Differential Privacy with Imperfect Randomness

Yevgeniy Dodis*
New York University

Adriana López-Alt
New York University

Ilya Mironov
Microsoft Research

Salil Vadhan†
Harvard University

May 31, 2012

Abstract

In this work we revisit the question of basing cryptography on imperfect randomness. Bosley and Dodis (TCC'07) showed that if a source of randomness $\mathcal{R}$ is "good enough" to generate a secret key capable of encrypting k bits, then one can deterministically extract nearly k almost uniform bits from $\mathcal{R}$, suggesting that traditional privacy notions (namely, indistinguishability of encryption) requires an "extractable" source of randomness. Other, even stronger impossibility results are known for achieving privacy under specific "non-extractable" sources of randomness, such as the γ-Santha-Vazirani (SV) source, where each next bit has fresh entropy, but is allowed to have a small bias $\gamma < 1$ (possibly depending on prior bits).

We ask whether similar negative results also hold for a more recent notion of privacy called *differential privacy* (Dwork et al., TCC'06), concentrating, in particular, on achieving differential privacy with the Santha-Vazirani source. We show that the answer is *no*. Specifically, we give a differentially private mechanism for approximating arbitrary "low sensitivity" functions that works even with randomness coming from a γ-Santha-Vazirani source, for any $\gamma < 1$. This provides a somewhat surprising "separation" between traditional privacy and differential privacy with respect to imperfect randomness.

Interestingly, the design of our mechanism is quite different from the traditional "additive-noise" mechanisms (e.g., Laplace mechanism) successfully utilized to achieve differential privacy with perfect randomness. Indeed, we show that *any* (non-trivial) "SV-robust" mechanism for our problem requires a demanding property called *consistent sampling*, which is strictly stronger than differential privacy, and cannot be satisfied by any additive-noise mechanism.

*Supported by NSF Grants CNS-1065134, CNS-1065288, CNS-1017471, CNS-0831299 and Google Faculty Award.

†Supported by a gift from Google, Inc. Work done in part while on leave as a Visiting Researcher at Microsoft Research SVC and a Visiting Scholar at Stanford University.

1 Introduction

Most cryptographic algorithms require randomness (for example, to generate their keys, probabilistically encrypt messages, etc.). Usually, one assumes that perfect randomness is available, but in many situations this assumption is problematic, and one has to deal with more realistic, "imperfect" sources of randomness $\mathcal{R}$. Of course, if one can (deterministically) extract nearly perfect randomness from $\mathcal{R}$, then one can easily implement traditional cryptographic schemes with $\mathcal{R}$. Unfortunately, many natural sources are not extractable [SV86, CG88, Zuc96]. The simplest example of such a source is the Santha-Vazirani (SV) source [SV86], which produces an infinite sequence of (possibly correlated) bits $\mathbf{x} = x_1, x_2, \ldots$, with the property that $\Pr[x_i = 0 \mid x_1 \ldots x_{i-1}] \in [\frac{1}{2}(1-\gamma), \frac{1}{2}(1+\gamma)]$, for any setting of the prior bits $x_1 \ldots x_{i-1}$. Namely, each bit has almost one bit of fresh entropy, but can have a small bias $\gamma < 1$ (possibly dependent on the prior bits). Yet, the celebrated result of Santha and Vazirani [SV86] showed that there exists no deterministic extractor $\mathsf{Ext}\colon \{0,1\}^n \to \{0,1\}$ capable of extracting even a *single* bit of bias *strictly* less than γ from the γ-SV source, irrespective of how many SV bits $x_1 \ldots x_n$ it is willing to wait for. In particular, outputting the first bit is already optimal in terms of traditional extraction.

Despite this pessimistic result, ruling out the "black-box compiler" from perfect to imperfect (e.g., SV) randomness for *all* applications, one may still hope that specific "non-extractable" sources, such as SV-sources, might be sufficient for *concrete* applications, such as simulating probabilistic algorithms or cryptography. Indeed, a series of celebrated results [VV85, SV86, CG88, Zuc96, ACRT99] showed that very "weak" sources (including SV-sources and much more) are sufficient for simulating probabilistic polynomial-time algorithms; namely, for problems which do not inherently need randomness, but which could potentially be sped up using randomization. Moreover, even in the area of cryptography — where randomness is *essential* (e.g., for key generation) — it turns out that many "non-extractable" sources (again, including SV sources and more) are sufficient for *authentication* applications, such as the designs of MACs [MW97, DKRS06] and even signature schemes [DOPS04] (under appropriate hardness assumptions). Intuitively, the reason for the latter "success story" is that authentication applications only require that it is hard for the attacker to completely guess (i.e., "forge") some long string, so having (min-)entropy in our source $\mathcal{R}$ should be sufficient to achieve this goal.

Privacy with Imperfect Randomness? In contrast, the situation appears to be much less bright when dealing with *privacy* applications, such as encryption, commitment, zero-knowledge, etc. First, McInnes and Pinkas [MP90] showed that unconditionally secure symmetric encryption cannot be based on SV sources, even if one is restricted to encrypting a single bit. This result was subsequently strengthened by Dodis et al. [DOPS04], who showed that SV sources are not sufficient for building even computationally secure encryption (again, even of a single bit), and, if fact, essentially any other cryptographic task involving "privacy" (e.g., commitment, zero-knowledge, secret sharing, etc.). Finally, Bosley and Dodis [BD07] showed an even more general result: if a source of randomness $\mathcal{R}$ is "good enough" to generate a secret key capable of encrypting k bits, then one can deterministically extract nearly k almost uniform bits from $\mathcal{R}$, suggesting that traditional privacy *requires* an "extractable" source of randomness. [1]

In this work we ask the question if similar pessimistic conclusions also hold for a more recent, but already very influential variant of privacy called *differential privacy* (DP), introduced by Dwork et al. [DMNS06], concentrating in particular on achieving differential privacy with the simple Santha-Vazirani source.

Main Question: Is it possible to achieve (non-trivial) differential privacy with SV-sources?

[1] On the positive side, [DS02], [BD07] showed that extractable sources are not strictly necessary for encrypting a "very small" number of bits. Still, for natural "non-extractable" sources, such as SV sources, it is known that encrypting even a single bit is impossible [SV86, DOPS04].

As our main result, we give a *positive* answer to this question, showing a somewhat surprising "separation" between traditional privacy and differential privacy. But, first, let us examine the above question more closely, gradually explaining the path towards our solution.

Differential Privacy. Differential privacy was introduced for the purposes of allowing the owner of a sensitive database D to securely release some "aggregate statistics" $f(D)$ while protecting the privacy of individual users whose data is in D. Unfortunately, revealing $f(D)$ by itself might violate the privacy of some individual records, especially if the attacker has a some partial information about D. Instead, we wish to design a *randomized mechanism* $M(D;\mathbf{r})$ which will approximate $f(D)$ with relatively high accuracy, but will use its randomness $\mathbf{r}$ to "add enough noise" to the true answer $f(D)$ to protect the privacy of the *individual* records of D. For simplicity, we will restrict our attention to real-valued queries f, so that we can define the *utility* ρ of M as the expected value (over uniform $\mathbf{r}$, for now) of $|f(D) - M(D;\mathbf{r})|$, which we want to minimize. For example, f might be a *counting query*, where $f(D)$ is the number of records in D satisfying som predicate π, in which case we seek to achieve utility $o(|D|)$ or even independent of $|D|$. More interestingly, we want M to satisfy the following very strong notion called *ε-differential privacy*: for any *neighboring databases* D_1 and D_2 (i.e. D_1 and D_2 differ on a single record) and for any potential output z, $\Pr_{\mathbf{r}}[M(D_1;\mathbf{r}) = z]/\Pr_{\mathbf{r}}[M(D_2;\mathbf{r}) = z]$ is between $e^{-\varepsilon} \approx 1-\varepsilon$ and $e^{\varepsilon} \approx 1+\varepsilon$ (assuming ε is close to 0). This definition shows one difference between standard privacy, which holds between *all* pairs of databases D_1 and D_2, and differential privacy, which only holds for *neighboring* databases. Related to the above, one cannot achieve any useful utility ρ if ε is required to be negligibly small (as then one can gradually transfer any D_1 to any other D_2 without noticeably changing the answers given by M). Instead, the one typically assumes that ε is a small constant *which can be pushed arbitrarily close to* 0, possibly at the expense of worse utility ρ. Motivated by these considerations, we will say that f admits a class of *non-trivial* mechanisms $\mathcal{M} = \{M_\varepsilon \mid \varepsilon > 0\}$ if there exists some fixed function $g(\cdot)$ s.t., for *all* $\varepsilon > 0$, M_ε is ε-DP and has utility $g(\varepsilon)$, independent of the size of the database D.

Additive-Noise Mechanisms. The simplest class of non-trivial differentially private mechanisms (with perfect randomness) are the so called *additive-noise mechanisms* [DMNS06, GRS09, HT10], introduced in the original work of [**?**, **?**, **?**, DMNS06]. These mechanisms have the form $M(D;\mathbf{r}) = f(D) + X(\mathbf{r})$, where X is an appropriately chosen "noise" distribution added to guarantee ε-DP. For example, for counting queries (and more general "low-sensitivity" queries where $|f(D_1) - f(D_2)|$ is bounded on all neighboring databases D_1 and D_2), the right distribution is the *Laplace* distribution with standard deviation $\Theta(1/\varepsilon)$ [DMNS06], giving the (additive-noise) Laplace mechanism for such functions, which is private and accurate (in fact, essentially optimal for a wide range of loss functions [GRS09]). One perceived advantage of additive-noise mechanisms comes from the fact that the noise is oblivious to the input, and it is natural to ask if it is possible to design additive-noise mechanisms which would be non-trivial even if the noise distribution is generated using the Santha-Vazirani source. For example, perhaps one can generate a "good enough" sample of the Laplace distribution even with SV sources? Unfortunately, we show that this is not the case. In fact, any non-trivial additive-noise mechanism for a source $\mathcal{R}$ implies the existence of a randomness extractor for $\mathcal{R}$, essentially collapsing the notion of differential privacy to that of traditional privacy, and showing the impossibility of non-trivial additive-noise mechanisms for SV sources.

Need for Consistent Sampling. In fact, the main reason why additive-noise mechanisms failed to handle SV sources comes from the fact that such algorithms use *disjoint sets* of coins to produce the same "noisy answer" on two databases having different "real answers". More formally, if $f(D_1) \neq f(D_2)$ and $T_i(z)$ is the set of coins $\mathbf{r}$ where $M(D_i;\mathbf{r}) = z$, an additive-noise mechanism must satisfy $T_1(z) \cap T_2(z) = \emptyset$. On the other hand, ε-DP requires that $\Pr[\mathbf{r} \in T_1(z)]/\Pr[\mathbf{r} \in T_2(z)] \leq 1+\varepsilon$. For the uniform distribution, this simply means that $|T_1| \approx |T_2|$. Since T_1 and T_2 are disjoint, the SV

adversary can try to bias the coins $\mathbf{r}$ so as *simultaneously* increase (or, at least maintain) the odds of hitting T_1, while decreasing the odds of hitting T_2. Indeed, in Lemma 2.4 we show that an SV adversary can always succeed in amplifying the ratio $\Pr[\mathbf{r} \in T_1]/\Pr[\mathbf{r} \in T_2]$ (and, hence, violate the differential privacy of our mechanism) whenever T_1 and T_2 have small intersection (e.g., are disjoint).

In fact, in Lemma 3.2 we prove that any "SV-robust" mechanism should strive to produce *identical* outputs on neighboring databases *for a majority of random tapes*; in particular, for any z, $|T_1(z) \cap T_2(z)| \approx |T_1(z)| \approx |T_2(z)|$ (see Definition 4.1 for the exact quantitative formulation). This general property, which we call *consistent sampling* (CS), is closely related to the "consistent sampling" methodology that has found applications in web search [BGMZ97] and parallel repetition theorems [Hol07], among others. Moreover, we show that ε-consistent sampling implies ε-differential privacy, but the converse is false.

Our Main Result. The lower bound above suggests a path forward toward building SV-robust mechanisms, which starts with the design of consistently samplable mechanisms. For example, the classical Laplace mechanism for low sensitivity functions could be viewed as sampling some noise x of expected magnitude $\rho = O(1/\varepsilon)$, and adding it to the exact solution $y = f(D)$. Being additive-noise, this mechanism is not CS. But, imagine a new mechanism which further rounds the answer $z = y + x$ to the nearest multiple z' of $1/\varepsilon$. Clearly, the expected utility has gone from ρ to at most $\rho' = \rho + 1/\varepsilon = O(\rho)$. Yet, it turns out that the new mechanism is now ε-CS, since, informally, the *rounded* answers on neighboring databases are only distinct on an ε-fraction of coins $\mathbf{r}$ (see Section 5).

Still, designing CS mechanisms was only a *necessary* condition for building SV-robust, differentially private mechanisms. For example, the basic notion of consistency ignores the binary representations of random coins $\mathbf{r}$ defining the needed pre-image sets T_1 and T_2, which are (intuitively) very important for handling SV sources since their randomness properties are bit-by-bit. Indeed, we show that consistency alone is *not* enough for SV-robustness, and we need an additional (fortunately, simply stated) property of our sampling to guarantee the latter. (As expected, this property asks something quite natural about the binary representations of the coins inside T_1 and T_2.) We call the resulting notion *SV-consistent sampling* (SVCS; Definition 4.3). Building a non-trivial mechanism satisfying this condition formed the main technical bulk of our work.

In particular, starting with the "rounded" Laplace mechanism, we show a careful implementation of this CS mechanism, so that the resulting mechanism is actually SVCS (with appropriate parameters guaranteeing ε-DP of the final mechanism against γ-SV sources). The details of this technical step, which uses properties of *arithmetic coding* (see [MNW98, WNC87]) applied to the specific Laplace distribution, are explained in Section 5. This gives us our main result (Theorem 5.3) and an affirmative answer to our Main Question: a non-trivial class of SV-robust mechanisms for counting queries and arbitrary low-sensitivity functions.

To maintain a clear presentation, we defer more technical proofs to Appendix A.

2 Random Sources and Differential Privacy

Notation. For a positive integer n, we use the notation $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use $\lfloor \cdot \rceil$ to denote the nearest integer function. For a distribution or random variable R, we write $r \leftarrow \mathsf{R}$ to denote the operation of sampling a random r according to R. For a randomized function h, we write $h(x\,; r)$ to denote the unique output of f on input x with random coins r. When the distribution of random coins R is understood from context, we write $h(x)$ to denote the random variable $h(x\,; r)$ for $r \leftarrow \mathsf{R}$. Finally, we denote a sequence of bits using boldface, e.g. $\mathbf{x} = x_1, x_2, \ldots$

We use calligraphic letters to denote families of the corresponding letter. For example, $\mathcal{F}$ denotes a family of functions f, $\mathcal{R}$ denotes a family of distributions R. We see a distribution over $\{0,1\}^*$ as continuously outputting (possibly correlated) bits. In particular, we let U be the distribution over $\{0,1\}^*$ that samples each bit independently and uniformly at random. When U is truncated after n

bits, the result is the distribution U_n, which is the uniform distribution over $\{0,1\}^n$, the bit-strings of length n.

2.1 Random Sources

We call a family $\mathcal{R}$ of distributions over $\{0,1\}^*$ a *source*. In this work, we model perfect randomness with the uniform source and imperfect randomness with the γ-Santha-Vazirani source [SV86], arguably the simplest type of a "non-extractable" source. The *uniform source* $\mathcal{U} \stackrel{\mathsf{def}}{=} \{\mathsf{U}\}$ is the set containing only the distribution U on $\{0,1\}^*$ that samples each bit uniformly at random. We define γ-Santha-Vazirani sources below.

Definition 2.1 (γ-Santha-Vazirani Source [SV86])**.** *Let $\gamma \in [0,1]$. A probability distribution $\mathsf{X} = (\mathsf{X}_1, \mathsf{X}_2, \ldots)$ over $\{0,1\}^*$ is a* γ-Santha-Vazirani distribution *if for all $i \in \mathbb{Z}^+$ and $x_1 \ldots x_{i-1} \in \{0,1\}^{i-1}$, it holds that*

$$\frac{1}{2}(1-\gamma) \leq \Pr[\mathsf{X}_i = 0 \mid \mathsf{X}_1 = x_1, \ldots \mathsf{X}_{i-1} = x_{i-1}] \leq \frac{1}{2}(1+\gamma).$$

We define the γ-Santha-Vazirani source $\mathcal{SV}(\gamma)$ *to be the set of all γ-Santha-Vazirani distributions. Finally, for a distribution $\mathsf{SV}(\gamma) \in \mathcal{SV}(\gamma)$, we let $\mathsf{SV}(\gamma, n)$ be the distribution $\mathsf{SV}(\gamma)$ restricted to the first n coins $(\mathsf{X}_1, \ldots, \mathsf{X}_n)$. We let $\mathcal{SV}(\gamma, n)$ be the set of all distributions $\mathsf{SV}(\gamma, n)$.*

We now define γ-biased semi-flat sources, which were introduced by [RVW04] (see also [DOPS04], where they were referred to as γ-biased halfspace sources).

Definition 2.2 (γ-Biased Semi-Flat Source)**.** *For $S \subset \{0,1\}^n$ of size $|S| = 2^{n-1}$, and $\gamma \in [0,1]$, the distribution $\mathsf{H}_S(\gamma, n)$ over $\{0,1\}^n$ is defined as follows: for all $x \in S$, $\Pr_{x \leftarrow \mathsf{H}_S(\gamma,n)}[x] = (1+\gamma) \cdot 2^{-n}$, and for all $x \notin S$, $\Pr_{x \leftarrow \mathsf{H}_S(\gamma,n)}[x] = (1-\gamma) \cdot 2^{-n}$. We define the* γ-biased semi-flat source $\mathcal{H}(\gamma, n)$ *to be the set of all distributions $\mathsf{H}_S(\gamma, n)$ for all $|S| = 2^{n-1}$.*

Lemma 2.3 ([RVW04, DOPS04])**.** *For any $n \in \mathbb{Z}^+$ and $\gamma \in [0,1]$, $\mathcal{H}(\gamma, n) \subset \mathcal{SV}(\gamma, n)$.*

We prove a general lemma about γ-semi-flat sources, which will be very useful in later sections.

Lemma 2.4. *Let $G, B \subseteq \{0,1\}^n$ such that $|G| \geq |B| > 0$, and let $\sigma \stackrel{\mathsf{def}}{=} \frac{|B \setminus G|}{|B|} \in [0,1]$. Then there exists $S \subseteq \{0,1\}^n$ of size $|S| = 2^{n-1}$ such that*

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{H}_S(\gamma,n)}[\mathbf{r} \in G]}{\Pr_{\mathbf{r} \leftarrow \mathsf{H}_S(\gamma,n)}[\mathbf{r} \in B]} \geq (1 + \gamma\sigma) \cdot \frac{|G|}{|B|}.$$

Proof. Let $G' \stackrel{\mathsf{def}}{=} G \setminus B$, $B' \stackrel{\mathsf{def}}{=} B \setminus G$, and $N \stackrel{\mathsf{def}}{=} G \cap B$. Let $\alpha \stackrel{\mathsf{def}}{=} |G'|, \beta \stackrel{\mathsf{def}}{=} |B'|$, and $\lambda \stackrel{\mathsf{def}}{=} |N|$. We consider two cases: 1. $\alpha \leq 2^{n-1}$, and 2. $\alpha > 2^{n-1}$

Case 1: First suppose that $\alpha \leq 2^{n-1}$, which means also $\beta \leq 2^{n-1}$ since we assume $\alpha \geq \beta$. Then pick any $S \subset \{0,1\}^n$ of size $|S| = 2^{n-1}$ such that $G' \subseteq S$ and $B' \cap S = \emptyset$. Let $\lambda_0 \stackrel{\mathsf{def}}{=} |S \cap N|$ and $\lambda_1 \stackrel{\mathsf{def}}{=} \lambda - \lambda_0$. Then

$$\begin{aligned}
\frac{\Pr_{x \leftarrow \mathsf{H}_S(\gamma,n)}[x \in G]}{\Pr_{x \leftarrow \mathsf{H}_S(\gamma,n)}[x \in B]} &= \frac{(1+\gamma)\alpha + (1+\gamma)\lambda_0 + (1-\gamma)\lambda_1}{(1-\gamma)\beta + (1+\gamma)\lambda_0 + (1-\gamma)\lambda_1} \\
&\geq \frac{(1+\gamma)\alpha + (1+\gamma)\lambda}{(1-\gamma)\beta + (1+\gamma)\lambda} \\
&= \frac{\alpha+\lambda}{\beta+\lambda}\left(\frac{\beta+\lambda}{\left(\frac{1-\gamma}{1+\gamma}\right)\beta + \lambda}\right) = \frac{\alpha+\lambda}{\beta+\lambda} \cdot \frac{1}{\Delta}, \qquad (2.1)
\end{aligned}$$

where $\Delta \stackrel{\mathsf{def}}{=} \frac{\left(\frac{1-\gamma}{1+\gamma}\right)\beta+\lambda}{\beta+\lambda}$. We have,

$$\Delta = \frac{\left(\frac{1-\gamma}{1+\gamma}\right)\beta+\lambda}{\beta+\lambda} = 1 - \frac{\left(\frac{2\gamma}{1+\gamma}\right)\beta}{\beta+\lambda} = 1 - \frac{2\gamma\sigma}{1+\gamma} = \frac{1+\gamma-2\gamma\sigma}{1+\gamma}.$$

Then, plugging in this value of Δ in (2.1), we have

$$\begin{aligned}\frac{\Pr_{x\leftarrow \mathsf{H}_S(\gamma,n)}[x\in G]}{\Pr_{x\leftarrow \mathsf{H}_S(\gamma,n)}[x\in B]} &\geq \frac{\alpha+\lambda}{\beta+\lambda}\left(\frac{1+\gamma}{1+\gamma-2\gamma\sigma}\right) = \frac{\alpha+\lambda}{\beta+\lambda}\left(1+\frac{2\gamma\sigma}{1+\gamma-2\gamma\sigma}\right)\\ &\geq \frac{\alpha+\lambda}{\beta+\lambda}\left(1+\gamma\sigma\right),\end{aligned}$$

where the last inequality follows from $\sigma < 1$.

Case 2: Now assume that $\alpha > 2^{n-1}$. Then pick any $S \subset \{0,1\}^n$ of size $|S| = 2^{n-1}$ such that $S \subseteq G'$. Let $\alpha_0 \stackrel{\mathsf{def}}{=} |S \cap G'| = 2^{n-1}$ and $\alpha_1 \stackrel{\mathsf{def}}{=} \alpha - \alpha_0$. Then

$$\begin{aligned}\frac{\Pr_{x\leftarrow \mathsf{H}_S(\gamma,n)}[x\in G]}{\Pr_{x\leftarrow \mathsf{H}_S(\gamma,n)}[x\in B]} &= \frac{(1+\gamma)\alpha_0+(1-\gamma)\alpha_1+(1-\gamma)\lambda}{(1-\gamma)\beta+(1-\gamma)\lambda}\\ &= \frac{\left(\frac{1+\gamma}{1-\gamma}\right)\alpha_0+\alpha_1+\lambda}{\beta+\lambda} = \frac{\alpha_0+\alpha_1+\lambda}{\beta+\lambda} + \frac{\left(\frac{2\gamma}{1-\gamma}\right)\alpha_0}{\beta+\lambda}\\ &\geq \frac{\alpha_0+\alpha_1+\lambda}{\beta+\lambda}\left(1+\left(\frac{2\gamma}{1-\gamma}\right)\frac{1}{2}\right) = \frac{\alpha+\lambda}{\beta+\lambda}\left(1+\frac{\gamma}{1-\gamma}\right)\\ &\geq \frac{\alpha+\lambda}{\beta+\lambda}\cdot(1+\gamma)\\ &\geq \frac{\alpha+\lambda}{\beta+\lambda}\cdot(1+\sigma\gamma).\end{aligned}$$

□

2.2 Differential Privacy and Utility

We start by briefly recalling the notion of differential privacy. Given a database containing confidential information, we wish to allow learning of statistical information about the contents of the database without violating the privacy of any of its individual entries. The standard cryptographic notion of privacy where negligible information is revealed, is not appropriate in this setting as it does not allow to learn even one bit of "global" information about the contents of the database. Therefore, a new privacy definition is needed for this setting, in particular, one that allows a better trade-off between privacy and utility. This is precisely what differential privacy achieves.

The Model. We model a *statistical database* as an array of rows, and say that two databases are *neighboring* if they differ in exactly one row. Throughout the paper, we let $\mathcal{D}$ be the space of all databases. We consider the *interactive* setting, in which interested parties submit queries, modeled as functions $f : \mathcal{D} \to \mathcal{Z}$, where $\mathcal{Z}$ is a specified range. In this paper, we are only concerned with queries with range $\mathcal{Z} = \mathbb{Z}$, and henceforth only consider this case. A *mechanism* M is a probabilistic algorithm that takes as input a database $D \in \mathcal{D}$ and a query $f : \mathcal{D} \to \mathbb{Z}$, and outputs a value $z \in \mathbb{Z}$. We assume that M's random tape is in $\{0,1\}^*$, that is, that M has at its disposal a possibly infinite number of random bits, but for a fixed outcome $z \in \mathbb{Z}$, M needs only a finite number of coins $n = n(D, f, z)$ to determine whether $M(D, f) = z$. Furthermore, we assume that if $\mathbf{r} \in \{0,1\}^n$ is a prefix of $\mathbf{r}' \in \{0,1\}^{n'}$ and $M(D, f\ ; \mathbf{r}) = z$ is already determined from $\mathbf{r}$, then $M(D, f\ ; \mathbf{r}') = z$ also. In other words, providing M with extra coins does not change its output.

Definitions. Informally, we wish $z = M(D, f)$ to approximate the true answer $f(D)$ without revealing too much information. We say a mechanism is *differentially private* for a class of queries $\mathcal{F}$ if for all queries $f \in \mathcal{F}$, replacing a real entry in the database with one containing fake information only changes the outcome of the mechanism by a small amount. In other words, evaluating the mechanism on the same query $f \in \mathcal{F}$, on two neighboring databases, does not change the output by much. On the other hand, we define its utility to be the expected difference between the true answer $f(D)$ and the output of the mechanism. Since the purpose of this work is to analyze mechanisms with respect to their sources of randomness, the following definitions of privacy and utility explicitly take the source of randomness $\mathcal{R}$ into account.

Definition 2.5 ($(\varepsilon, \mathcal{R})$-Differential Privacy)**.** *Let $\varepsilon \geq 0$, $\mathcal{R}$ be a source, and $\mathcal{F} = \{f : \mathcal{D} \to \mathbb{Z}\}$ be a class of functions. A mechanism M is* $(\varepsilon, \mathcal{R})$-differentially private *for $\mathcal{F}$ if for any pair $D_1, D_2 \in \mathcal{D}$ of neighboring databases, all $f \in \mathcal{F}$, all possible outputs $z \in \mathbb{Z}$ of M, and all $\mathsf{R} \in \mathcal{R}$:*

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{R}}[M(D_1, f\ ; \mathbf{r}) = z]}{\Pr_{\mathbf{r} \leftarrow \mathsf{R}}[M(D_2, f\ ; \mathbf{r}) = z]} \leq 1 + \varepsilon.$$

This is a very strong definition. Not only does it give a *statistical* guarantee, making it independent of the computation power of any adversary, but it is also strictly stronger than the requirement that the statistical distance between $M(D_1, f)$ and $M(D_2, f)$ is at most ε (for example, the latter allows some low-probability outcomes of $M(D_1, f)$ to never occur under $M(D_2, f)$). We also note that, traditionally, differential privacy has been defined by having the ratio of probabilities be bounded by e^{ε}. We instead bound it by $1+\varepsilon$, since this formulation makes some of our calculations slightly cleaner. This is fine since we always have $1 + \varepsilon \leq e^{\varepsilon}$, and, when $\varepsilon \in [0, 1]$ (which is the key range of interest), we anyway have $e^{\varepsilon} \approx 1 + \varepsilon$.

If a mechanism M is $(\varepsilon, \mathcal{R})$-differentially private for some randomness source $\mathcal{R}$, then a mechanism M' that runs M as a black box and then performs some post-processing on the output, is also $(\varepsilon, \mathcal{R})$-differentially private. Intuitively, this is because given only $z = M(D, f)$, M' cannot reveal more information about D than z itself. In our work we only consider the case where M' evaluates a *deterministic* function h of $z = M(D, f)$, so that M and h do not have to "share" the random source $\mathcal{R}$.

Lemma 2.6. *Let M be a $(\varepsilon, \mathcal{R})$-differentially private mechanism, and let h be any function. Define $M'(D, f) \stackrel{\mathsf{def}}{=} h(M(D, f))$. Then M' is $(\varepsilon, \mathcal{R})$-differentially private.*

Definition 2.7 ($(\rho, \mathcal{R})$-Utility)**.** *Let $\rho > 0$, let $\mathcal{R}$ be a source, and let $\mathcal{F} = \{f : \mathcal{D} \to \mathbb{Z}\}$ be a class of functions. We say a mechanism M has* $(\rho, \mathcal{R})$-utility *for $\mathcal{F}$ if for all databases $D \in \mathcal{D}$, all queries $f \in \mathcal{F}$, and all distributions $\mathsf{R} \in \mathcal{R}$,*

$$\mathbb{E}_{\mathbf{r} \leftarrow \mathsf{R}}[|f(D) - M(D, f\ ; \mathbf{r})|] \leq \rho.$$

At the extremes, a mechanism that always outputs 0 is $(0, \mathcal{R})$-differentially private, while a mechanism that outputs the true answer $f(D)$ has $(0, \mathcal{R})$-utility. Neither of these mechanisms is very interesting—the first gives no utility, while the second provides no privacy. Instead, we wish to find mechanisms that achieve a good trade-off between privacy and utility. This motivates the following definition.

Definition 2.8 (Non-Triviality)**.** *We say a function family $\mathcal{F}$* admits non-trivial differentially private mechanisms w.r.t. $\mathcal{R}$ *if there exists a function $g(\cdot)$ such that for all $\varepsilon > 0$ there exists a mechanism M_ε that is $(\varepsilon, \mathcal{R})$-differentially private and has $(g(\varepsilon), \mathcal{R})$-utility. We call $\mathcal{M} = \{M_\varepsilon\}$ a class of non-trivial mechanism for $\mathcal{F}$ w.r.t. $\mathcal{R}$.*

We make a few remarks regarding this definition. First, we require that the utility $\rho = g(\varepsilon)$ is independent of $|D|$. Second, we note that non-triviality implies that we can achieve $(\varepsilon, \mathcal{R})$-differential privacy for *any* $\varepsilon > 0$ (possibly at the expense of utility). E.g., when $\mathcal{R} = \mathcal{SV}(\gamma)$, we should be able to achieve $\varepsilon \ll \gamma$, which is below the "extraction barrier" for SV-sources. Finally, we note that for the purpose of satisfying this definition, we can assume w.l.o.g. that $\varepsilon \leq 1$, which is anyway the case of most interest. Moreover, we can assume that $1/\varepsilon$ is an integer, since otherwise we can simply take a slightly smaller ε for which this is the case.

Infinite-Precision Mechanisms. As we will see shortly, it is sometimes easier to describe mechanisms using samples from some *continuous* random variable X, instead of using a (discrete) random tape in $\{0,1\}^*$. Moreover, the notions of privacy, utility, and non-triviality definitions can be analogously defined for this case as well (which we omit for brevity). Of course, to actually "implement" such abstract mechanisms in practice, one must specify how to approximate them using a "finite precision" random tape in $\{0,1\}^*$, without significantly affecting their privacy and/or utility. When perfect randomness $\mathcal{U}$ is available, this is typically quite easy (and usually not spelled out in most differential privacy papers), by simply approximating a continuous sample from X within some "good enough" finite precision. In contrast, our mechanisms will have to deal with imperfect randomness $\mathcal{SV}(\gamma)$, so rounding a given "continuous" mechanism into a "discrete" mechanism will be non-trivial and require utmost care. In particular, we will have to design quite special "infinite-precision" mechanisms which will be "SV-friendly" toward appropriate "finite-precision rounding".

Additive Noise Mechanisms. One type of non-trivial mechanisms follow the following blueprint: first, they sample *data-independent* noise x from some (discrete or continuous) distribution X, calculate the true answer $f(D)$, and output $z = f(D) + x$. We call such mechanisms, *additive-noise mechanisms* (examples of additive-noises mechanisms include the Laplacian mechanism [DMNS06], the geometric mechanism [GRS09], and the K-norm mechanism for multiple linear queries [HT10]). If $\mathbb{E}[|\mathsf{X}|]$ is bounded, then the mechanism has bounded utility. However, to argue that such bounded "noise" X is sufficient to ensure the differential privacy of such mechanisms, we must first restrict our query class $\mathcal{F}$. In particular, it turns out that additive-noise mechanisms achieve differential privacy for a pretty large class of useful functions, called *bounded sensitivity* functions.

Definition 2.9 (Sensitivity)**.** *For $f : \mathcal{D} \to \mathbb{Z}$, the* sensitivity *of f is defined as*

$$\Delta f \stackrel{\mathsf{def}}{=} \max_{D_1, D_2} \|f(D_1) - f(D_2)\|$$

for all neighboring databases $D_1, D_2 \in \mathcal{D}$. For $d \in \mathbb{Z}^+$, we define $\mathcal{F}_d = \{f : \mathcal{D} \to \mathbb{Z} \mid \Delta f \leq d\}$ to be the class of functions with sensitivity at most d.

Intuitively, low sensitivity functions do not change too much on neighboring databases, which suggests that relatively small noise can "mask" the difference between $f(D_1)$ and $f(D_2)$. The particular (continuous) distribution turns out to be the Laplacian distribution, defined below.

Definition 2.10 (Laplacian Distribution)**.** *The* Laplacian distribution *with mean μ and standard deviation $\sqrt{2}b$, denoted $\mathsf{Lap}_{\mu,b}$, has probability density function $\mathsf{Lap}_{\mu,b}(x) = (1/2b) \cdot e^{-|x-\mu|/b}$. The cumulative distribution function is $\mathsf{CDF}^{\mathsf{Lap}}_{\mu,b}(x) = (1/2b) \cdot (1 + \mathrm{sgn}(x) \cdot (1 - e^{|x-\mu|/b}))$.*

We also define the distribution obtained from sampling the Laplacian distribution $\mathsf{Lap}_{\mu,b}$ and rounding to the nearest integer $\lfloor \mathsf{Lap}_{\mu,b} \rceil$. We call this the "rounded" Laplacian distribution and denote it by $\mathsf{RLap}_{\mu,b}$.

In particular, for any sensitivity bound d, Dwork et al. [DMNS06] showed the following class of (infinite-precision) additive-noise mechanisms $\mathcal{M}^{\mathsf{Lap}} = \{M^{\mathsf{Lap}}_{\varepsilon}\}$ is non-trivial for $\mathcal{F}_d$. Given a

database $D \in \mathcal{D}$, a query $f \in \mathcal{F}_d$ and the target value of ε, the mechanism $M_\varepsilon^{\mathsf{Lap}}$ computes $f(D)$ and adds noise from the Laplacian distribution with mean 0 and standard deviation $(\sqrt{2} \cdot d)/\varepsilon$; i.e. $M_\varepsilon^{\mathsf{Lap}}(D, f) \stackrel{\mathsf{def}}{=} f(D) + \mathsf{Lap}_{0,d/\varepsilon}$. Equivalently, we can also view this mechanism as computing $y = f(D)$ and outputting a sample from the distribution $\mathsf{Lap}_{y,d/\varepsilon}$. Moreover, it is easy to see that this infinite-precision mechanism achieves utility $O(d/\varepsilon)$.

In order to ensure that the output of the mechanism of [DMNS06] is an integer, the result can be rounded to the nearest integer. Since this is post-processing, by Lemma 2.6, the result has the same privacy guarantees. Furthermore, since $f(D) \in \mathbb{Z}$, we have $\lfloor f(D) + \mathsf{Lap}_{0,d/\varepsilon} \rceil = y + \lfloor \mathsf{Lap}_{0,d/\varepsilon} \rceil$. In particular, for queries of integer range, the mechanism of [DMNS06] can be seen as computing $y = f(D)$ and outputting $z = \mathsf{RLap}_{y,d/\varepsilon}$. We denote this (still infinite-precision, but now integer range) variant by $M_\varepsilon^{\mathsf{RLap}}$. Clearly, it still has utility $O(d/\varepsilon)$.

Finally, we must describe how to approximate this mechanism family $\mathcal{M}^{\mathsf{RLap}}$ by a finite precision family $\overline{\mathcal{M}}^{\mathsf{RLap}}$ w.r.t. $\mathcal{U}$, without significantly affecting privacy or utility. As it turns out, a good enough approximation can be accomplished by sampling each value $z \in \mathbb{Z}$ with precision roughly proportional to $\Pr[z]$ (under $M_\varepsilon^{\mathsf{RLap}}$), which requires $n(z) = O(|z| \log(d/\varepsilon))$ (truly random) coins $\mathsf{U}_{n(z)}$, and increases both ε and ρ by at most a constant factor. Since we will not use the resulting (finite-precision) mechanism in this paper [2], we state the end result without further justification.

Lemma 2.11 ([DMNS06])**.** *For any $d \in \mathbb{Z}^+$, there exists a family $\overline{\mathcal{M}}^{\mathsf{RLap}} = \{\overline{M}_\varepsilon^{\mathsf{RLap}}\}$ of non-trivial mechanisms for $\mathcal{F}_d$ w.r.t. the uniform source $\mathcal{U}$, with utility function $g^{\mathsf{RLap}}(\varepsilon) = O(d/\varepsilon)$.*

Our Question. Lemma 2.11 shows that for all $d \in \mathbb{Z}^+$ there exists a class of non-trivial mechanisms for $\mathcal{F}_d$ w.r.t. $\mathcal{U}$. The main goal of this work is to determine if this is also true for other randomness sources, in particular, for the γ-Santha-Vazirani sources.

Main Question (Restated)**:** Does there exist a class $\mathcal{M} = \{M_\varepsilon\}$ of non-trivial mechanisms for $\mathcal{F}_d$ w.r.t. $\mathcal{SV}(\gamma)$ for all $\gamma \in [0, 1)$? If so, can they be additive-noise mechanisms?

For clarity, from now we will focus on the case $d = 1$; however, all our results extend to any sensitivity bound d. We will prove that non-trivial mechanisms for $\mathcal{F}_1$ w.r.t. $\mathcal{SV}(\gamma)$ cannot be additive noise, answering the second question in the negative. Despite this, however, we will answer the first question positively by displaying a class $\mathcal{M} = \{M_\varepsilon\}$ of non-trivial (non-additive-noise) mechanisms for $\mathcal{F}_1$ w.r.t. $\mathcal{SV}(\gamma)$.

3 Naive Approaches and a Lower Bound

We will start by showing a few naive approaches that will explain the intuition behind why non-trivial mechanisms for $\mathcal{F}_1$ w.r.t. $\mathcal{SV}(\gamma)$ cannot be additive noise. Moreover, we will prove a general lower bound restricting the type of mechanisms "friendly" to SV-sources, which will motivate a very special type of mechanisms that we will introduce in Section 4.

First Attempt. A first approach to answer our main question would be to prove that *any* class of non-trivial mechanisms for $\mathcal{F}_1$ w.r.t. $\mathcal{U}$ is also non-trivial w.r.t. $\mathcal{SV}(\gamma)$. This turns out to be far too optimistic. To see this, take any mechanism M w.r.t. $\mathcal{U}$, and assume that with high probability M needs at most n random coins, where n is odd. Define (artificial) mechanism M' as follows. Whenever M needs a fresh coin b, M' samples n coins $b_1 \dots b_n$ and simply sets $b = \textsc{maj}_n(b_1, \dots, b_n)$, where $\textsc{maj}_n(\cdot)$ is the majority of n bits. Clearly, M' has the same differential privacy and utility guarantees as M w.r.t. $\mathcal{U}$, since majority of perfectly random bits is perfectly random. On the other hand, by biasing each bit towards 0 (resp. 1), a Santha-Vazirani adversary can fix every n-bit majority function to 0 (resp. 1) with probability at least $(1 - e^{-\gamma^2 n/2})$, which means that he can fix all n coins of M to any

[2] Indeed, we will see in Lemma 3.1 that no additive-noise mechanism can be non-trivial w.r.t. $\mathcal{SV}(\gamma)$.

desired outcome with probability at least $(1-ne^{-\gamma^2 n/2}) \approx 1$. Hence, the Santha-Vazirani adversary for M' can effectively fix the random tape of M, making it deterministic (with probability exponentially close to 1). On the other hand, it is easy to see that no deterministic mechanism having non-trivial utility (i.e., giving distinct answers on some two neighboring databases) can be differentially private.

Hence, non-trivial mechanisms w.r.t. the uniform source $\mathcal{U}$ are not necessarily non-trivial w.r.t. γ-Santha-Vazirani sources $\mathcal{SV}(\gamma)$.

Second Attempt. A seemingly less naive idea would be to prove that *any* class of non-trivial mechanisms for $\mathcal{F}_1$ w.r.t. $\mathcal{U}$ is also non-trivial w.r.t. $\mathcal{SV}(\gamma)$ if we first run some extractor Ext on the randomness. More precisely, suppose $\mathcal{M} = \{M_\varepsilon\}$ is non-trivial w.r.t. $\mathcal{U}$ and suppose M_ε uses n coins. Can we construct a deterministic extractor $\mathsf{Ext} : \{0,1\}^m \to \{0,1\}^n$ (for some sufficiently large $m \gg n$) and let $M'_\varepsilon \stackrel{\mathsf{def}}{=} M_\varepsilon(D, f\,; \mathsf{Ext}(\mathbf{r}))$, such that $\mathcal{M}' = \{M'_\varepsilon\}$ *is non-trivial w.r.t.* $\mathcal{SV}(\gamma)$ *whenever* $\mathcal{M} = \{M_\varepsilon\}$ *is non-trivial w.r.t.* $\mathcal{U}$? More generally, one can define an analogous "extractor conjecture" for any imperfect source $\mathcal{R}$ in place of $\mathcal{SV}(\gamma)$. Unfortunately, we show that this naive approach does not work for any "non-extractable" source $\mathcal{R}$, such as $\mathcal{SV}(\gamma)$. To show this, we look at the family of *additive-noise* mechanisms for the family $\mathcal{F}_1$ of sensitivity-1 functions given by Lemma 2.11, and observe that applying an extractor to any additive-noise mechanism is *still* an additive-noise mechanism. Then, we show a more general statement that *any* non-trivial additive-noise mechanism for $\mathcal{F}_1$ under $\mathcal{R}$ implies the existence of a bit extractor for $\mathcal{R}$, which is impossible for non-extractable $\mathcal{R}$, such as $\mathcal{SV}(\gamma)$.

Lemma 3.1. *Assume $\mathcal{R}$ is a source and $\mathcal{M} = \{M_\varepsilon\}$ is a family of* additive-noise *mechanisms for $\mathcal{F}_1$, where each M_ε is $(\varepsilon, \mathcal{R})$-differentially private. Then, for* all *$\varepsilon > 0$, one can deterministically extract an ε-biased bit from $\mathcal{R}$. In particular, (a) there does* not *exist a class $\mathcal{M} = \{M_\varepsilon\}$ of non-trivial* additive-noise *mechanisms for $\mathcal{F}_1$ w.r.t. $\mathcal{SV}(\gamma)$; and, by Lemma 2.11, (b) the "extractor-conjecture" is false for any "non-extractable" $\mathcal{R}$, such as $\mathcal{SV}(\gamma)$.*

Proof. Given any class $\mathcal{M} = \{M_\varepsilon\}$ of additive-noise mechanisms for $\mathcal{F}_1$, define a new class $\mathcal{M}' = \{M'_\varepsilon\}$ of binary-output mechanisms, where $M'_\varepsilon(D, f; \mathbf{r}) = M_\varepsilon(D, f; \mathbf{r}) \bmod 2$. Notice, if M_ε is $(\varepsilon, \mathcal{R})$-differentially private, then so is M'_ε by Lemma 2.6 (since mod 2 is a deterministic post-processing function). Also, since M_ε is additive-noise, we can write $M_\varepsilon(D, f; \mathbf{r}) = f(D) + x$, where $x = \mathsf{Ext}_\varepsilon(\mathbf{r})$ for some function $\mathsf{Ext}_\varepsilon : \{0,1\}^* \to \mathbb{Z}$. Thus, if we let $\mathsf{Ext}'_\varepsilon(\mathbf{r}) = \mathsf{Ext}_\varepsilon(\mathbf{r}) \bmod 2$, we can write $M'_\varepsilon(D, f; \mathbf{r}) = (D(f) \bmod 2) \oplus \mathsf{Ext}'_\varepsilon(\mathbf{r})$, where $\oplus$ is the exclusive-or operator on one bit.

Now fix any two neighboring databases D_0 and D_1 and any $f \in \mathcal{F}_1$ s.t. $|f(D_0) - f(D_1)| = 1$. In fact, by swapping D_0 and D_1, if necessary, we can assume that $f(D_b) \bmod 2 = b$, for $b \in \{0,1\}$. This means that $M'_\varepsilon(D_b, f; \mathbf{r}) = (D_b(f) \bmod 2) \oplus \mathsf{Ext}'_\varepsilon(\mathbf{r}) = b \oplus \mathsf{Ext}'_\varepsilon(\mathbf{r})$. By $(\varepsilon, \mathcal{R})$-differential privacy of M'_ε applied to D_0 and D_1, this means that for any $\mathsf{R} \in \mathcal{R}$, we must have $\Pr[\mathsf{Ext}'_\varepsilon(\mathsf{R}) = 0] \in [\frac{1}{2}(1-\varepsilon), \frac{1}{2}(1+\varepsilon)]$, which would mean that $\mathsf{Ext}'_\varepsilon$ defines an ε-biased one-bit extractor for $\mathcal{R}$. □

General Lower Bound. The failure of our naive approaches suggests that one cannot take any non-trivial mechanism w.r.t. uniform randomness $\mathcal{U}$, and apply some simple transformation to its randomness to derive a non-trivial mechanism w.r.t. $\mathcal{SV}(\gamma)$. Indeed, we will show that *any* non-trivial mechanism w.r.t. $\mathcal{SV}(\gamma)$ must in fact satisfy a pretty restrictive condition w.r.t. to the uniform source. In particular, this condition (later called *consistent-sampling*) is never satisfied by additive-noise mechanisms.

First, we need some important notation. Consider a mechanism M with randomness space $\{0,1\}^*$, and let $D \in \mathcal{D}$. We define the set $T(D, f, z) \stackrel{\mathsf{def}}{=} \{\mathbf{r} \in \{0,1\}^n \mid z = M(D, f\,; \mathbf{r})\}$ to be the set of random coins $\mathbf{r} \in \{0,1\}^*$ such that M outputs z when run on database D, query f, and random coins $\mathbf{r}$. We remark that since we assume that only $n = n(f, z, D)$ coins need to be sampled to determine if $M(D, f) = z$, we can assume w.l.o.g. that $T(f, z, D) \subseteq \{0,1\}^n$. In the interest of clarity, we simply write T when f, D, and z are understood from context.

Without loss of generality, we assume that the function family $\mathcal{F}$ is by itself non-trivial, meaning that there exist two neighboring databases D_1, D_2 and a query f such that $f(D_1) \neq f(D_2)$. We also let $T_1 \stackrel{\mathsf{def}}{=} T(D_1, f, z)$ and $T_2 \stackrel{\mathsf{def}}{=} T(D_2, f, z)$. Fix $z \in \mathbb{Z}, f \in \mathcal{F}, \mathsf{R} \in \mathcal{R}$. To show that M is $(\varepsilon, \mathcal{R})$-differentially private for $\mathcal{F}$ w.r.t. randomness source $\mathcal{R}$, we are concerned with bounding the following ratio by $1 + \varepsilon$:

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{R}}[M(D_1, f\ ; \mathbf{r}) = z]}{\Pr_{\mathbf{r} \leftarrow \mathsf{R}}[M(D_2, f\ ; \mathbf{r}) = z]} = \frac{\Pr_{\mathbf{r} \leftarrow \mathsf{R}}[\mathbf{r} \in T_1]}{\Pr_{\mathbf{r} \leftarrow \mathsf{R}}[\mathbf{r} \in T_2]}$$

As we show below, bounding the above ratio for all Santha-Vazirani sources introduces a non-trivial constraint of M. For illustration, let us first look at any additive-noise mechanism M and re-derive the conclusion of Lemma 3.1 directly. If $z = M(D_1, f\ ; \mathbf{r}_1) = M(D_2, f\ ; \mathbf{r}_2)$ then $z = f(D_1) + x_1 = f(D_2) + x_2$ for $x_1, x_2 \leftarrow \mathsf{X}$. Since we assumed $f(D_1) \neq f(D_2)$ then $x_1 \neq x_2$, which means that $\mathbf{r}_1 \neq \mathbf{r}_2$. Thus, $T_1 \cap T_2 = \emptyset$. Furthermore, we can assume w.l.o.g. that $|T_1| \geq |T_2|$ since otherwise we can switch D_1 and D_2. Using Lemma 2.4 with $G = T_1$ and $B = T_2$, and the fact that $\mathcal{H}(\gamma, n) \subset \mathcal{SV}(\gamma, n)$, we have that there exists $\mathsf{SV}(\gamma) \in \mathcal{SV}(\gamma)$ such that

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)}[M(D_1, f\ ; \mathbf{r}) = z]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)}[M(D_2, f\ ; \mathbf{r}) = z]} \geq (1 + \gamma) \cdot \frac{|T_1|}{|T_2|} \geq 1 + \gamma,$$

which is the same conclusion as the one obtained in the proof of Lemma 3.1.

More generally, coming back to arbitrary mechanisms, since Lemma 2.4 works even when $G \cap B \neq \emptyset$, we get the following much stronger result. Suppose $\sigma \stackrel{\mathsf{def}}{=} \frac{|T_2 \setminus T_1|}{|T_2|} \in [0, 1]$. Then there exists $\mathsf{SV}(\gamma) \in \mathcal{SV}(\gamma)$ such that

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)}[M(D_1, f\ ; \mathbf{r}) = z]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)}[M(D_2, f\ ; \mathbf{r}) = z]} \geq 1 + \gamma\sigma.$$

This shows that a *necessary* condition to achieve $(\varepsilon, \mathcal{SV}(\gamma))$-differential privacy is that $\sigma \leq \varepsilon/\gamma = O(\varepsilon)$. We summarize this in the following lemma.

Lemma 3.2. *Assume $\gamma > 0$ and M is $(\varepsilon, \mathcal{SV}(\gamma))$-differentially private mechanism for some class $\mathcal{F}$. Fix any $z \in \mathbb{Z}, f \in \mathcal{F}$, and any neighboring databases $D_1, D_2 \in \mathcal{D}$ s.t. $f(D_1) \neq f(D_2)$. Let $T_1 \stackrel{\mathsf{def}}{=} T(D_1, f, z)$, $T_2 \stackrel{\mathsf{def}}{=} T(D_2, f, z)$, and assume that $|T_1| \geq |T_2|$. Then $\sigma \stackrel{\mathsf{def}}{=} \frac{|T_2 \setminus T_1|}{|T_2|} \leq \frac{\varepsilon}{\gamma} = O(\varepsilon)$.*

4 SV-Consistent Sampling

Recall that we defined $T(D, f, z) \stackrel{\mathsf{def}}{=} \{\mathbf{r} \in \{0,1\}^n \mid z = M(D, f\ ; \mathbf{r})\}$ to be the set of all coins $\mathbf{r}$ such that M outputs z when run on database D, query f and randomness $\mathbf{r}$. Further recall that for neighboring databases D_1, D_2, we let $T_1 \stackrel{\mathsf{def}}{=} T(D_1, f, z)$ and $T_2 \stackrel{\mathsf{def}}{=} T(D_2, f, z)$.

By Lemma 3.2 we know that in order to achieve $(\varepsilon, \mathcal{SV}(\gamma))$-differential privacy we must have $\frac{|T_2 \setminus T_1|}{|T_2|} = O(\varepsilon)$. This means that for arbitrary $\varepsilon > 0$, it must be that $\frac{|T_2 \setminus T_1|}{|T_2|} \to 0$ as $\varepsilon \to 0$. This motivates our definition of *$\widetilde{\varepsilon}$-consistent sampling*. Later we will define ε in terms of $\widetilde{\varepsilon}$ such that $\varepsilon \to 0$ as $\widetilde{\varepsilon} \to 0$. We remark that our definition of $\widetilde{\varepsilon}$-consistent sampling is similar to the definition of [Man94, Hol07], which has already been used in the context of differential privacy [MMP+10].

Definition 4.1. *We say M has* $\widetilde{\varepsilon}$-consistent sampling ($\widetilde{\varepsilon}$-CS) *if for all $z \in \mathbb{Z}, f \in \mathcal{F}$ and neighboring databases $D_1, D_2 \in \mathcal{D}$ such that $T_2 \neq \emptyset$, we have*

$$\frac{|T_1 \setminus T_2|}{|T_2|} \leq \widetilde{\varepsilon}.$$

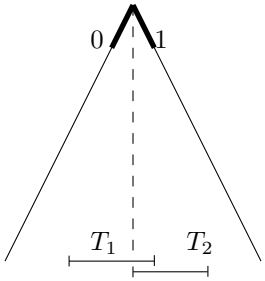

Figure 4.1: Example of how a $\mathsf{SV}(\gamma) \in \mathcal{SV}(\gamma)$ distribution can increase the ratio $\frac{\Pr_{\mathbf{r}\leftarrow\mathsf{SV}(\gamma)}[\mathbf{r}\in T_1]}{\Pr_{\mathbf{r}\leftarrow\mathsf{SV}(\gamma)}[\mathbf{r}\in T_2]}$.

We make a few remarks about Definition 4.1. First, notice that w.l.o.g. we can assume that $|T_1| \geq |T_2|$ since in this case we have $\frac{|T_2\backslash T_1|}{|T_1|} \leq \frac{|T_1\backslash T_2|}{|T_2|}$. Second, notice that $\widetilde{\varepsilon}$-consistent sampling also guarantees that $\frac{|T_2\backslash T_1|}{|T_2|} \leq \frac{|T_1\backslash T_2|}{|T_2|} \leq \widetilde{\varepsilon}$, which Lemma 3.2 tells us is a necessary condition for non-trivial differential privacy. Finally, it is easy to see that if a mechanism has $\widetilde{\varepsilon}$-consistent sampling, then it is $(\widetilde{\varepsilon}, \mathcal{U})$-differentially private, as

$$\frac{\Pr_{\mathbf{r}\leftarrow\mathsf{U}_n}[\mathbf{r} \in T_1]}{\Pr_{\mathbf{r}\leftarrow\mathsf{U}_n}[\mathbf{r} \in T_2]} = \frac{|T_1|}{|T_2|} = \frac{|T_1 \cap T_2|}{|T_2|} + \frac{|T_1\backslash T_2|}{|T_2|} \leq 1 + \widetilde{\varepsilon}.$$

To summarize, $\widetilde{\varepsilon}$-consistent sampling is *sufficient* to achieve $(\widetilde{\varepsilon}, \mathcal{U})$-differential privacy and is essentially *necessary* to achieve $(\gamma\widetilde{\varepsilon}, \mathcal{SV}(\gamma))$-differential privacy. But is it sufficient to achieve $(p(\widetilde{\varepsilon}), \mathcal{SV}(\gamma))$-differential privacy for some function p such that $p(\widetilde{\varepsilon}) \to 0$ as $\widetilde{\varepsilon} \to 0$? This turns out not to be the case, as it is still possible for a Santha-Vazirani distribution to increase the probability of $T_1\backslash T_2$ while simultaneously decreasing the probability of T_2. For instance, consider the example in Figure 4.1, where pictorially, we view each coin $\mathbf{r} \in \{0,1\}^*$ as defining a path down a binary tree. In this example, $T_1\backslash T_2$ and T_2 are positioned precisely to the left and right of $1/2$, respectively. After the first coin, the SV-distribution can focus on either targeting $T_1\backslash T_2$ or avoiding T_2. If the height of this tree is big, then the SV distribution can greatly increase our ratio. This suggests that in order to handle γ-Santha Vazirani distributions, we need to make more restrictions on the mechanism.

New Observations. We make two observations that will help us guarantee that the example described in Figure 4.1 does not arise, but we first define some notation. For $m \in \mathbb{Z}^+$ and a bit sequence $\mathbf{x} = x_1, \ldots, x_m \in \{0,1\}^m$, we define $\textsc{suffix}(\mathbf{x}) \stackrel{\mathsf{def}}{=} \{\mathbf{y} = y_1, y_2, \ldots \in \{0,1\}^* \mid x_i = y_i \text{ for all } i \in [m]\}$ to be the set of all bit strings that have $\mathbf{x}$ as a prefix. For $n \in \mathbb{Z}^+$ such that $m \leq n$, we define $\textsc{suffix}(\mathbf{x}, n) \stackrel{\mathsf{def}}{=} \textsc{suffix}(\mathbf{x}) \cap \{0,1\}^n$.

Our first observation is that if we consider the longest prefix $\mathbf{u}$ of all elements in $T_1 \cup T_2$, then the ratio is the same as when the probabilities are conditioned on $\mathbf{r}$ having this prefix. This is because in order for $\mathbf{r} \in T_1\backslash T_2$ or $\mathbf{r} \in T_2$, it must be the case that $\mathbf{r} \in \textsc{suffix}(\mathbf{u}, n)$.

Our second observation is that we want to ensure that $\textsc{suffix}(\mathbf{u}, n)$ is a good approximation of $T_1 \cup T_2$, that is, that $|\textsc{suffix}(\mathbf{u}, n)| \approx |T_1 \cup T_2|$. This guarantees that we never encounter the problem that arose in the example in Figure 4.1. For this to be the case, however, we must first ensure that $T_1 \cup T_2$ are "close together". We therefore make the following definition.

Definition 4.2. *We say $\mathcal{M}$ is an* interval mechanism *if for all queries $f \in \mathcal{F}$, databases $D \in \mathcal{D}$, and possible outcomes $z \in \mathbb{Z}$, the values in T constitute an interval, that is, $T \neq \emptyset$ and the set $\{\textsc{int}(\mathbf{r}) \mid \mathbf{r} \in T\}$ contains consecutive integers, where for $\mathbf{r} = r_1 \ldots r_n \in \{0,1\}^n$, we define $\textsc{int}(\mathbf{r}) \stackrel{\mathsf{def}}{=} \sum_{i=1}^n r_i \cdot 2^{n-i}$.*

We now formalize the requirement we described above. Let D_1, D_2 be two neighboring databases, let $f \in \mathcal{F}$, let z be a possible outcome, and let $n \stackrel{\mathsf{def}}{=} \max(n(D_1, f, z), n(D_2, f, z))$. We let $\mathbf{u}$ be the

longest prefix such that $T_1 \cup T_2 \subseteq \textsc{suffix}(\mathbf{u}, n)$. Formally,

$$\mathbf{u} \stackrel{\mathsf{def}}{=} \text{argmax}\{|\mathbf{u}'| \mid \mathbf{u}' \in \{0,1\}^{\leq n} \text{ and } T_1 \cup T_2 \subseteq \textsc{suffix}(\mathbf{u}', n)\}$$

Definition 4.3. *Let $\widetilde{\varepsilon} > 0, c > 1$. We say that an interval mechanism M has $(\widetilde{\varepsilon}, c)$-*SV-consistent sampling ($(\widetilde{\varepsilon}, c)$-SVCS) *if it has $\widetilde{\varepsilon}$-consistent sampling and for all queries $f \in \mathcal{F}$, all neighboring databases $D_1, D_2 \in \mathcal{D}$ and all possible outcomes $z \in \mathbb{Z}$, which define $\mathbf{u}$ as above, we have:*

$$\frac{|\textsc{suffix}(\mathbf{u}, n)|}{|T_1 \cup T_2|} \leq c$$

We now show that $(\widetilde{\varepsilon}, c)$-SV-consistent sampling is sufficient to obtain $(\varepsilon, \mathcal{SV}(\gamma))$-differential privacy for an interesting value of ε, that is, for an ε that can be made arbitrarily small by decreasing $\widetilde{\varepsilon}$.

Theorem 4.4. *If M has $(\widetilde{\varepsilon}, c)$-SV-consistent sampling, then M is $(\varepsilon, \mathcal{SV}(\gamma))$-differentially private, where*

$$\varepsilon = (8\widetilde{\varepsilon})^{1-\log(1+\gamma)} \left(\frac{1+\gamma}{1-\gamma}\right)^{\log(8c)}$$

In particular, for $\gamma \in [0, 1)$ and $c = O(1)$, we have $\varepsilon \to 0$ as $\widetilde{\varepsilon} \to 0$.

Before proving Theorem 4.4, we make two additional definitions and prove a lemma. Let D_1, D_2 be two neighboring databases, $f \in \mathcal{F}$, z be a possible outcome, and $n = \max(n(D_1, f, z), n(D_2, f, z))$.

- Define $\mathbf{v}$ to be the longest prefix such that $T_1 \backslash T_2 \subseteq \textsc{suffix}(\mathbf{v}, n)$. Formally,

$$\mathbf{v} = \text{argmax}\{|\mathbf{v}'| \mid \mathbf{v} \in \{0,1\}^{\leq n} \text{ and } T_1 \backslash T_2 \subseteq \textsc{suffix}(\mathbf{v}', n)\}$$

- Define $\mathbf{w}$ to be the shortest prefix such that $\textsc{suffix}(\mathbf{w}, n) \subseteq T_2$. Formally,

$$\mathbf{w} = \text{argmin}\{|\mathbf{w}'| \mid \mathbf{w}' \in \{0,1\}^{\leq n} \text{ and } \textsc{suffix}(\mathbf{w}', n) \subseteq T_2\}$$

  We remark that $\mathbf{w}$ may not be unique. In this case, any of the possible values is just as good since we will be concerned with the value $|\mathbf{w}|$ which is the same across all possible values of $\mathbf{w}$.

See Figure 4.2 for a pictorial representation of $\mathbf{u}, \mathbf{v}, \mathbf{w}$. Note the asymmetry of the definitions of $\mathbf{u}, \mathbf{v}$, and $\mathbf{w}$. Also note that we define $\mathbf{v}$ and $\mathbf{w}$ in such a way that $\textsc{suffix}(\mathbf{v}) \cap \textsc{suffix}(\mathbf{w}) = \emptyset$. Informally, $|\mathbf{v}| - |\mathbf{w}|$ is roughly the number of coins that the Santha-Vazirani distribution needs to use to increase the probability of landing in $T_1 \backslash T_2$ without affecting the probability of landing in T_2, while $|\mathbf{w}| - |\mathbf{u}|$ is roughly the number of coins that it can use to decrease the probability of landing in T_2 without affecting the probability of landing in $T_1 \backslash T_2$. We first prove a lemma that says that if M has $(\widetilde{\varepsilon}, c)$-SV-consistent sampling then $|\mathbf{v}| - |\mathbf{w}| = \Omega(\log(1/\widetilde{\varepsilon}))$ and $|\mathbf{w}| - |\mathbf{u}| = O(1)$.

Lemma 4.5. *If M has $(\widetilde{\varepsilon}, c)$-SV-consistent sampling then for all neighboring databases $D_1, D_2 \in \mathcal{D}$ which define $\mathbf{u}, \mathbf{v}, \mathbf{w}$ as above, we have:*

$$|\mathbf{v}| - |\mathbf{w}| \geq \log\left(\frac{1}{8\widetilde{\varepsilon}}\right) \quad \textit{and} \quad |\mathbf{w}| - |\mathbf{u}| \leq \log(8c)$$

Proof. By $(\widetilde{\varepsilon}, c)$-SV-consistent sampling we know that

$$\frac{|T_1 \backslash T_2|}{|T_2|} \leq \widetilde{\varepsilon} \quad \text{and} \quad \frac{|\textsc{suffix}(\mathbf{u}, n)|}{|T_1 \cup T_2|} \leq c$$

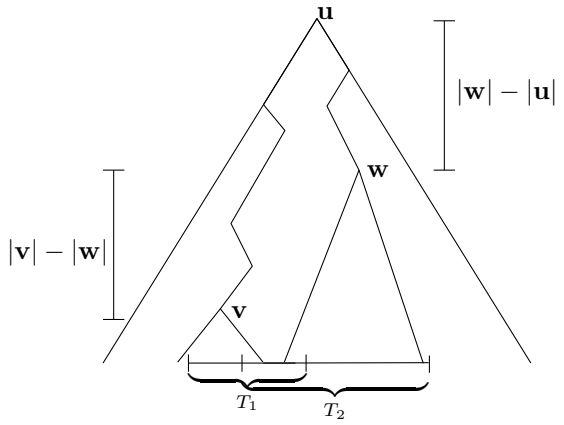

Figure 4.2: Definitions of $\mathbf{u}, \mathbf{v}, \mathbf{w}$.

We therefore have,

$$|\textsc{suffix}(\mathbf{v}, n)|/2 \leq |T_1 \backslash T_2| \leq \widetilde{\varepsilon} \cdot |T_2| \leq 4\widetilde{\varepsilon} \cdot |\textsc{suffix}(\mathbf{w}, n)|$$

$$n - |\mathbf{v}| \leq \log(8\widetilde{\varepsilon}) + n - |\mathbf{w}|$$

Reorganizing yields the first inequality. We also have,

$$|\textsc{suffix}(\mathbf{u}, n)| \leq c \cdot |T_1 \cup T_2| \leq c \cdot (|T_1 \backslash T_2| + |T_2|) \leq (1 + \widetilde{\varepsilon}) \cdot |T_2| \leq 2c \cdot |T_2| \leq 8c \cdot |\textsc{suffix}(\mathbf{w}, n)|$$

$$n - |\mathbf{u}| \leq \log(8c) + n - |\mathbf{w}|$$

Reorganizing yields the second inequality. □

We now prove Theorem 4.4.

Proof of Theorem 4.4: Fix $z \in \mathbb{Z}, f \in \mathcal{F}$, and neighboring databases $D_1, D_2 \in \mathcal{D}$. Let $n = \max(n(D_1, f, z), n(D_2, f, z))$. Also fix a γ-Santha-Vazirani distribution $\mathsf{SV}(\gamma) \in \mathcal{SV}(\gamma)$. Then,

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_1]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_2]} = 1 + \frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_1 \backslash T_2]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_2]}$$

So we need only prove that

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_1 \backslash T_2]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_2]} \leq \varepsilon$$

By the total probability theorem,

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_1 \backslash T_2]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_2]} = \frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_1 \backslash T_2 \mid \mathbf{r} \in \textsc{suffix}(\mathbf{u})]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_2 \mid \mathbf{r} \in \textsc{suffix}(\mathbf{u})]}$$

By our definition of $\mathbf{v}$ and $\mathbf{w}$, we have

$$\frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_1 \backslash T_2 \mid \mathbf{r} \in \textsc{suffix}(\mathbf{u})]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in T_2 \mid \mathbf{r} \in \textsc{suffix}(\mathbf{u})]} \leq \frac{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in \textsc{suffix}(\mathbf{v}) \mid \mathbf{r} \in \textsc{suffix}(\mathbf{u})]}{\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma,n)}[\mathbf{r} \in \textsc{suffix}(\mathbf{w}) \mid \mathbf{r} \in \textsc{suffix}(\mathbf{u})]}$$

By the definition of a Santha-Vazirani source and the fact that we are conditioning on $\mathbf{r} \in \text{SUFFIX}(\mathbf{u})$,

$$\begin{aligned}\frac{\Pr_{\mathbf{r}\leftarrow\mathsf{SV}(\gamma,n)}[\mathbf{r} \in \text{SUFFIX}(\mathbf{v}) \mid \mathbf{r} \in \text{SUFFIX}(\mathbf{u})]}{\Pr_{\mathbf{r}\leftarrow\mathsf{SV}(\gamma,n)}[\mathbf{r} \in \text{SUFFIX}(\mathbf{w}) \mid \mathbf{r} \in \text{SUFFIX}(\mathbf{u})]} &\leq \frac{\left(\frac{1}{2}(1+\gamma)\right)^{|\mathbf{v}|-|\mathbf{u}|}}{\left(\frac{1}{2}(1-\gamma)\right)^{|\mathbf{w}|-|\mathbf{u}|}} \\ &= \left(\frac{1}{2}(1+\gamma)\right)^{|\mathbf{v}|-|\mathbf{w}|}\left(\frac{1+\gamma}{1-\gamma}\right)^{|\mathbf{w}|-|\mathbf{u}|} \\ &= \left(\left(\frac{1}{2}\right)^{|\mathbf{v}|-|\mathbf{w}|}\right)^{1-\log(1+\gamma)}\left(\frac{1+\gamma}{1-\gamma}\right)^{|\mathbf{w}|-|\mathbf{u}|} \\ &\leq (8\widetilde{\varepsilon})^{1-\log(1+\gamma)}\left(\frac{1+\gamma}{1-\gamma}\right)^{\log(8c)}\end{aligned}$$

where we used Lemma 4.5 in the second inequality. □

5 Non-Trivial SVCS Mechanisms for Bounded Sensitivity Functions

In this section we show a mechanism, which we call $\overline{M}^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}$, that achieves $(\widetilde{\varepsilon}, O(1))$-SVCS for $\mathcal{F}_d$ – the class of functions with bounded sensitivity $d \in \mathbb{Z}^+$. By Theorem 4.4 this gives us a $(\varepsilon, \mathcal{SV}(\gamma))$-differentially private mechanism, where $\varepsilon \to 0$ as $\widetilde{\varepsilon} \to 0$. Furthermore, by our observation in Section 4, the mechanism is also $(\widetilde{\varepsilon}, \mathcal{U})$-differentially private. We highlight that for convenience, we parametrize the mechanism $\overline{M}^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}$ with the privacy parameter $\widetilde{\varepsilon}$ w.r.t. $\mathcal{U}$, and state the privacy and utility guarantees w.r.t. $\mathcal{SV}(\gamma)$ as a function of $\widetilde{\varepsilon}$ (see Lemma 5.1 and Lemma 5.2). For clarity, we focus on the case $d = 1$.

We start with the $(\widetilde{\varepsilon}, \mathcal{U})$-differentially private mechanism of Dwork et.al. [DMNS06], $M^{\mathsf{RLap}}_{\widetilde{\varepsilon}}(D, f) = f(D) + \mathsf{RLap}_{0,1/\widetilde{\varepsilon}}$. Note that since $M^{\mathsf{RLap}}_{\widetilde{\varepsilon}}$ is additive-noise, then any finite-precision implementation will also be additive-noise, and by Lemma 3.1 we know it cannot be non-trivial for $\mathcal{F}_1$ w.r.t. $\mathcal{SV}(\gamma)$. This is because the set of random coins that make the mechanism output $z \in \mathbb{Z}$ on two neighboring databases is *disjoint*. We will therefore need to make several changes to ensure not only that these sets overlap, but that their intersection is large, thus ensuring $\widetilde{\varepsilon}$-consistent sampling. Moreover, we must carefully implement our mechanism with finite precision so that the resulting mechanism is $(\widetilde{\varepsilon}, O(1))$-SV-consistent, ensuring that pathological cases, such as the one in Figure 4.1, do not occur. Finally, in performing all these changes we must also keep in mind that we want a good bound on utility. We first describe a new infinite-precision mechanism, which we call $M^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}$, and then show how to implement it with finite precision to ensure $(\widetilde{\varepsilon}, O(1))$-SV-consistency. The final mechanism $\overline{M}^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}$ is shown in Figure 5.1.

A New Infinite-Precision Mechanism. Recall that $M^{\mathsf{RLap}}_{\widetilde{\varepsilon}}(D, f) = f(D) + \mathsf{RLap}_{0,1/\widetilde{\varepsilon}} = \lfloor f(D) + \mathsf{Lap}_{0,1/\widetilde{\varepsilon}} \rceil$. For our new mechanism, which we call $M^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}$, we choose to perform the rounding step differently. $M^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}(D, f)$ computes $f(D) + \mathsf{Lap}_{0,1/\widetilde{\varepsilon}}$ as before but then rounds the final outcome to the nearest multiple of $1/\widetilde{\varepsilon}$. Recall that w.l.o.g. we can assume that $1/\widetilde{\varepsilon} \in \mathbb{Z}$ since otherwise we can choose a smaller $\widetilde{\varepsilon}$ so that this is indeed the case. Formally, $M^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}(D, f)$ computes $y \stackrel{\mathsf{def}}{=} f(D)$ and outputs $z \leftarrow 1/\widetilde{\varepsilon} \cdot \lfloor \widetilde{\varepsilon} \cdot \mathsf{Lap}_{y,1/\widetilde{\varepsilon}} \rceil$. We let Z_y denote the induced distribution of the outcome z. We remark that $M^{\mathsf{SVCS}}_{\widetilde{\varepsilon}}$ is not additive-noise, since the rounding ensures that the "noise" introduced is dependent on $y = f(D)$. Further, the output distribution is only defined on multiples of $1/\widetilde{\varepsilon}$, i.e. for $k/\widetilde{\varepsilon}$ where $k \in \mathbb{Z}$.

Consistent Sampling. We now give some intuition as to why this mechanism already satisfies $\widetilde{\varepsilon}$-consistent sampling. Since we are considering only queries in $\mathcal{F}_1$, for any two neighboring databases

D_1, D_2, we can assume w.l.o.g. that $f(D_1) = y$ and $f(D_2) = y - 1$. Then for $k \in \mathbb{Z}$,

$$\frac{\Pr[M_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(f, D_1) = k/\widetilde{\varepsilon}]}{\Pr[M_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(f, D_2) = k/\widetilde{\varepsilon}]} = \frac{\Pr\left[\frac{k-1/2}{\widetilde{\varepsilon}} \le y + \mathsf{Lap}_{0,1/\widetilde{\varepsilon}} < \frac{k+1/2}{\widetilde{\varepsilon}}\right]}{\Pr\left[\frac{k-1/2}{\widetilde{\varepsilon}} \le y - 1 + \mathsf{Lap}_{0,1/\widetilde{\varepsilon}} < \frac{k+1/2}{\widetilde{\varepsilon}}\right]}$$
$$= \frac{\Pr\left[\frac{k-1/2}{\widetilde{\varepsilon}} \le \mathsf{Lap}_{y,1/\widetilde{\varepsilon}} < \frac{k+1/2}{\widetilde{\varepsilon}}\right]}{\Pr\left[\frac{k-1/2}{\widetilde{\varepsilon}} + 1 \le \mathsf{Lap}_{y,1/\widetilde{\varepsilon}} < \frac{k+1/2}{\widetilde{\varepsilon}} + 1\right]}$$

Notice that both the intervals defined in the numerator and denominator have size $1/\widetilde{\varepsilon}$, and that the interval in the denominator is simply the interval in the numerator, shifted by 1. Therefore, their intersection is roughly a $1 - \widetilde{\varepsilon}$ fraction of their size, which is precisely what is required by $\widetilde{\varepsilon}$-consistent sampling. Of course, we now need to implement this $\widetilde{\varepsilon}$-consistent mechanism with finite precision, so as to achieve a stronger form of $(\widetilde{\varepsilon}, O(1))$-SV-consistency. For that, we will use *arithmetic coding* and some specific properties of the Laplace distribution.

From Infinite to Finite Precision via Arithmetic Coding. In what follows, we use the following notation: for a sequence $\mathbf{x} = x_1, x_2, \ldots \in \{0,1\}^*$, we define its *real representation* to be the real number $\textsc{real}(\mathbf{x}) \stackrel{\mathsf{def}}{=} 0.x_1x_2x_3\ldots \in [0,1]$. Arithmetic coding gives us a way to approximate any distribution X on $\mathbb{Z}$ from a bit string $\mathbf{r} \in \{0,1\}^*$, as follows. Let $\mathsf{CDF}^{\mathsf{X}}$ be the cumulative distribution of X, so that $\mathsf{X}(x) = \mathsf{CDF}^{\mathsf{X}}(x) - \mathsf{CDF}^{\mathsf{X}}(x-1)$. Let $s(x) \stackrel{\mathsf{def}}{=} \mathsf{CDF}^{\mathsf{X}}(x)$. Then the set of points $\{s(x)\}_{x\in\mathbb{Z}}$ partitions the interval $[0,1]$ into infinitely many intervals $\{I^{\mathsf{X}}(x) \stackrel{\mathsf{def}}{=} [s(x-1), s(x))\}_{x\in\mathbb{Z}}$, where $\mathsf{X}(x) = |I^{\mathsf{X}}(x)|$. Note that if a value $x \in \mathbb{Z}$ has zero probability, then we can simply ignore it as its corresponding interval will be empty. We can obtain distribution X from U by sampling a sequence of bits $\mathbf{r} = r_1, r_2, r_3, \ldots$ and outputting the unique $x \in \mathbb{Z}$ such that $\textsc{real}(\mathbf{r}) \in I^{\mathsf{X}}(x)$. Note that arithmetic coding has the very nice property that intervals $I^{\mathsf{X}}(x)$ and $I^{\mathsf{X}}(x+1)$ are always consecutive for any $x \in \mathbb{Z}$.

Since for some $x \in \mathbb{Z}$ we can have that $s(x)$ has an infinite binary decimal representation, there is no *a priori* bound on the number of coins to decide whether $\textsc{real}(\mathbf{r}) \in I^{\mathsf{X}}(x)$ or $\textsc{real}(\mathbf{r}) \in I^{\mathsf{X}}(x+1)$. To avoid this, we simply round each endpoint $s(x)$ to its most $n = n(x)$ significant figures, for some $n = n(x) > 1$ which potentially depends on x. We will need to make sure that $n(x)$ is *legal*, in the sense that rounding with respect to $n(x)$ should not cause intervals to "disappear" or for consecutive intervals to "overlap". We use a bar to denote rounded values: $\overline{s}(x)$ for the rounded endpoint, and $\overline{I}^{\mathsf{X}}(x)$ for the rounded interval.

A New Finite Precision Mechanism. We now show how to sample Z_y, the output distribution of $M_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(D, f)$ using arithmetic coding. This yields a new finite precision mechanism, which we call $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$, and let $\overline{\mathsf{Z}}_y$ be its output distribution which will approximate Z_y. The distribution Z_y is the Laplacian distribution $\mathsf{Lap}_{y,1/\widetilde{\varepsilon}}$ where for all $k \in \mathbb{Z}$, the probability mass in the interval $\left[\frac{k-1/2}{\widetilde{\varepsilon}}, \frac{k+1/2}{\widetilde{\varepsilon}}\right)$ collapses to the point $k/\widetilde{\varepsilon}$. Let $s_y(k) \stackrel{\mathsf{def}}{=} \mathsf{CDF}^{\mathsf{Z}_y}\left(\frac{k+1/2}{\widetilde{\varepsilon}}\right)$, and let $\overline{s}_y(k)$ be $s_y(k)$, rounded to its $n = n(y,k)$ most significant figures. Then the set of points $\{\overline{s}_y(k)\}_{k\in\mathbb{Z}}$ partition the interval $[0,1]$ into infinitely many intervals $\{\overline{I}_y(k) \stackrel{\mathsf{def}}{=} [\overline{s}_y(k-1), \overline{s}_y(k))\}_{k\in\mathbb{Z}}$, where $\Pr[\overline{\mathsf{Z}}_y = k/\widetilde{\varepsilon}] = |\overline{I}_y(k)|$. We obtain distribution $\overline{\mathsf{Z}}_y$ from U by sampling a sequence of bits $\mathbf{r} \in \{0,1\}^*$ and outputting $k/\widetilde{\varepsilon}$ where $k \in \mathbb{Z}$ is the unique integer such that $\textsc{real}(\mathbf{r}) \in \overline{I}_y(k)$. We have not yet defined what the precision $n = n(y,k)$ is; we will do this below, but first we give some intuition as to why $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ will satisfy $(\widetilde{\varepsilon}, O(1))$-SV-consistent sampling for some "good-enough" precision.

SV-Consistent Sampling. Recall that since we assume $f \in \mathcal{F}_1$, for any two neighboring databases D_1, D_2 we can assume that $f(D_1) = y$ and $f(D_2) = y - 1$, so that for any $k \in \mathbb{Z}$

$$\frac{\Pr[\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(f, D_1) = k/\widetilde{\varepsilon}]}{\Pr[\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(f, D_2) = k/\widetilde{\varepsilon}]} = \frac{\Pr[\overline{\mathsf{Z}}_y = k/\widetilde{\varepsilon}]}{\Pr[\overline{\mathsf{Z}}_{y-1} = k/\widetilde{\varepsilon}]} = \frac{|\overline{I}_y(k)|}{|\overline{I}_{y-1}(k)|}$$

We thus wish to prove that the mechanism has $(\widetilde{\varepsilon}, c)$-SV-consistent sampling where $T_1 = \overline{I}_y(k) \approx I_y(k)$ and $T_2 = \overline{I}_{y-1}(k) \approx I_{y-1}(k)$ in Definition 4.3. For now, let us assume that we use arithmetic coding with infinite precision, that is, we do not round the endpoints. We will give intuition as to why our mechanism satisfies an "infinite-precision analogue" of SV-consistent sampling. We can define $\mathbf{u}$ to be the longest prefix of all coins in $I \stackrel{\mathsf{def}}{=} I_y(k) \cup I_{y-1}(k)$, and let $\mathbf{u}_\ell \stackrel{\mathsf{def}}{=} \mathbf{u}, 0, 0, \ldots$ and $\mathbf{u}_r \stackrel{\mathsf{def}}{=} \mathbf{u}, 1, 1, \ldots$. Informally, $\mathbf{u}$ is the longest prefix such that $\mathbf{u}_\ell$ is to the left of I and $\mathbf{u}_r$ is to the right of I. Then an "infinite-precision analogue" of $(\cdot, O(1))$-SV-consistent sampling is the following:

$$\frac{\textsc{real}(\mathbf{u}_r) - \textsc{real}(\mathbf{u}_\ell)}{|I_y(k) \cup I_{y-1}(k)|} = O(1) \tag{5.1}$$

By construction, we have $\textsc{real}(\mathbf{u}_r) - \textsc{real}(\mathbf{u}_\ell) \approx 2^{-|\mathbf{u}|}$. Furthermore, arithmetic coding ensures that $I_y(k) \cap I_{y-1}(k) \neq \emptyset$; indeed, we can view $I_{y-1}(k)$ as having "shifted" $I_y(k)$ slightly to the right. We can therefore view $I = I_y(k) \cup I_{y-1}(k)$ as one single interval that is slightly bigger. Moreover, arithmetic coding and our use of the Laplacian distribution ensures that smaller intervals are farther from the center than bigger ones, and in fact, the size of the interval that contains I and everything to its right (or left, depending on whether I is to the right or left of $1/2$, respectively) is a constant factor of $|I|$. This means that $|I_y(k) \cup I_{y-1}(k)| = |I| = c \cdot 2^{-|\mathbf{u}|}$ for a constant c, and we thus obtain the ratio required in Equation (5.1).

Defining the Precision. Now we just need to round all the points $s_y(k)$ with enough precision so that the rounding is "legal" (i.e., preserves the relative sizes of all intervals $I_y(k)$ and $I_y(k) \backslash I_{y-1}(k)$ to within a constant factor), so that our informal analysis of SV-consistency above still holds after the rounding. Formally, we let $I'_y(k) \stackrel{\mathsf{def}}{=} I_y(k) \backslash I_{y-1}(k)$, be the interval containing the coins that will make the mechanism output $k/\widetilde{\varepsilon}$ when it is run on D_1 but output $(k-1)/\widetilde{\varepsilon}$ when run on D_2. We then let

$$n(y, k) = n(D, f, z) \stackrel{\mathsf{def}}{=} \log\left(\frac{1}{|I'_y(k)|}\right) + 3$$

and round $s_y(k)$ to its $\max(n(y+1, k+1), n_(y, k+1))$ most significant figures. The resulting mechanism $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ in shown in Figure 5.1.

We can now state our main results about SV-consistency and SV-privacy of our mechanism:

Lemma 5.1. *Mechanism $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ has $(27\widetilde{\varepsilon}, 57)$-SV-consistent sampling. In particular, $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ is $(27\widetilde{\varepsilon}, \mathcal{U})$-differentially private and $(\varepsilon, \mathcal{SV}(\gamma))$-differentially private for $\varepsilon = (216\widetilde{\varepsilon})^{1-\log(1+\gamma)}\left(\frac{1+\gamma}{1-\gamma}\right)^9$.*

Utility. We have showed that our mechanism $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ achieves $(\widetilde{\varepsilon}, O(1))$-SV-consistent sampling and thus $(\varepsilon, \mathcal{SV}(\gamma))$-differential privacy, where $\varepsilon \to 0$ as $\widetilde{\varepsilon} \to 0$. We now argue that the mechanism also has non-trivial utility. It is easy to see that when the randomness source is uniform, rounding to the nearest multiple of $1/\widetilde{\varepsilon}$ only affects utility by an additive factor of $1/\widetilde{\varepsilon}$, thus maintaining $(O(1/\widetilde{\varepsilon}), \mathcal{U})$-utility. This is comparable to the utility of the mechanism $\overline{M}_{\varepsilon}^{\mathsf{RLap}}$ of [DMNS06]; see Lemma 2.11.

To analyze utility w.r.t. $\mathcal{SV}(\gamma)$, we first bound the probability that a coin sampled from a γ-Santha-Vazirani distribution $\mathbf{r} \leftarrow \mathsf{SV}(\gamma)$, lands in the interval $\overline{I}_y(k)$, since this is the probability that

$\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ outputs $k/\widetilde{\varepsilon}$ when the real answer is $y = f(D)$. We consider the longest commont prefix $\mathbf{a}$ of all coins in $\overline{I}_y(k)$ and upper bound the probability of landing in $\overline{I}_y(k)$ by the probability that $\mathbf{r}$ has $\mathbf{a}$ as prefix. We can then upper bound this probability by $\left(\frac{1+\gamma}{2}\right)^{\log\left(\frac{1}{|\overline{I}_y(k)|}\right)}$. This allows us to prove, by multiplying by $|k/\widetilde{\varepsilon} - y|$ and summing over all $k \in \mathbb{Z}$, that any γ-Santha-Vazirani distribution can worsen utility by at most an (asymptotic) factor of $\frac{1}{1-\gamma}$.

Lemma 5.2. *Mechanism $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ has $(O(1/\widetilde{\varepsilon}), \mathcal{U})$-utility and $(\rho, \mathcal{SV}(\gamma))$-utility, where $\rho = O\left(\frac{1}{\widetilde{\varepsilon}} \cdot \frac{1}{1-\gamma}\right)$.*

Finally, combining Lemma 5.1 and Lemma 5.2 yields our main theorem.

Theorem 5.3. *For all $\gamma < 1$, $\overline{\mathcal{M}}^{\mathsf{SVCS}} = \{\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}\}$ is a class of non-trivial mechanisms for $\mathcal{F}_1$ w.r.t. $\mathcal{SV}(\gamma)$.*

$\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(D, f\ ; \mathbf{r})$: Compute $y \stackrel{\mathsf{def}}{=} f(D)$ and output a sample from the distribution $\mathbb{Z}_y \stackrel{\mathsf{def}}{=} 1/\widetilde{\varepsilon} \cdot \lfloor \widetilde{\varepsilon} \cdot \mathsf{Lap}_{y,1/\widetilde{\varepsilon}} \rceil$ by using arithmetic coding as explained below.

- Let $n(y,k) = n(D,f,z) \stackrel{\mathsf{def}}{=} \log\left(\frac{1}{|I'_y(k)|}\right) + 3$ and let $\mathbf{r}'_{y,k}$ be the $n(y,k)$ most significant figures of $\mathbf{r}$.
- Output the the unique $z = k/\widetilde{\varepsilon}$ such that $\frac{k-1/2}{\widetilde{\varepsilon}} \leq \textsc{real}(\mathbf{r}'_{y,k}) < \frac{k+1/2}{\widetilde{\varepsilon}}$.

Figure 5.1: Finite precision mechanism $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ that has $(27\widetilde{\varepsilon}, 57)$-SV-consistent sampling.

References

[ACRT99] Alexander E. Andreev, Andrea E. F. Clementi, José D. P. Rolim, and Luca Trevisan. Weak random sources, hitting sets, and bpp simulations. *SIAM J. Comput.*, 28(6):2103–2116, 1999.

[BD07] Carl Bosley and Yevgeniy Dodis. Does privacy require true randomness? In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2007.

[BGMZ97] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks*, 29(8-13):1157–1166, 1997.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *FOCS*, pages 196–205. IEEE Computer Society, 2004.

[DS02] Yevgeniy Dodis and Joel Spencer. On the (non)universality of the one-time pad. In *FOCS*, pages 376–385. IEEE Computer Society, 2002.

[GRS09] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In Michael Mitzenmacher, editor, *STOC*, pages 351–360. ACM, 2009.

[Hol07] Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 411–419. ACM, 2007.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In Leonard J. Schulman, editor, *STOC*, pages 705–714. ACM, 2010.

[Man94] Udi Manber. Finding similar files in a large file system. In *Proceedings of the USENIX Winter 1994 Technical Conference on USENIX Winter 1994 Technical Conference*, pages 2–2, Berkeley, CA, USA, 1994. USENIX Association.

[MMP+10] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil P. Vadhan. The limits of two-party differential privacy. In *FOCS*, pages 81–90. IEEE Computer Society, 2010.

[MNW98] Alistair Moffat, Radford M. Neal, and Ian H. Witten. Arithmetic coding revisited. *ACM Trans. Inf. Syst.*, 16(3):256–294, 1998.

[MP90] James L. McInnes and Benny Pinkas. On the impossibility of private key cryptography with weakly random keys. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 421–435. Springer, 1990.

[MW97] Ueli M. Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 307–321. Springer, 1997.

[RVW04] Omer Reingold, Salil Vadhan, and Avi Widgerson. A note on extracting randomness from Santha-Vazirani sources. Personal communication, 2004.

[SV86] Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *J. Comput. Syst. Sci.*, 33(1):75–87, 1986.

[VV85] Umesh V. Vazirani and Vijay V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *FOCS*, pages 417–428. IEEE Computer Society, 1985.

[WNC87] Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Commun. ACM*, 30(6):520–540, 1987.

[Zuc96] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, 1996.

A Proofs

A.1 Proof of Lemma 5.1

In Section 5, we gave some intuition to argue that the infinite precision mechanism $M_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ has $\widetilde{\varepsilon}$-consistent sampling. Here we will prove formally that this is indeed the case (modulo a constant factor). Recall that we define $s_y(k) \stackrel{\mathsf{def}}{=} \mathsf{CDF}^{\mathsf{Z}_y}\left(\frac{k+1/2}{\widetilde{\varepsilon}}\right)$ and $I_y(k) \stackrel{\mathsf{def}}{=} [s_y(k-1), s_y(k))$. Further recall that $I'_y(k) \stackrel{\mathsf{def}}{=} I_y(k) \backslash I_{y-1}(k) = [s_y(k-1), s_{y-1}(k-1))$.

Lemma A.1. *For all $y, k \in \mathbb{Z}$,*

$$\frac{|I'_y(k)|}{|I_{y-1}(k)|} \leq 6\widetilde{\varepsilon}.$$

Proof. We must consider four cases:

1. If $\frac{1}{2} \leq s_y(k-1) < s_{y-1}(k-1) < s_{y-1}(k-1)$, then $\frac{|I'_y(k)|}{|I_{y-1}(k)|} = \frac{e^{\widetilde{\varepsilon}+1}-e}{e-1}$.
2. If $s_y(k-1) < \frac{1}{2} \leq s_{y-1}(k-1) < s_{y-1}(k-1)$, then $\frac{|I'_y(k)|}{|I_{y-1}(k)|} \leq \frac{2(e^{\widetilde{\varepsilon}+1}-e)}{e-1}$.
3. If $s_y(k-1) < s_{y-1}(k-1) < \frac{1}{2} \leq s_{y-1}(k-1)$, then $\frac{|I'_y(k)|}{|I_{y-1}(k)|} \leq \frac{1-e^{-\widetilde{\varepsilon}}}{2(e-1)}$.
4. If $s_y(k-1) < s_{y-1}(k-1) < s_{y-1}(k-1) < \frac{1}{2}$, then $\frac{|I'_y(k)|}{|I_{y-1}(k)|} = \frac{1-e^{-\widetilde{\varepsilon}}}{e-1}$.

For $\widetilde{\varepsilon} \in (0,1)$, we have

$$\frac{1-e^{-\widetilde{\varepsilon}}}{2(e-1)} < \frac{1-e^{-\widetilde{\varepsilon}}}{e-1} < \frac{e^{\widetilde{\varepsilon}+1}-e}{e-1} < \frac{2(e^{\widetilde{\varepsilon}+1}-e)}{e-1} < 6\widetilde{\varepsilon}$$

. □

We now prove that the *finite-precision* mechanism $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ has $O(\widetilde{\varepsilon})$-consistent sampling, as defined in Definition 4.1. Recall that we round the endpoints of $s_y(k-1)$ and $s_{y-1}(k-1)$ of the interval $I'_y(k)$ to its $n \stackrel{\mathsf{def}}{=} \max(n(y,k), n(y-1,k))$ most significant figures, where $n(y,k) \stackrel{\mathsf{def}}{=} \log\left(\frac{1}{|I'_y(k)|}\right) + 1$. Further recall that for $O(\widetilde{\varepsilon})$-consistent sampling, we wish to prove that the number of n-bit strings in $\overline{I'}_y(k)$ is an $O(\widetilde{\varepsilon})$ fraction of the number of coins in $\overline{I}_{y-1}(k)$. We thus define the following notation. For an interval $I = [a,b) \subset [0,1]$, we let $\textsc{str}(I,n) \stackrel{\mathsf{def}}{=} \{\mathbf{r} \in \{0,1\}^n \mid \textsc{real}(\mathbf{r}) \in I\}$ be the set of all n-bit strings whose real representation lies in I. As an intermediate step towards proving $O(\widetilde{\varepsilon})$-consistent sampling, we show that $T_1 \backslash T_2 = \textsc{str}(\overline{I'}_y(k), n)$ has constant size, and $T_2 = \textsc{str}(\overline{I}_{y-1}(k), n)$ has size $\Omega(1/\widetilde{\varepsilon})$.

We begin by showing that rounding the endpoints as described, does not alter the size of the intervals $I'_y(k)$ or $I_{y-1}(k)$ by much.

Lemma A.2. *For all $y, k \in \mathbb{Z}$ and $n = \max(n(y,k), n(y-1,k))$, we have*

$$|I'_y(k)| - 2^{-n} \leq |\overline{I'}_y(k)| \leq |I'_y(k)| + 2^{-n}$$
$$|I_{y-1}(k)| - 2^{-n} \leq |\overline{I}_{y-1}(k)| \leq |I_{y-1}(k)| + 2^{-n}$$

Proof. Since each endpoint changes by at most $2^{-n}/2$, then rounding changes the size of the interval by at most 2^{-n}. □

Let $n = \max(n(y,k), n(y-1,k))$. The following lemma says that the number of n-bit strings inside $\overline{I'}_y(k)$ is constant, while the number of n-bit strings inside $\overline{I}_{y-1}(k)$ is at least a constant factor of $1/\widetilde{\varepsilon}$. Since for consistent sampling we are interested in rounding the ratio between the number of n-bit strings in $\overline{I'}_y(k)$ and $\overline{I}_{y-1}(k)$, this will yield $O(\widetilde{\varepsilon})$-consistent sampling.

Lemma A.3. *For all $y, k \in \mathbb{Z}$ and $n = \max(n(y,k), n(y-1,k))$, we have,*

$$|\textsc{str}(\overline{I'}_y(k), n)| \leq 9 \quad \textit{and} \quad |\textsc{str}(\overline{I}_{y-1}(k), n)| \geq \frac{1}{3\widetilde{\varepsilon}}$$

Proof. We can see $|\overline{I'}_y(k)|$ as the probability of sampling a sequence $\mathbf{r}$ from U_n such that $\mathbf{r} \in \textsc{str}(\overline{I'}_y(k), n)$. Therefore,

$$|\overline{I'}_y(k)| = \sum_{\mathbf{r} \in \textsc{str}(\overline{I'}_y(k), n)} \left(\frac{1}{2}\right)^n = |\textsc{str}(\overline{I'}_y(k), n)| \cdot 2^{-n}$$

and by Lemma A.2,

$$|\textsc{str}(\overline{I'}_y(k), n)| = 2^n \cdot |\overline{I'}_y(k)| \leq 2^{n(y,k)}(|I'_y(k)| + 2^{-n(y,k)}) \leq 2^{\log\left(\frac{1}{|I'_y(k)|}\right)+3}|I'_y(k)| + 1 = 9$$

Similarly,

$$|\overline{I}_{y-1}(k)| = \sum_{\mathbf{r} \in \textsc{str}(\overline{I}_{y-1}(k), n)} \left(\frac{1}{2}\right)^n = |\textsc{str}(\overline{I}_{y-1}(k), n)| \cdot 2^{-n}$$

and by Lemma A.2 and Lemma A.1,

$$|\textsc{str}(\overline{I}_{y-1}(k), n)| = 2^n \cdot |\overline{I}_{y-1}(k)| \geq 2^n(|I_{y-1}(k)| - 2^{-n}) = 8 \cdot \frac{|I_{y-1}(k)|}{|I'_y(k)|} - 1 \geq \frac{8}{6\widetilde{\varepsilon}} - 1 = \frac{1}{3\widetilde{\varepsilon}}.$$

□

Corollary A.4. *For all $y, k \in \mathbb{Z}$ and $n = \max(n(y,k), n(y-1,k))$, we have*

$$\frac{|\textsc{str}(\overline{I'}_y(k), n)|}{|\textsc{str}(\overline{I}_{y-1}(k), n)|} \leq 27\widetilde{\varepsilon}$$

In particular, $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ has $27\widetilde{\varepsilon}$-consistent sampling.

Corollary A.4 shows that $\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}$ has $27\widetilde{\varepsilon}$-consistent sampling. We now show that in fact, it has $(27\widetilde{\varepsilon}, c)$-SV-consistent sampling for $c = O(1)$.

Proof of Lemma 5.1: By Lemma A.3, we know that $|T_1 \cup T_2| \geq |T_2| = |\textsc{str}(\overline{I}_{y-1}(k), n)| \geq 1/3\widetilde{\varepsilon}$. It thus suffices to prove that $|\textsc{suffix}(\mathbf{u}, n)| = O(1/\widetilde{\varepsilon})$, where $\mathbf{u}$ is the longest common prefix of all strings in $I \stackrel{\mathsf{def}}{=} I_y(k) \cup I_{y-1}(k)$. Let $\overline{\mathbf{u}}$ be the longest common prefix of all strings in $\overline{I} \stackrel{\mathsf{def}}{=} \overline{I}_y(k) \cup \overline{I}_{y-1}(k)$. By rounding we must have that $|\textsc{suffix}(\mathbf{u}, n)| \leq |\textsc{suffix}(\overline{\mathbf{u}}, n)| + 2$. Moreover, we can bound $|\textsc{suffix}(\overline{\mathbf{u}}, n)|$ by bounding the number of n-bit strings to the left or right of $\overline{I}$ (depending on where $\overline{I}_y(k)$ and $\overline{I}_{y-1}(k)$ are located in the interval $[0,1]$).

Recall that $I'_y(k) = [s_y(k-1), s_{y-1}(k-1))$ and $I_{y-1}(k) = [s_{y-1}(k-1), s_{y-1}(k-1))$. We first calculate the size of the interval $[s_y(k), 1]$ (resp. $[0, s_{y-1}(k)]$), that is, the interval taking all values to the left (resp. to the right) and including I. This will give us a good approximation of the size of $[\overline{s}_y(k), 1]$ (resp. $[0, \overline{s}_{y-1}(k)]$). From this we can calculate how many n-bit strings there are to left or right of $\overline{I}$. We have to consider four cases:

1. If $\frac{1}{2} \leq s_y(k-1) < s_{y-1}(k-1) < s_{y-1}(k-1)$, then in this case, $|I'_y(k)| = |I'_{y+1}(k)| \cdot (1/e^{\widetilde{\varepsilon}})$.
2. If $s_y(k-1) < \frac{1}{2} \leq s_{y-1}(k-1) < s_{y-1}(k-1)$, then in this case, $|I'_y(k)| \geq |I'_{y+1}(k)| \cdot (1/(2e^{\widetilde{\varepsilon}}))$.
3. If $s_y(k-1) < s_{y-1}(k-1) < \frac{1}{2} \leq s_{y-1}(k-1)$, then in this case, $|I'_{y+1}(k)| \geq |I'_y(k)| \cdot (1/(2e^{\widetilde{\varepsilon}}))$.
4. If $s_y(k-1) < s_{y-1}(k-1) < s_{y-1}(k-1) < \frac{1}{2}$, then in this case, $|I'_{y+1}(k)| = |I'_y(k)| \cdot (1/e^{\widetilde{\varepsilon}})$.

In all cases, $I'_y(k)$ and $I'_{y+1}(k)$ are consecutive intervals and $I'_{y+1}(k)$ is located to the left of $I'_y(k)$. We only analyze case 2; the other cases are analogous and yield the same bound.

Let $J \stackrel{\text{def}}{=} [s_y(k), 1]$. Then,

$$|J| = \sum_{j=-\infty}^{y} |I'_j(k)| \leq \sum_{j=-\infty}^{y} |I'_y(k)|(e^{-\widetilde{\varepsilon}}/2)^{y-j} = |I'_y(k)| \sum_{j=0}^{\infty} (e^{-\widetilde{\varepsilon}}/2)^j = \frac{|I'_y(k)|}{1 - e^{-\widetilde{\varepsilon}}/2} \leq \frac{2 \cdot |I'_y(k)|}{\widetilde{\varepsilon}}$$

Since we round one endpoint, this means that for $\overline{J} = [\overline{s}_y(k), 1]$,

$$|\overline{J}| \leq \frac{2 \cdot |I'_y(k)|}{\widetilde{\varepsilon}} + \frac{2^{-n(y,k)}}{2} = |I'_y(k)| \left(\frac{2}{\widetilde{\varepsilon}} + \frac{1}{16} \right)$$

At the same time, $|\overline{J}|$ is the probability of sampling a sequence $\mathbf{r}$ from U_n such that $\mathbf{r} \in \textsc{str}(\overline{J}, n)$. Therefore,

$$|\overline{J}| = \sum_{\mathbf{r} \in \textsc{str}(\overline{J}, n)} \left(\frac{1}{2} \right)^n = |\textsc{str}(\overline{J}, n)| \cdot 2^{-n}$$

Thus,

$$|\textsc{str}(\overline{J}, n)| = 2^n \cdot |\overline{J}| \leq 2^{n(y,k)} |I'_y(k)| \left(\frac{2}{\widetilde{\varepsilon}} + \frac{1}{16} \right) = \frac{16}{\widetilde{\varepsilon}} + \frac{1}{2} \leq \frac{17}{\widetilde{\varepsilon}}$$

We can therefore conclude that

$$|\textsc{suffix}(\mathbf{u}, n)| \leq |\textsc{suffix}(\overline{\mathbf{u}}, n)| + 2 \leq |\textsc{str}(\overline{J}, n)| + 2 \leq \frac{17}{\widetilde{\varepsilon}} + 2 \leq \frac{19}{\widetilde{\varepsilon}}$$

Finally, since we know that $|T_1 \cup T_2| \geq |T_2| = |\textsc{str}(\overline{I}_{y-1}(k), n)| \geq 1/3\widetilde{\varepsilon}$, we conclude that

$$\frac{|\textsc{suffix}(\mathbf{u}, n)|}{|T_1 \cup T_2|} \leq 57$$

□

A.2 Proof of Lemma 5.2

Proof of Lemma 5.2: For any $\mathsf{SV}(\gamma) \in \mathcal{SV}(\gamma)$, we want to bound $\mathbb{E}_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)}[|f(D) - \overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}|]$. Let $y = f(D)$, then:

$$\mathbb{E}_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)} \left[\left| \overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(D, f;\ \mathbf{r}) - y \right| \right] = \sum_{k=-\infty}^{\infty} \Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)} \left[\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(D, f; \mathbf{r}) = k/\widetilde{\varepsilon} \right] \cdot |k/\widetilde{\varepsilon} - y|$$

We upper bound $\Pr\left[\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(D, f) = k/\widetilde{\varepsilon} \right]$. Under the uniform distribution, this probability is exactly $\overline{I}_y(k)$. For $\mathsf{SV}(\gamma) \in \mathcal{SV}(\gamma)$ we can upper bound this probability by noticing that if $\mathbf{a}$ is the common ancestor of all strings in $\overline{I}_y(k)$, then

$$\Pr_{\mathbf{r} \leftarrow \mathsf{SV}(\gamma)} \left[\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(D, f;\ \mathbf{r}) = k/\widetilde{\varepsilon} \right] \leq \left(\frac{1+\gamma}{2} \right)^{n(y,k) - |\mathbf{a}|} \leq \left(\frac{1+\gamma}{2} \right)^{\log\left(\frac{1}{|\overline{I}_y(k)|} \right)}$$

We know $|\overline{I}_y(k)| \leq 9/8 \cdot |I_y(k)|$ by Lemma A.2. Furthermore, $|I_y(k)| = (1/2)e^{-1/2}(e-1)e^{-|k - \widetilde{\varepsilon} y|}$, so that $\log\left(\frac{1}{|\overline{I}_y(k)|} \right) \geq |k - \widetilde{\varepsilon} y|$.

Moreover, w.l.o.g. we can assume that $0 \le y \le 1/\widetilde{\varepsilon}$ since translation by $1/\widetilde{\varepsilon}$ does not affect our result. Then,

$$\begin{aligned}
&\sum_{k=-\infty}^{\infty} \Pr_{\mathbf{r}\leftarrow\mathsf{SV}(\gamma)}\left[\overline{M}_{\widetilde{\varepsilon}}^{\mathsf{SVCS}}(D,f;\ \mathbf{r}) = k/\widetilde{\varepsilon}\right] \cdot |k/\widetilde{\varepsilon} - y| \\
&\quad \le \frac{1}{\widetilde{\varepsilon}} \cdot \sum_{k=1}^{\infty} \left(\frac{1+\gamma}{2}\right)^{|k-\widetilde{\varepsilon}y|} |k - \widetilde{\varepsilon}y| + \frac{1}{\widetilde{\varepsilon}} \cdot \sum_{k=-\infty}^{0} \left(\frac{1+\gamma}{2}\right)^{|k-\widetilde{\varepsilon}y|} |k - \widetilde{\varepsilon}y| \\
&\quad \le \frac{1}{\widetilde{\varepsilon}} \cdot \sum_{k=1}^{\infty} \left(\frac{1+\gamma}{2}\right)^{k-1} k + \frac{1}{\widetilde{\varepsilon}} \cdot \sum_{k=-\infty}^{0} \left(\frac{1+\gamma}{2}\right)^{-k} (-k+1) \\
&\quad = \frac{2}{\widetilde{\varepsilon}} \cdot \sum_{k=1}^{\infty} \left(\frac{1+\gamma}{2}\right)^{k-1} k = \frac{2}{\widetilde{\varepsilon}} \cdot \frac{1}{1 - \left(\frac{1+\gamma}{2}\right)^2} = O\left(\frac{1}{\widetilde{\varepsilon}} \cdot \frac{1}{1-\gamma}\right).
\end{aligned}$$

□