, 16] have 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 $\hat{D} = \mathcal{A}(D)$ 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 [29].

[1] The webpage `http://research.microsoft.com/en-us/projects/databaseprivacy/` is also a comprehensive reference

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 $\hat{D}$, 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 $c(x_1,\ldots,x_n) = x_{i_1} \wedge x_{i_2} \wedge \cdots x_{i_k}$ for $i_1,\ldots,i_k \in [d]$, the fraction of rows in $\hat{D}$ that satisfy c equals the fraction of rows in D that satisfy p 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.[2]

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. [17] 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. all 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 [31].

Our Result. 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 $b, b' \in \{0,1\}$.)

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

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.

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.

Our proof is obtained by combining the hard-to-sanitize databases of Dwork et al. [17] 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 counting queries 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 to forge a signature.

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 [25, 26, 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, 20] and cryptographic assumptions for hardness of representation-independent learning [31, 23]).

- 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)**.** [16] 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 $\mathcal{C}$ of boolean predicates $c : \{0,1\}^d \to \{0,1\}$. Then each predicate c induces a *counting query* that on database $D = (x_1, \ldots, x_n) \in (\{0,1\}^d)^n$ returns

$$c(D) = \frac{|\{i \in [n] \mid c(x_i) = 1\}|}{|D|}$$

If the output of $\mathcal{A}$ is a synthetic database $\hat{D}$, then $c(\mathcal{A}(D))$ is simply the fraction of rows of $\hat{D}$ 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 efficiently computable 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 $c \in \mathcal{C}$. $\mathcal{E}(Z, c)$ 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 $c(\mathcal{E}(\mathcal{A}(D), c))$.

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 $c(D)$. Specifically

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 use $f(n) = \text{negl}(n)$ if $f(n) = o(n^{-c})$ for every $c > 0$ and say that $f(n)$ is *negligible*.

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 [17], we define the following notions

Definition 2.3 (Database Distribution Ensemble)**.** Let $\mathcal{D} = \mathcal{D}_d$ be an ensemble of distributions on tuples (D, aux), where $D \in (\{0,1\}^d)^{n+1}$ is a d-column databases with $n+1$ rows, for $n = n(d)$, and $aux \in \{0,1\}^{\text{poly}(d)}$ is a string of *auxiliary information about* D. Let $(D, D', i, aux) \leftarrow_{\text{R}} \tilde{\mathcal{D}}$ denote the experiment in which we choose a tuple $(D_0, aux) \leftarrow_{\text{R}} \mathcal{D}$ 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 $\mathcal{T}(\mathcal{A}(D), aux)$ outputs a row of D:

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

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

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

where x_i is the i-th row of D.

In [17], 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. *[17] If a distribution ensemble $\mathcal{D} = \mathcal{D}_d$ on $n(d)$-row databases and auxiliary information 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))**.** A *family of CSPs*, denoted Γ, is a set of boolean predicates on q variables. For every $d \geq q$, let $\mathcal{C}_\Gamma^{(d)}$ be the class consisting of all predicates $c : \{0,1\}^d \to \mathbb{R}$ of the form $c(u_1, \ldots, u_d) = \gamma(u_{i_1}, \ldots, u_{i_q})$. We call $\mathcal{C}_\Gamma = \cup_{d=q}^{\infty} \mathcal{C}_\Gamma^{(d)}$ 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

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

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 3.2 (inapproximability under Karp reductions)**.** A family of CSPs Γ is *α-hard-to-approximate under Karp reductions* if there exists $\gamma \in [0, 1 - \alpha]$, and a polynomial-time computable function R such that for every C with input size $\overline{d}$, if we set $\varphi_C = R(C) \subseteq \mathcal{C}_\Gamma$, then

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

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 3.3 (inapproximability under Levin reductions)**.** A family of CSPs Γ is *α-hard-to-approximate under Levin reductions* if there exists $\gamma \in [0, 1-\alpha]$ and polynomial-time computable functions R, Enc, Dec such that for every C with input of size $\overline{d}$ if we set $\varphi_C = R(C) \subseteq \mathcal{C}_\Gamma$ then

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

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 3.3. We now argue that for every circuit C, the formula $\varphi_C = R(C)$ satisfies conditions 1 and 2 of Definition 3.2. 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.

It follows from the PCP Theorem 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 3.4. *For every family of CSPs Γ that is closed under negation and contains a function that depends on at least two variables, there is a constant $\alpha = \alpha(\Gamma)$ such that either 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. [24]. 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 holds 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 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 [28] 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 [24] 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 [22]) 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.)

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 [17], 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 3.3, 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. $(sk, vk) \leftarrow_R Gen(1^\kappa)$.
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 [21] it is shown how to modify known constructions [27, 30] to obtain a super-secure digital signature scheme from any one-way function.

4.2 Main Hardness Result

We are now ready to now state and prove our hardness result. Let Γ be a family of 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. *For every CSP Γ such that $\Gamma \cup \neg\Gamma$ is α-hard-to-approximate under Levin reductions, and every polynomial $n(d)$, there exists a distribution ensemble $\mathcal{D} = \mathcal{D}_d$ on $n(d)$-row databases that is $(\alpha, \mathcal{C}_\Gamma^{(d)})$-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 $\gamma \in [0, 1-\alpha]$ be the constant from Definition 3.3. Let $\kappa = d^\tau$ for a constant $\tau > 0$ to be defined later.

Let $n = n(d) = \text{poly}(d)$ and use $\ell(s_1)$ to denote the length of the string s_1, $\mathcal{U}_S$ to denote the uniform distribution over the set S, and $s_1 \| s_2$ to denote the concatenation of s_1 and s_2. 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:

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

$(sk, vk) \leftarrow_{\text{R}} Gen(1^\kappa)$
$(m_1, \ldots, m_{n+1}) \leftarrow_{\text{R}} \mathcal{U}_{(\{0,1\}^\kappa)^{n+1}}$
for $i = 1$ to $n+1$ **do**
  $x'_i := Enc(m_i \| Sign_{sk}(m_i), C_{vk})$
  $x_i := x'_i \parallel 0^{d - \ell(x'_i)}$
end for
$D_0 := (x_1, \ldots, x_{n+1})$
return (D_0, vk)

Since $\ell(x'_i) = \text{poly}(\kappa) = \text{poly}(d^\tau)$, we can choose the constant $\tau > 0$ to be small enough so that $\ell(x'_i) < d$, 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 $\varphi_{C_{vk}} = R(C_{vk})$.

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

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

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

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. Formally, we define the privacy adversary by means of a subroutine that tries to PCP-decode each row of the input database:

Subroutine $\mathcal{T}_0(\hat{D}, vk)$:

$(\hat{x}_1, \ldots, \hat{x}_{\hat{n}}) := \hat{D}$
$\varphi_{C_{vk}} = R(C_{vk})$
for $i = 1$ to $\hat{n}$ **do**
  if $\text{val}(\varphi_{C_{vk}}, \hat{x}_i) \geq \gamma - \alpha$ **then**
    return $Dec(\hat{x}_i, C_{vk})$
  end if
end for
return $\perp$

Privacy Adversary $\mathcal{T}(\hat{D}, vk)$:

return $Enc(\mathcal{T}_0(\hat{D}, vk), C_{vk})$

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

Claim 4.6. *If $\hat{D} = \mathcal{A}(D)$ is α-accurate for $\mathcal{C}_\Gamma^{(d)}$, then $\mathcal{T}_0(\hat{D}, vk)$ outputs a pair (m, σ) s.t. $C_{vk}(m, \sigma) = 1$.*

Proof. First we show that if $\hat{D}$ is α-accurate, then $\mathcal{T}_0(\hat{D}, vk) \neq \perp$. Since every $(m_i, Sign_{sk}(m_i))$ pair is a satisfying assignment to C_{vk}, the definition of Enc (Definition 3.3) implies that each row x_i of D has $\text{val}(\varphi_{C_{vk}}, x_i) \geq \gamma$. Thus if $\varphi_{C_{vk}} = \{\varphi_1, \ldots, \varphi_m\}$, then

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

Since $\hat{D}$ is α-accurate, then for every constraint $\varphi_j \in \varphi_{C_{vk}}$, we have $\varphi_j(\hat{D}) \geq \varphi_j(D) - \alpha$. Thus

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

So for at least one row $\hat{x} \in \hat{D}$ it must be the case that $\text{val}(\varphi_{C_{vk}}, \hat{x}) \geq \gamma - \alpha$. The definition of Dec (Definition 3.3) implies $C_{vk}(Dec(\hat{x}, C_{vk})) = 1$. □

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

Notice that running $\mathcal{F}$ in the weak forgery game is equivalent to running $\mathcal{T}$ in the experiment of inequality (1), except that $\mathcal{F}$ does not re-encode the output of $\mathcal{T}_0(\mathcal{A}(D), vk)$. 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 $\mathcal{T}(\mathcal{A}(D), vk) = Enc(\hat{x}^*, C_{vk}) \in D$ (except with negligible probability). Thus, we have

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

which is equivalent to the statement of the lemma. □

Lemma 4.7.

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

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. □

One remark about our proof is that the construction of our signature-forging adversary works even if the sanitizer $\mathcal{A}$ gets the verification key as auxiliary input. This extra generality makes our negative result stronger, but auxiliary input was omitted from the definition of the sanitizer to maintain consistency with the standard definition of differential privacy.

Theorem 1.1 in the introduction follows by combining Theorems 3.4 and 4.4.

Acknowledgments

We thank Boaz Barak, Irit Dinur, Cynthia Dwork, Vitaly Feldman, Oded Goldreich, Johan Håstad, Valentine Kabanets, Dana Moshkovitz, Anup Rao, 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. In *Electronic Colloquium on Computational Complexity (ECCC)* (1998), vol. 5.

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

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

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

[16] 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.

[17] 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).

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

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

[20] 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.

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

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

[23] 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.

[24] 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.

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

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

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

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

[29] 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.

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

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