Zip to Zip-it: Compression to Achieve Local Differential Privacy

Francesco Taurone, Daniel E. Lucani and Qi Zhang

DIGIT, Department of Electrical and Computer Engineering, Aarhus University

 $\{francesco.taurone,\,daniel.lucani,\,qz\}@ece.au.dk$

Abstract—Local differential privacy techniques for numerical data typically transform a dataset to ensure a bound on the likelihood that, given a query, a malicious user could infer information on the original samples. Queries are often solely based on users and their requirements, limiting the design of the perturbation to processes that, while privatizing the results, do not jeopardize their usefulness. In this paper, we propose a privatization technique called Zeal, where perturbator and aggregator are designed as a unit, resulting in a locally differentially private mechanism that, by-design, improves the compressibility of the perturbed dataset compared to the original, saves on transmitted bits for data collection and protects against a privacy vulnerabilities due to floating point arithmetic that affect other state-of-the-art schemes. We prove that the utility error on querying the average is invariant to the bias introduced by Zeal in a wide range of conditions, and that under the same circumstances, Zeal also guarantee protection against the aforementioned vulnerability. Our numerical results show up to 94 % improvements in compression and up to 95 % more efficient data transmissions, while keeping utility errors within 2 %.

Index Terms—differential privacy, compression, floating point.

I. INTRODUCTION

The way we collect, process and store data is usually designed to achieve goals like low communication costs, private queries or reduced database size. These are generally competing objectives, and tend to be addressed with techniques that specialize in a single aspect, without synergizing with each other. For sensitive data, it is important to use privatization strategies to avoid leakage of information about their sources or owners. The design of a privatization mechanism depends on the queries it is meant to protect against, since its effectiveness for different data aggregations, like averaging or finding extrema, might vary. Queries are usually considered as inputs to the system, which limits the way we can perturb data without causing unacceptable errors on the output. Moreover, when dealing with floating point numbers, some privatization techniques relying on adding noise to the original samples are vulnerable to privacy leaks due to floating point arithmetic [1].

A. Contribution

This paper proposes a novel privatization and data-sharing method based on local differential privacy (LDP) called Zeal,





Fig. 1. Comparison between a standard perturbator - aggregator scheme and *Zeal*. In the latter, smaller packets have to be transmitted to the database, and they are more compressible than the original, while achieving the same result.

which combines the design of both the perturbation algorithm and the query by extending the *piecewise mechanism* in [2] with the *addition transform* in [3]. The combination of these mechanisms with a judicious selection of *Zeal* parameter \overline{A} , allow to achieve improving compression and transmission savings for growing \overline{A} with low utility errors, while simultaneously protecting against floating point number vulnerabilities that would otherwise affect the original piecewise mechanism.

This vulnerability has other solutions in the literature, ranging from approximating the output of the perturbator [1], to strategies using smart iterative noise sampling [4] or integer implementations for floating point perturbation [5]. We argue that Zeal, while being equally effective in solving the vulnerability, require simpler computations and theoretical analyses. There are also other attempts of studying the relation between privatizing data and its compressibility. The results in [6] show how specific data representations, like the wavelet transform [7], when combined with proper perturbation can achieve better compressibility of data under specific privacy constraints. However, those techniques rely on privacy notions different from differential privacy, and focus on how to avoid possible vulnerabilities in the choice of additive noise. In [8], the aim is to provide minimal private representations of the data transmitted to the database by building succinct histograms. This proves to be effective in terms of accuracy

This work was supported by the IoTalentum Project within the Framework of Marie Skłodowska-Curie Actions Innovative Training Networks (ITN)-European Training Networks (ETN), which is funded by the European Union Horizon 2020 Research and Innovation Program under Grant 953442.

for frequency estimation, but does not cover the query of the average nor investigate the design of dedicated aggregators in combination to perturbators to enhance compression.

B. System model

We consider a system illustrated in Fig. 1, where n IoT sensors of the same kind $\{S_1, \ldots, S_n\}$ need to send a single floating point number x_i each to an untrusted database DB for storage. We assume that the final user is interested in querying the DB for the average of the dataset $DS = \{x_1, \ldots, x_n\},\$ where the minimum and maximum value the sensors are reasonably expected to produce are represented using \overline{H} and h so that $x_i \in [\bar{H} - h, \bar{H} + h]$. Moreover, the data being collected are sensitive, therefore mandating the need of being privatized with a perturbation function $f(x_1, \ldots, x_n)$ before reaching the DB. We propose a specific perturbator $x_i^* = f^*(x_i, \bar{A})$, where the A is a publicly known parameter common to all sensors, in combination with a modified version of the average query, namely AVG* $\{x_1^*, \ldots, x_n^*, \bar{A}\}$. Future work can consider couplings where each sensor may have a different A. This new method guarantees equal privatization level, while allowing for transmission savings and better compressibility in the DB. An example is a pharmaceutical company monitoring the average glucose measurements of their devices for patients with diabetes. Throughout this paper, we rely on the floating point standard IEEE-754 [9] with double precision (64-bits).

II. BACKGROUND

A. Local differential privacy

The main goal of differential privacy is to give a bound on the likelihood that the result of a query to a dataset will allow unwanted insights on its original values [10]. This is achieved by using a *perturbator function*, where the cost for this privatization is that the private query result deviates from the one on the original data: we refer to this as utility error. In the central model of differential privacy [10], the randomization is applied directly on the collection of real data, which implies the need of a trusted party to receive the data and perturb them. The *local model* assumes the presence of no trusted party, meaning that the randomization needs to occur before data is shared with the DB. Despite resulting in generally higher utility errors, the local version intuitively provides a more secure way of collecting and sharing data. Data leakages from the DB, or malicious cloud providers, pose less of a threat than in the central model, since they only hold an already privatized version of the sensitive data. We exploit the fact that the local privatization needs to happens between sensors and DB to design a perturbator reducing transmission data size. In order to quantify privacy, we use the privacy budget $\epsilon > 0$ which controls the privacy-utility trade off: the more privacy we want, the lower ϵ we should choose.

Definition II.1. A perturbation function f is ϵ -locally differentially private if and only if for any two inputs $x_i \neq x_j$ in the domain of f, and for any output x^* of f, we have

$$P[f(x_i) = x^*] \le e^{\epsilon} \cdot P[f(x_j) = x^*].$$
 (1)



Fig. 2. Example of piecewise mechanism PDF. After perturbation, the privatized version of both x_i and x_j could be any $x^* \in [x_{\min}^*, x_{\max}^*]$.

Given a dataset $DS = \{x_1, x_2, \dots, x_n\}$ and a privacy budget ϵ , we want to transform each x_i to x_i^* with a randomizer $f : x_i \to x_i^*$ so that Eq. (1) holds.

B. Piecewise LDP mechanism

The original locally differentially private piecewise mechanism [2] is a randomizer f whose key features are its bounded output $x_i^* \in [x_{\min}^*, x_{\max}^*]$, its lower utility error compared to other locally differentially private mechanisms, like the Laplace transformation [10], and the fact that it is unbiased, namely the expectation of x_i^* is $\mathbb{E}[x_i^*] = x_i$. Another peculiar characteristic is that the probability density (PDF) from which the privatized values x_i^* are sampled, changes shape according to x_i , while x_{\min}^* and x_{\max}^* are fixed, as per Fig. 2. Wang et al. [2] limited the original perturbator to DS with $x_i \in [-1.0, 1.0]$, mentioning the steps to generalize it to DS in $x_i \in [-r, r]$ by scaling the dataset. In Subsection III-A we propose a generalization of the perturbator itself so that $x_i \in [\overline{H} - h, \overline{H} + h]$ and introduce the parameter \overline{A} for introducing a bias to the output.

C. Addition transform and floating point compression

Data compression algorithms usually rely on identifying repeated symbols or sequences in the data being processed, in order to provide a compact representation of the same information. Considering a dataset of n floating point numbers $DS = \{x_1, \ldots, x_n\}$, a specific kind of repeated pattern is when the j - th bit of the 64 bits in each x_i has the same binary value $\forall x_i \in DS$. In this case, it is a shared bit. The addition transform [3] aims to improve DS compressibility by increasing the number of shared bits in a dataset upon transforming the original values through the addition

$$\tilde{x}_i = x_i + \bar{A} \quad \forall x_i \in \mathrm{DS}.$$
 (2)

The reason why the new DS typically has more shared bits than DS lies in the binary representation of a floating point number x, which is composed of a sign, an exponent and a mantissa. We refer to them as ν , E and M, when interpreted as unsigned integer, and use them to compute x with

$$x = (-1)^{\nu} \cdot 2^{E_U} \cdot (1 + M \cdot 2^{-L}), \tag{3}$$

where $E_U = E - b$ is the unbiased exponent, b is its bias and L is the length of the mantissa in bits: in double precision, b = 1023 and L = 52. From Eq. (3), we notice that the smallest difference between two numbers $x_1 > x_2$ having the

TABLE I Nomenclature

-			~ . ~ .		
А	Bias of f^*	CR	Compression Ratio	TR	Transmission Ratio
x	DS sample	DS, DS^*	Dataset and privatized dataset	S _{DS}	Sum of x_i in DS
x^*_{\max}	Max feasible x_i^*	Δ_{AVG}	Absolute error on the average	ULP(x)	Unit in the last place of x
x_{\min}^*	Min feasible x_i^*	AVG*	Zeal aggregator	$L(\cdot), R(\cdot)$	Parameters of PDF
\overline{E}	Exponent	LDP	Local differential privacy	δ_{AVG}	Relative error on the average
x^*	Output of f^*	p_C	Cumulative probability	$\delta_{\mathbb{E}(x_i)}$	Relative error on expected value of x_i^*
f^*	Zeal perturbator	\mathcal{F}	Error estimation due to \bar{A}		Number of samples in DS
f	Perturbator	h, \bar{H}	Parameters on feasible DS	γ_{\min}	Min number of shared bits per sample
E_U	Unbiased exponent	p	PDF probability level	$P(\cdot)$	Probability
E_U^*	Selected unbiased exponent	ϵ	Privacy budget		

same unbiased exponent E_U^* is $x_1 - x_2 = 2^{E_U^* - 52}$. It is called *unit in the last place*, or ULP(·), defined as

ULP
$$\left(x \in \left[2^{E_U^*}, 2^{E_U^*+1}\right)\right) = 2^{E_U^*-52}.$$
 (4)

ULP(x) effectively represents the *precision* of x, since all $x' \in (x - \text{ULP}(x)/2; x + \text{ULP}(x)/2)$, assuming rounding to nearest, will be represented and stored as x. If we apply Eq. (2) choosing \overline{A} large enough so that we can fit all \tilde{x}_i in a single exponent region $[2^{E_U^*}, 2^{E_U^*+1}]$, the k most significant bits in \tilde{x}_i mantissas will be shared when

$$\exists j \in \left[0, 2^{k} - 1\right] \subset \mathbb{N} : \tilde{x_{i}} \in 2^{E_{U}^{*}} \cdot \left[1 + \frac{j}{2^{k}}, 1 + \frac{j + 1}{2^{k}}\right).$$
(5)

Note that the transformation in Eq. (2) is lossy, as shown in [3]. A function g is lossy when, for some x_i

$$g^{-1}\left(g\left(x_{i}\right)\right) \neq x_{i}.$$
(6)

III. ZEAL

The original formulation of the piecewise mechanism is susceptible to a privacy vulnerability due to floating point arithmetic. Since the image of the perturbator $f(x_i)$ is a finite set of floating point numbers \mathcal{I}_{x_i} , and for some $x_i \neq x_j$, $\mathcal{I}_{x_i} - \mathcal{I}_{x_j} \neq \emptyset$, some privatized samples can be traced back to their original datum, breaking Eq. (1). Zeal's perturbator $f^*(x_i, \bar{A})$ extends $f(x_i)$ feasible inputs and introduces the bias \bar{A} : in Section IV, we show that the extended images $\mathcal{I}_{x_i}^*$ and $\mathcal{I}_{x_j}^*$ guarantee $\mathcal{I}_{x_i}^* - \mathcal{I}_{x_j}^* = \emptyset \ \forall x_i, x_j \in DS$, solving the vulnerability. Moreover, since a similar bias is used as part of the addition transform to improve compressibility, we benefit from both advantages with the same concept.

A. Extension of Piecewise Mechanism

Given the input dataset $DS = \{x_1, \dots, x_n\}$ and the desired privacy level ϵ , the PDF of the output $x_i^* = f^*(x_i, \overline{A})$ is

$$P\left(f^{*}\left(x_{i}\right)=x_{i}^{*}\right)=\begin{cases} \frac{p}{e^{\epsilon}} & \text{if } x_{i}^{*}\in\left[x_{\min}^{*},\mathcal{L}\left(x_{i}\right)\right)\\ p & \text{if } x_{i}^{*}\in\left[\mathcal{L}\left(x_{i}\right),\mathcal{R}\left(x_{i}\right)\right]\\ \frac{p}{e^{\epsilon}} & \text{if } x_{i}^{*}\in\left(\mathcal{R}\left(x_{i}\right),x_{\max}^{*}\right] \end{cases}$$
(7)

where $x_i^* \in [x_{\min}^*, x_{\max}^*] \forall x_i \in DS^*$, with DS^* being the privatized dataset $\{x_1^*, \ldots, x_n^*\}$, and

$$(x_{\min}^*, x_{\max}^*) = (\bar{H} - C + \bar{A}, \bar{H} + C + \bar{A}),$$
 (8)

$$p = \left(e^{\epsilon} - e^{\epsilon/2}\right) / \left(2h\left(e^{\epsilon/2} + 1\right)\right),\tag{9}$$

$$C = h \cdot \left(e^{\epsilon/2} + 1\right) / \left(e^{\epsilon/2} - 1\right).$$
(10)

The elements that cause the PDF to vary depending on x_i are

$$\mathcal{L}(x_i) = \frac{\mathcal{C} + \mathcal{h}}{2} \left(\frac{x_i - \bar{\mathcal{H}}}{\mathcal{h}} \right) - \frac{\mathcal{C} - \mathcal{h}}{2} + \bar{\mathcal{H}} + \bar{\mathcal{A}}, \qquad (11)$$

$$\mathbf{R}(x_i) = \mathbf{L}(x_i) + \mathbf{C} - \mathbf{h}.$$
 (12)

 f^* is ϵ -locally differentially private, since the original proof in [2] holds. \overline{A} introduces a bias to the output, as per Thm. III.1, while \overline{H} and h affect the variance according to Thm. III.2.

Theorem III.1. Given $x_i^* = f^*(x_i, \bar{A})$, its expected value is

$$\mathbb{E}\left[x_{i}^{*}\right] = x_{i} + \bar{\mathcal{A}}.\tag{13}$$

Proof.

$$\mathbb{E}\left[x_{i}^{*}\right] = \int_{x_{\min}^{*}}^{\mathcal{L}(x_{i})} \frac{p}{e^{\epsilon}} x \, dx + \int_{\mathcal{L}(x_{i})}^{\mathcal{R}(x_{i})} px \, dx + \int_{\mathcal{R}(x_{i})}^{x_{\max}^{*}} \frac{p}{e^{\epsilon}} x \, dx$$
$$= \frac{p\left(1 - e^{\epsilon}\right)}{2e^{\epsilon}} \left[\mathcal{L}^{2} - \mathcal{R}^{2}\right] + \frac{p}{2e^{\epsilon}} \left[4\left(\bar{\mathcal{H}} + \bar{\mathcal{A}}\right)C\right] = x_{i} + \bar{\mathcal{A}}$$

Theorem III.2. Given $x_i^* = f^*(x_i, \overline{A})$, its variance is

$$\operatorname{VAR}(x_i^*) = h^2 \cdot \left(\frac{\left(\frac{x_i - H}{h}\right)^2}{e^{\epsilon/2} - 1} + \frac{e^{\epsilon/2} + 3}{3(e^{\epsilon/2} - 1)^2} \right).$$
(14)

Since we introduce the same bias \overline{A} to all samples of DS^* , we propose to remove it by using an altered version of the standard average computation, defined as

AVG^{*} (DS^{*},
$$\bar{A}$$
) = $\frac{1}{n} \sum_{i=1}^{n} x_i^* - \bar{A}$. (15)

Probability bounds of the error on the average: The error Δ_{AVG} between the calculated average from the privatized database and the original one is defined as

$$\Delta_{AVG} = AVG^* (DS^*) - AVG (DS).$$
(16)

Since DS^* is the output of a stochastic process with variance as per Eq. (14), it is possible to formulate a probabilistic bound on $|\Delta_{AVG}|$ according to Thm. III.3.

Theorem III.3. Given a dataset $DS = \{x_1, \ldots, x_n\}$ and a utility error $\lambda \ge 0$, an upper bound on the probability of the absolute value of the error being greater than λ is

$$P\left(\left|\Delta_{\text{AVG}}\right| \ge \lambda\right) \le e^{-\frac{\frac{1}{2}(n\lambda)^2}{\sum_{i=1}^n \text{VAR}(x_i^*) + \frac{1}{3}(C+h)n\lambda}}.$$
 (17)

Proof. We can use the independent and unbiased random variable $V_i = x_i^* - \bar{A} - x_i$ to write

$$P\left(\left|\Delta_{\text{AVG}}\right| \ge \lambda\right) = P\left(\left|\sum_{i=1}^{n} V_{i}\right| \ge n \cdot \lambda\right).$$
 (18)

We can reach the formulation in Eq. (17) using the Bernstein inequality together with the output bounds in Eq. (8), the sampling expected value in Eq. (13) and variance Eq. (14).

Since we are more interested in expressing the error on the average relatively to the original average, which is

$$\delta_{\text{AVG}} = \left(\text{AVG}^*\left(\text{DS}^*\right) - \text{AVG}\left(\text{DS}\right)\right) / \text{AVG}\left(\text{DS}\right), \quad (19)$$

we can adapt the bound in Thm. III.3 to have a relative formulation using δ_{AVG} , as per Thm. III.4.

Theorem III.4. Given a dataset $DS = \{x_1, \ldots, x_n\}$, $S_{DS} = \sum_{i=1}^n x_i$ and a utility error $\lambda \ge 0$, an upper bound on the probability of the absolute value of the relative error being greater than λ is

$$P\left(\left|\delta_{\text{AVG}}\right| \ge \lambda\right) \le e^{-\frac{\frac{1}{2}\left(nS_{\text{DS}}\lambda\right)^2}{\sum_{i=1}^n \text{VAR}\left(t_i^*\right) + \frac{1}{3}\left(C+E\right)nS_{\text{DS}}\lambda}}.$$
(20)

Remark 1. Since neither Eq. (17) nor Eq. (20) depend on \overline{A} , both error bounds are \overline{A} invariant. In Section V we empirically show that errors are \overline{A} invariant as well, meaning that selecting different values of \overline{A} result in similar $|\Delta_{AVG}|$ and similar $|\delta_{AVG}|$. However, this is true only when assuming infinitely precise numbers. In Subsection III-B we detail the reasons and show possible consequences of finite precision.

B. Selection of addition transform parameter

The \overline{A} parameter allows us to manipulate the binary representation of the private dataset so that some bits are shared by all x_i^* . An effective selection of \overline{A} fulfilling the recommendations in [3] and the conditions in Subsection II-C, while guaranteeing that all sign and exponent bits are shared, is presented in Thm. III.5.

Theorem III.5. Given a dataset DS, we define the unbiased exponent E_U^{enc} of the smallest region of numbers with equal exponents that can enclose the privatized dataset DS^{*} as

$$E_U^{\text{enc}} = \left\lceil \log_2(2\mathbf{C}) \right\rceil.$$
 (21)

By selecting $E_U^* \ge E_U^{enc} > -1022 \in \mathbb{Z}$, we compute \overline{A} as

$$\bar{\mathbf{A}} = 2^{E_U^* + 1} - 2 \cdot \text{ULP}\left(2^{E_U^*}\right) - \bar{\mathbf{H}} - \mathbf{C}.$$
 (22)

This formulation ensures that all sign and exponent bits are shared $\forall x_i^* \in DS^*$. The larger the selected E_U^* , the more mantissa bits will be shared as well.

The selection of \overline{A} should also take into consideration the loss due to the floating point finite precision, since the expected value in Eq. (13), and the error bounds in Eq. (17) and Eq. (20) hold only assuming infinite precision. However, the error with finitely precise numbers is significant only when



Fig. 3. Effects of large \overline{A} on expected value, δ_{AVG} and PDF, with \mathcal{F} as an estimation. x_i^* in the bottom plot are unbiased to be able to compare them.

A is computed according to Eq. (22) with $E_U^* \gg E_U^{\text{enc}}$, potentially resulting in unacceptable alterations to Zeal.

We examine the effects of large A on Zeal by considering a uniformly distributed DS with n = 1000, $\overline{H} = 1000$, h = 100and the privatization f^* (DS, \overline{A}) for various values of \overline{A} , as per Fig. 3. We first investigate the expected value of $x_1^* = f^*(x_1, \overline{A})$ with $x_1 = 1000$ by plotting $\delta_{\mathbb{E}(x_1)}$, defined as

$$\delta_{\mathbb{E}(x_i)} = \left(\mathbb{E}(x_i^*) - \bar{\mathbf{A}} - x_i\right) / x_i, \tag{23}$$

where $\mathbb{E}(x_1^*)$ is estimated by averaging 10^5 samples of x_1^* . For this dataset, when $\bar{A} \ge 10^{18}$ (red area), $\delta_{\mathbb{E}(x_1)}$ starts to deviate from the ideal 0% we would have with infinite precision and infinite samples, clipping to 100% for $\bar{A} \ge 10^{20}$ since

$$f^*(x_1) = \bar{A} \implies \mathbb{E}(x_1^*) = \bar{A}.$$
 (24)

When Eq. (24) is true $\forall x_i \in DS$, there is

AVG^{*}
$$(x_1^*, \dots, x_n^*) = 0.0,$$
 (25)

which causes the $|\delta_{AVG}|$ to clip at 100% as well. It should be noted that $|\delta_{AVG}|$ could potentially be greater than 100% for some \bar{A} , since this approximation effects drastically change f^* PDF as well, as we can see in the plot at the bottom of Fig. 3. For $\bar{A} \ge 10^{20}$, $(x_{\min}^*, L(x_1), R(x_1), x_{\max}^*) - \bar{A} = (0, 0, 0, 0)$. In order to estimate both $\delta_{\mathbb{E}(x_1)}$ and δ_{AVG} for large \bar{A}

In order to estimate both $\delta_{\mathbb{E}(x_1)}$ and δ_{AVG} for large A without knowing DS but only h and \overline{H} we introduce \mathcal{F} , computed as the average of the approximation error on the values $\overline{H} - C$ and $\overline{H} + C$ due to \overline{A} . \mathcal{F} is defined as

$$\mathcal{F} = \left[\left(\Delta x_{\min}^* / \left(\bar{\mathrm{H}} - \mathrm{C} \right) \right) + \left(\Delta x_{\max}^* / \left(\bar{\mathrm{H}} + \mathrm{C} \right) \right) \right] / 2, \quad (26)$$

where $\Delta x_{\min}^* = \overline{H} - C - (x_{\min}^* - \overline{A})$ and $\Delta x_{\max}^* = \overline{H} + C - (x_{\max}^* - \overline{A})$. In Fig. 3 we plot \mathcal{F} and notice that it is pretty accurate in describing when \overline{A} is too large to maintain acceptable error. Therefore, we can use it to select an appropriate \overline{A} .

C. Transmission savings

As discussed in Subsection III-B, the selection of \overline{A} depends only on \overline{H} , h and ϵ , thus \overline{A} can be fixed even before the sensors are deployed. Assuming that \overline{A} is chosen as per Eq. (22), we can guarantee that a minimum number of bits per sample γ_{\min} , computed according to Thm. III.6, is shared by all x_i^* and whose binary value is known a priori. Therefore, if each one of the *n* sensors transmits only $64 - \gamma_{\min}$ bits per sample, the database will still be able to reconstruct DS^{*}. We measure the savings in communication costs with this strategy by defining the *transmission ratio* (TR) as

$$TR(DS) = \frac{DS \text{ size in bits} - n \cdot \gamma_{\min}}{DS \text{ size in bits}} = 1 - \frac{\gamma}{64}.$$
 (27)

Theorem III.6. Given a privatized dataset $DS^* = f^*(DS, \overline{A})$ with \overline{A} computed according to Eq. (22), the minimum number of guaranteed bits shared per sample in DS^* is

$$\gamma_{\min} = 1 + 11 + E_U^* - \left[\log_2(2 \cdot C + 3 \cdot ULP\left(2^{E_U^*}\right) \right].$$
 (28)

Proof. Computing \overline{A} as per Eq. (22) ensures that

$$x_i^* \in \left[2^{E_U^*+1} - 2 \cdot \mathbf{C} - 3 \cdot \mathrm{ULP}\left(2^{E_U^*}\right), 2^{E_U^*+1}\right) \, \forall x_i^*.$$
 (29)

To represent all floating point numbers in the interval, we need the number m of changing mantissa bits to be such that

$$2^{m} \cdot \mathrm{ULP}\left(2^{E_{U}^{*}}\right) \geq 2 \cdot \mathrm{C} + 3 \cdot \mathrm{ULP}\left(2^{E_{U}^{*}}\right), \quad (30)$$

from which we can find m_{\min} . The expression in Thm. III.6 follows by summing the count of guaranteed shared mantissa bits, namely $52 - m_{\min}$, with 1 for the single sign bit and 11 for the exponent bits, that are all shared due to Eq. (22).

IV. FLOATING POINT VULNERABILITY

In differential privacy for numerical datasets, perturbing data generally involves sampling a random variable with specific PDF. When the datatype is floating point, this step should be performed with particular caution, since its arithmetic rules might lead to privacy leaks, as discussed in [1] and [4].

One of the most common methods to sample a PDF is to use its CDF^{-1} , namely the inverse of its cumulative distribution function. CDF^{-1} has the cumulative probability as input, with values $p_C \in [0.0, 1.0)$, and $x_i^* \in [x_{\min}^*, x_{\max}^*]$ as output: it can be proved that by sampling a uniformly distributed random variable from 0.0 to 1.0, and then applying the CDF^{-1} on the sample, the output is distributed as the desired PDF [11]. Given x_i , the vulnerability arises from the fact that the set of possible x_i^* if finite, meaning that some floating point number in $[x_{\min}^*, x_{\max}^*]$ might be unreachable output of $f^*(x_i, \bar{A})$. As depicted in Fig. 4, if the reachability of any $x_i^* \in [x_{\min}^*, x_{\max}^*]$ depends on x_i , the mechanism is not locally differentially private as per Def II.1.

Zeal solves the vulnerability by guaranteeing no unreachable x_i^* , since the selection of \overline{A} according to Thm. IV.1 implies that each x_i^* is the output of at least one p_C , as per the second row in Fig. 4.

Theorem IV.1. A preventing the vulnerability described in Section IV from causing privacy leaks are computed according to Eq. (22) with $E_U^* \ge E_U^{\text{vul}}$, where

$$E_U^{\text{vul}} = \max\left(\left\lceil -1 + \log_2\left(e^{\epsilon}/p\right)\right\rceil, E_U^{\text{enc}}\right). \tag{31}$$

Proof. In order to ensure that every $x_i^* \in [x_{\min}^*, x_{\max}^*]$ has a corresponding p_C pointing to it via CDF^{-1} , any interval $CDF^{-1}(ULP(p_C))$ wide should always contain at least one floating point x_i^* . Since CDF^{-1} is composed of lines with angular coefficient $a_i \in \{\frac{e^c}{p}, \frac{1}{p}\}$, we need

$$\mathrm{ULP}\left(p_{C}\right) \cdot a_{i} \leq \mathrm{ULP}\left(x_{i}^{*}\right),\tag{32}$$

where $a_i = \frac{e^{\epsilon}}{p}$ and $p_C \in [0.5, 1.0)$ are the worst scenario, since steeper lines and bigger ULP (p_C) lead to more x_i^* being unreachable. Under these conditions, Eq. (32) becomes

$$2^{E_U^* - 52} \cdot 2^{53} \ge (e^{\epsilon}/p) \implies E_U^* \ge -1 + \log_2(e^{\epsilon}/p).$$
(33)

The Eq. (31) comes from combining Eq. (33) with $E_U^* \ge E_U^{\text{enc}}$, since we need ULP (x_i^*) to be constant $\forall x_i^* \in \text{DS}^*$. \Box

V. RESULTS

In this section we present the results of Zeal on the first 5000 elements of the dataset "aarhus-citylab-humidity" [12] and the first 1000 of "chicago-taxi-trips-fares" [13]. Given their extrema, we assume that the former has feasible values $x_i \in [23.5, 83.9]$, and the latter $x_i \in [1.0, 120.0]$. To measure compression, we use the compression ratio (CR), defined as

$$CR(DS) = \frac{Compressed DS \text{ size in bits}}{Uncompressed DS \text{ size in bits}}.$$
 (34)

The compressor used for these analyses is *Greedy-GD* [14], which is based on bits deduplication. As reported in [3], *Greedy-GD* benefits from an increased number of shared bits.

Utility error on the average: In Fig. 5 we compare the probabilistic upper bounds on the utility error, both in absolute (Δ_{AVG}) and in relative terms (δ_{AVG}) , with the probability based on samples averaged over 10 iterations of *Zeal*. The two are comparable, and both show that the probability decreases as λ increases. Moreover, for sufficiently large λ values both probabilities are guaranteed to be zero: under ϵ -local differential privacy, this can not be achieved by perturbators with unbounded outputs, like the Laplace mechanism.

In order to estimate the impact of ϵ and \bar{A} on Zeal's utility error, we compute δ_{AVG} over 100 iterations for multiple ϵ and \bar{A} , as shown in Fig. 6. We see that with smaller ϵ the perturbator has to introduce more noise to privatize the data, increasing the utility error, and that similar ϵ result in different errors depending on the dataset. Moreover, as per Remark 1, different \bar{A} values result in similar utility errors.

Transmission savings and compressibility: In Fig. 7, we analyze transmission ratio (TR) and compression ratio (CR) against δ_{AVG} for a range of \overline{A} , with $\epsilon = 1.0$. TR and CR improve with very similar rates by increasing \overline{A} , since lower TR means less data to transmit from the sensors to the DB, and lower CR means better compressed privatized dataset.



Fig. 4. Floating point vulnerability for a DS with $\overline{H} = 10$, h = 5 and $\epsilon = 1$, with and without \overline{A} . When $\overline{A} = 0.0$, the reachable x_i^* could change depending on x_i . With a large enough \overline{A} , all x^* are reachable, causing no differentiating results that would lead to a privacy leak.



Fig. 5. Probability bound on the error of the average compared to sampling.



Fig. 7. Transmission savings and compressibility against δ_{AVG} . Any \bar{A} in the green area achieves improvements with comparable error levels.

For $\bar{A} < 10^{18}$ (end of green area), the relative utility error $|\delta_{AVG}|$ remains nearly the same as for $\bar{A} = 0$. Using the largest feasible \bar{A} in the green area, we reach up to a 94% improvement in CR, and up to a 95% improvement in TR.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced Zeal, a privatization technique composed of a perturbator and an aggregator for measuring the average of a dataset, which aims at reducing transmission data size and data storage, as well as solving a privacy vulnerability related to floating point data. We presented its characteristics, proved the necessary conditions for solving the vulnerability and showed its capabilities on a real-life dataset. We plan to improve Zeal so to process datasets with multiple dimensions, both in terms of attributes and as time-series, and expand the core ideas to other privatization mechanisms based on local differential privacy and to metrics other than the average.

REFERENCES

- I. Mironov, "On significance of the least significant bits for differential privacy," in CCS, 2012. [Online]. Available: tinyurl.com/5n7r5jsc
- [2] N. Wang *et al.*, "Collecting and analyzing multidimensional data with local differential privacy," in *IEEE ICDE*, 2019. [Online]. Available: tinyurl.com/bddsmued
- [3] F. Taurone *et al.*, "Change a bit to save bytes: Compression for floating point time-series data," in *IEEE ICC*, 2023. [Online]. Available: arxiv.org/abs/2303.04478
- [4] S. Haney *et al.*, "Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers," *ICML*, 2022. [Online]. Available: arxiv.org/abs/2207.13793
- [5] Google, "Secure noise generation," tinyurl.com/3z6fdn69, 2020.
- [6] S. Papadimitriou *et al.*, "Time series compressibility and privacy," in VLDB 2007.
- [7] D. B. Percival *et al.*, *Wavelet methods for time series analysis*. Cambridge university press, 2000, vol. 4.
- [8] R. Bassily *et al.*, "Local, private, efficient protocols for succinct histograms," in *ACM STOC*, 2015.
- [9] IEEE 754-2019 Standard for Floating-Point Arithmetic, Std., 2019.
- [10] C. Dwork et al., "The algorithmic foundations of differential privacy," Foundations and Trends® in Theoretical Computer Science, 2014.
- [11] R. C. Larson et al., Urban operations research, 1981, ch. 7.1.3.
- [12] Aarhus Kommune, "Sensordata," tinyurl.com/heeth2fd, 2017.
- [13] City of Chicago, "Taxi trips dataset," tinyurl.com/4rypurjp.
- [14] A. Hurst et al., "GreedyGD : Enhanced generalized deduplication for direct analytics in IoT," 2023. [Online]. Available: arxiv.org/abs/2304. 07240