Differentially Private Publication of Database Streams via Hybrid Video Coding

Javier Parra-Arnau, Thorsten Strufe and Josep Domingo-Ferrer

Abstract—While most anonymization technology available today is designed for static and small data, the current picture is of massive volumes of dynamic data arriving at unprecedented velocities. From the standpoint of anonymization, the most challenging type of dynamic data are data streams. However, while the majority of proposals deal with publishing either count-based or aggregated statistics about the underlying stream, little attention has been paid to the problem of continuously publishing the stream *itself* with differential privacy guarantees. In this work, we propose an anonymization method that can publish multiple numerical-attribute, finite microdata streams with high protection as well as high utility, the latter aspect measured as data distortion, delay and record reordering. Our method, which relies on the well-known differential pulse-code modulation scheme, adapts techniques originally intended for hybrid video encoding, to favor and leverage dependencies among the blocks of the original stream and thereby reduce data distortion. The proposed solution is assessed experimentally on two of the largest datasets in the scientific community working in data anonymization. Our extensive empirical evaluation shows the trade-off among privacy protection, data distortion, delay and record reordering, and demonstrates the suitability of adapting video-compression techniques to anonymize database streams.

Index Terms—Database anonymization, data streams, privacy, video encoding.

1 Introduction

Much of what we touch and work with today automatically generates data that someone is disposed to collect and analyze. The availability of massive amounts of such data —frequently at the individual level— play a fundamental role in the extraction of knowledge and decision-making in contexts as varied as business competitiveness, marketing, social relationships, transportation, health and wellbeing, education and politics [1].

Despite the economic and societal good that comes from big-data research, raising tensions exist with the perceived risks to individuals' privacy [2], [3], [4]. To deal with these tensions, current legal frameworks in Europe and other regions limit the collection, processing and sharing of personally identifiable information (PII). Basically, the controllers of PII have a series of obligations towards the individuals to whom the PII corresponds, which include, among others, seeking their consent, guaranteeing them rights to access, rectification and erasure.

The advent of big data, together with the development of data science in general and machine learning in particular, has raised the question of how to leverage those PII-data for secondary purposes (i.e., other than the purpose at collection time), since complying with the above-mentioned legal obligations is extremely difficult in a scenario where a bunch of controllers may exchange and fuse data. It is precisely in this situation where *anonymization* comes into the picture, as the tool that legitimately allows circumventing the legal restrictions applicable to those data.

Differential privacy (DP) [5] is one of the most prominent privacy notions in the field of anonymization. In the interactive setting, the assumption is that an anonymization mechanism sits between an analyst submitting queries and the database[1] answering them. In the non-interactive scenario, on the other hand, a protected version of the original database is generated and released, which allows any entity (not necessarily the data analyst in question) to perform *any* analyses on the protected data, and permits using such data, possibly in combination with other information, for secondary purposes.

The assumption in most of the current anonymization technology, however, is that the original database does not change over time and there is no need to publish it more than once [6]. Nonetheless, in the current context where colossal amounts of data are generated every single day [7], this is by no means realistic.

Our work tackles the problem of anonymizing *dynamic* databases with DP guarantees. We focus on the most challenging case, *data streams*, where only new records and record updates are published at certain release times, data freshness is critical, and the order in which the protected data are released matters. For this type of data, the vast majority of proposals deal with publishing either count-based or aggregated statistics about the underlying dynamic data (e.g., [8], [9]). To the best of our knowledge, only [10]

Javier Parra-Arnau is both with the Universitat Politècnica de Catalunya (UPC), Dept. Network Engineering, 08034, Barcelona, Spain, and with the Karlsruhe Institute of Technology (KIT), Institut für Telematik - Praktische IT-Sicherheit, 76131, Karlsruhe, Germany. Thorsten Strufe is with the KIT, Institut für Telematik - Praktische IT-Sicherheit, 76131, Karlsruhe, Germany. Josep Domingo-Ferrer is with the Universitat Rovira i Virgili, Department of Computer Science and Mathematics, UNESCO Chair in Data Privacy, 43007 Tarragona, Catalonia.
E-mail: javier.parra@upc.edu, javier.parra-arnau@kit.edu, strufe@kit.edu, josep.domingo@urv.cat
Manuscript prepared May, 2021.

[1] Throughout this work, we shall use the terms data set and database interchangeably.

has studied the publication of the database *itself* (rather than statistics derived from it) in a context of data stream. Nonetheless, that work is intended only for data sets with a single attribute and does not contemplate record updates, which renders the anonymization scheme useless for practical stream-data based systems.

1.1 Contribution and Plan of this Paper

The main contribution of this paper is an anonymization method that can publish multiple numerical-attribute, finite database streams with DP guarantees through hybrid video encoding techniques. The proposed method relies on the signal compression scheme *differential pulse-code modulation* (DPCM), and is optimized in a number of different ways to allow record updates and to provide high-privacy protection and high-utility guarantees in terms of data distortion, delay and record reordering.

Our solution operates with blocks of records, which are input into a closed loop consisting of several modules: preprocessing, analysis-synthesis, quantization, prediction and encoder control. On the one hand, the preprocessing, prediction and encoder control modules work jointly to select a permutation of the records of the block and a configuration of the prediction module that minimize the error in predicting the block; the prediction module can be configured to both leverage statistical dependencies inside frames (i.e., groups of blocks protected together) and exploit dependencies among the different frames of the database stream. On the other hand, the analysis, synthesis and quantization modules operate jointly to choose the transform coding scheme and the number of transform coefficients that will be protected in order to minimize the mean squared error (MSE) incurred in releasing the synthetized, protected block (instead of the original one).

The proposed solution is evaluated experimentally on two real data sets, "(Very) Large Census" and "Quant Forest", which are two of the largest datasets in the community of statistical disclosure control. A variety of empirical results shows the trade-off among privacy protection, data distortion, delay and record reordering, and demonstrates the suitability of our approach.

The remainder of this paper is organized as follows. Sec. 2 establishes some preliminaries and reviews the state of art relevant to this work. Sec. 3 formally states the problem tackled in this paper. Sec. 4 describes our approach to generate DP database streams through hybrid video encoding techniques. Sec. 5 conducts an experimental evaluation of the proposed anonymization method. Sec. 6 discusses previous work on differentially-private transform coding. Finally, conclusions are drawn in Sec. 7.

2 Preliminaries

2.1 Differential Privacy

DP was originally proposed as a privacy model in a interactive setting to protect the outcomes of queries to a database. In this setting, the assumption is that an anonymization mechanism sits between a user submitting queries and a (trusted) database curator answering them.

Our work focuses on a *non-interactive* setting, where the curator releases a protected version of the database, allowing the user to perform hopefully any analysis on the data without further interacting with the curator.

Central to DP is the notion of *neighbor* databases, which can be interpreted in two different ways. On the one hand, the *unbounded* case assumes one entry is either removed or added. On the other hand, the *bounded* notion considers the replacement of one record by another. An important difference is that the former case assumes the size n of the database to be publicly known, whereas the latter assumes this parameter is private. Nonetheless, the two notions of neighborhood are very related and mechanisms satisfying one can be adapted to meet the other. For the sake of mathematical simplicity, we use the latter definition.

We shall consider *central* DP[2], as defined below.

Definition 1 (L1-sensitivity [5])**.** *Let $\mathcal{D}$ be the class of possible data sets. The global sensitivity or L1-sensitivity of a query function $f\colon \mathcal{D} \to \mathbb{R}^d$ is defined as*

$$\mathrm{GS}(f) = \max_{\mathbf{x},\mathbf{x}' \in \mathcal{D}} \|f(\mathbf{x}) - f(\mathbf{x}')\|_1,$$

where $\mathbf{x}, \mathbf{x}'$ are any two neighbor databases in the sense described above.

Definition 2 (ε-Differential privacy [5])**.** *A randomized mechanism $\mathcal{M}$ on a query function f satisfies ε-differential privacy with $\varepsilon \geqslant 0$ if, for all pairs of neighbor databases $\mathbf{x}, \mathbf{x}'$ and for all $\mathcal{O} \subseteq$ range($\mathcal{M}$),*

$$\frac{\mathrm{P}\{\mathcal{M}(f(\mathbf{x})) \in \mathcal{O}\}}{\mathrm{P}\{\mathcal{M}(f(\mathbf{x}')) \in \mathcal{O}\}} \leqslant \exp(\varepsilon).$$

2.2 Related Work

In this subsection, we review the state of the art relevant to this work. We first examine the classical approaches to anonymize static data sets, and secondly analyze those proposals aimed to protect dynamic data. In both cases, the privacy model assumed is DP.

2.2.1 Histograms versus Record Masking

Even if DP was initially proposed to limit disclosure risk in database queries, mechanisms to generate DP data sets (i.e., the so-called non-interactive setting) appeared soon after its inception. Nonetheless, except for the simplest data domains, publishing useful DP data sets (i.e., data sets that well approximate the original ones) remains a highly challenging task.

There exist two main approaches to generate DP data sets: *histograms* and *record masking*. In the former case, given an original data set $\mathbf{x}$, we generate a histogram h through a suitable partitioning of the data domain. From this point on, we discard $\mathbf{x}$ and the target of protection is h. Hence, the goal is to publish h^{ε}, an ε-DP version of h. In the latter case, the aim is to generate $\mathbf{x}^{\varepsilon}$, an ε-DP version of $\mathbf{x}$, that is, an anonymized version of the data in the *original format*.

The histogram approach takes advantage of the low sensitivity of counting queries over a partition of the data domain [11]. The naive application of this mechanism, however, becomes problematic as the complexity of the data

[2] It is also called as user-level DP in data streaming applications.

domain increases. Note that, for a fixed accuracy, the cardinality of the partition (number of bins) grows exponentially with the number of attributes, which may have important effects on the computational cost and the accuracy of the protected data.

Some mitigation strategies have been proposed to tackle the issues caused by data dimensionality. In [12], given a partition, the authors propose an algorithm that minimizes the error for a given family of counting queries. In [13], data summarization techniques are utilized to reduce the time and space complexity, by making time and space proportional to the number of non-empty cells in the summarized data set. An alternative way to deal with those issues is to apply dimensionality-reduction techniques. This is the strategy followed by [14], which models the dependency between attributes to generate the DP data set from a set of low-order marginals.

The alternative to generate DP data sets based on *record masking* avoids partitioning the data domain. Instead, the data set is protected by masking the original records. However, masking each record by adding a Laplace-distributed noise with magnitude proportional to the record sensitivity is not a feasible solution. Since the purpose of DP is to hide the presence of any single record, such a naive approach inescapably needs to introduce too much noise, thereby producing significant utility damage.

As a result, a wide body of research has investigated how to reduce the sensitivity of the queries used to generate the DP data sets. A few examples include [15], [16], [17], where microaggregation [18] is utilized with that purpose. In the cited works, rather than querying each original record, only the representatives of the microaggregation clusters are queried. Since a cluster representative is an aggregation of the records in the cluster, intuitively its global sensitivity is smaller than that of any single record. Clearly, the amount of sensitivity reduction depends on how such representative values are computed.

2.2.2 Differentially Private Publication of Dynamic Data

The aforementioned anonymization schemes assume that the original data set does not change over time and, therefore, that there is no need to publish them more than once. However, in the current context of big data, this seems not realistic.

Obviously, a straightforward application of the previous schemes to the scenario at hand would still be possible. Nonetheless, applying those methods *independently* at each release time, i.e., without considering correlations between consecutive releases or the dynamics of the data stream, may not be an appropriate approach.

Few recent works have tackled the problem of protecting dynamic data sets with DP guarantees. Essentially, the privacy research community has focused on two distinct scenarios. In the former scenario, all available data or a synopsis thereof (e.g., histograms) are anonymized periodically, although not necessarily at regular time instants. On the contrary, in the latter scenario, (i) data items are not republished in multiple versions, i.e., only new or updated data are protected at a given release time; (ii) time is critical, in the sense that a new or updated data item must be anonymized and published within a predefined, short time frame; and (iii) the order in which the protected data are released matters. Following the terminology of [19], we shall refer to these two scenarios respectively as *multiple release* and *data stream*.

Distinct technologies have been developed for each case. In the multiple-release scenario, [20] studies the problem of publishing histograms of dynamic data sets. Instead of generating a DP histogram at each release time, the cited work proposes computing only new histograms when the update is significant, that is, when a distance measure between the current histogram and the latest released histogram exceeds a threshold. The proposed strategy is independent of how histograms are computed at each release time, and the goal is to adjust the threshold adaptively based on data dynamics. The main problem of this proposal is that it suffers from all the limitations of the static histogram approach mentioned in Sec. 2.2.1.

Another proposal for multiple release is [21], which deals with the publication of histograms as well, but combines sampling [22] with clustering (i.e., time units with similar trends are grouped) to improve utility. The proposed solution, however, adopts an event-level DP approach [23], which protects the presence of an individual event, i.e., an individual's contribution to the data stream at a single time point, rather than their presence or contribution to the entire publication series (also known as user-level DP).

In the case of data stream, the vast majority of proposals focus on publishing either count-based or aggregated statistics. One of those works is [8], which aims to protect count series (e.g., the daily count of people diagnosed with HIV/AIDS) over individuals continuously. The proposed scheme provides user-level DP and assumes the series are generated by an underlying process from which predictions are made to enhance the accuracy of the released data. However, a statistical model of the process needs to be assumed or inferred from public data with similar patterns, and therefore the anonymization scheme may not be effective when the actual data deviate from it.

PeGaSus [24] is another proposal that aims to release continuous count-based statistics. Unlike [8], the notion of neighborhood between databases (and so DP) is modified here to suit streaming analytics but it is only intended to protect single-data events, analogously to event-level DP.

A more recent work is OptStream [9], which generates a sequence of protected data where each term represents a private version of the aggregated data (e.g., a count) up to a given time instant. The proposed solution relies on the w-event framework [25], which extends the definition of DP to protect stream analytics. However, like PeGaSus, it cannot be applied to release the database stream itself and, besides, the target of protection are not individuals' full contributions to the stream[3].

To the best of our knowledge, only [10] has studied the publication of the database itself (rather than statistics derived from it) in a context of data stream, which is the focus of this work. δ-DOCA, as the method is called, adopts a record-masking approach and provides central ε-DP, which means all contributions (and not only some

[3] w-event privacy does not protect event sequences occurring beyond a time window of size w.

consecutive pieces thereof) are protected. Nonetheless, it is intended only for data sets with a single attribute and does not contemplate record updates.

3 Problem Statement

We shall follow the convention of using uppercase letters for random variables (r.v.'s), lowercase letters for the particular values they take on, and bold letters for matrices. Probability density functions (PDFs) and probability mass functions (PMFs) are denoted by p and subindexed by the corresponding r.v. We adopt the same notation for vectors in [26] and use parentheses to construct column vectors from comma-separated lists.

We study the protection of *database streams*[4] with central DP guarantees, which means there is a trusted entity (i.e., the curator) that gathers data continuously from a population and takes charge of protecting them from the outside world.

There are multiple ways to define DP in such a data streaming setting, e.g., at the granularity of attributes [27], events [23], windows of events [25], records or individuals. This work assumes the required protection is at the *individual* level (also known as user-level DP), that is, the curator aims to protect *all* tuples or records corresponding to any individual in the stream database.

Mathematically, we model database streams as discrete time vector processes. An *original* database stream $\{S_i\}$ is defined, accordingly, as a sequence of continuously incoming tuples $S_i = (I_i, A_{i1}, \ldots, A_{id})$, where I_i is an r.v. denoting the identity of the subject to whom S_i corresponds, and $A_{i1}, \ldots, A_{id}$ are r.v.'s representing d attributes of that subject. Throughout this work, we shall also refer to the tuples of a database stream as *records*.

In general, the protection of a stream requires some sort of distortion (e.g., Laplace-noise addition) of the original attribute values, and therefore implies inevitably some information loss. We denote by $\{S_i^\varepsilon\}$ an ε-DP version of the original database stream $\{S_i\}$, that is, a sequence of continuously output tuples $S_i^\varepsilon = (A_{i1}^\varepsilon, \ldots, A_{id}^\varepsilon)$, where identities are removed and $A_{i1}^\varepsilon, \ldots, A_{id}^\varepsilon$ are suitably distorted versions of the attribute values in the original tuple corresponding to S_i^ε.

To quantify how well the distorted attribute values approximate the original ones, we shall use the sum of squared errors (SSE), a measure of distortion frequently employed in the evaluation of DP mechanisms.

The degree of distortion of the protected attribute values is one dimension of the information loss incurred by a protection method. The other dimensions are related to the fact that some, or all, of the records in the original stream may be delayed and reordered; obviously, any method for data streams must buffer incoming tuples before protecting them. Next, we slightly generalize the delay-constraint definition of [28].

Definition 3 (Delay constraint). *Let $\mathcal{M}$ be a protection mechanism that takes as input a database stream $\{S_i\}$ and outputs an ε-DP stream $\{S_i^\varepsilon\}$. For a positive integer δ, $\mathcal{M}$ is said to satisfy the* delay constraint δ *if, upon receiving any new tuple S_i, $\mathcal{M}$ has already output all the protected tuples corresponding to tuples in $\{S_i\}$ with position less than $i - \delta + 1$.*

While delay constraints are common in the context of data stream, to the best of our knowledge no attempt has been made to preserve the order of the incoming records. In [10], for example, tuples are reordered as much as needed to satisfy maximum attribute homogeneity for a given delay, ignoring the value of the information encoded in such order. To make our analysis as comprehensive as possible, we shall quantify the impact of such reordering through a *reordering cost function*.

Unlike [10], we also contemplate tuple updates, meaning there can be tuples arriving at different time instants that belong to a same subject but contain different attribute values. In this work, we require that such updates satisfy the following mild constraint.

Definition 4 (Tuple-update constraint). *Let $\{S_i\}$ be an original database stream and $T \subseteq \{S_i\}$ the sequence of all tuples corresponding to a given subject. For a positive integer α, the original stream satisfies the* tuple-update constraint α *if, for any subject and any two consecutive tuples of T, such two tuples differ at least in α positions in $\{S_i\}$.*

Informally, Definition 4 tells us that we should expect a lag between a tuple and its update, or between two consecutive updates. With a mild loss of generality, this work will assume $\alpha \geqslant \delta$. Since, by Definition 3, the maximum number of buffered tuples at any moment is δ, the tuple-update constraint ensures those two tuples (i.e., a tuple and its update, or two consecutive updates) will not coincide in the buffer. In real practice, however, if the condition $\alpha \geqslant \delta$ is not met, only the most recent tuple will be output.

A direct consequence of the fact that tuples can be updated is the *finite* length of the protected database stream. Since the level of protection ε is necessarily finite, by the sequential composition property of DP [29] the privacy budget will be consumed completely at some time instant. We shall denote by l the *target length* of the protected database stream, that is, the number of incoming records the database curator wishes to protect.

Given all such considerations, the problem tackled in this work is as follows. We aim to design a DP mechanism suitable for database streams that, for a given ε and l, achieves serviceable points of operation in the privacy-utility trade-off, being utility measured as distortion, delay and reordering.

4 Differentially Private Continuous Publication of Data Sets via Hybrid Video Encoding

This section describes our methodology to publish DP database streams through hybrid video encoding techniques.

In this work, we propose the masking of database streams at the *record level*, instead of at the histogram level. Doing so is computationally efficient, since the cost is linear with the number of records. However, plain independent masking of the records in the original database stream may degrade utility severally, as we describe next.

[4] For brevity, we shall refer occasionally to a database stream simply as "stream".

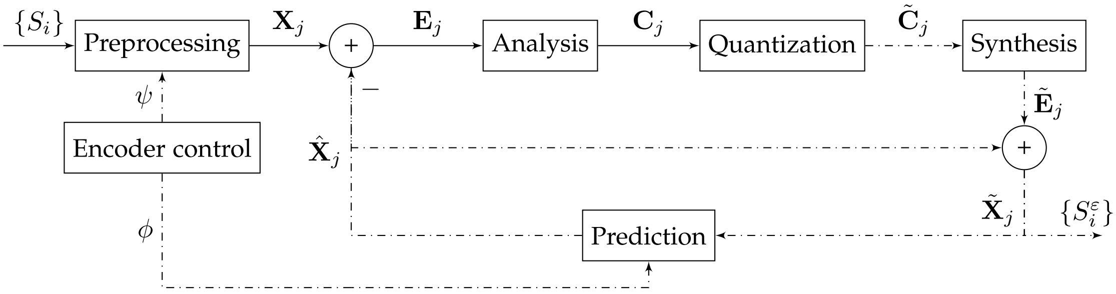


Fig. 1: Overview of the proposed scheme to generate DP database streams. Dashed and continuous lines indicate the data at those points are respectively protected and unprotected.

For a positive integer r, define the *identity function* $I_r(\{S_i\})$ as the function that returns the attribute values of the r-th element (i.e., record) of $\{S_i\}$. Since the whole process $\{S_i\}$ can be interpreted as the collected answers —except for subjects' identity— to the queries $I_r(\{S_i\})$ for all available elements, an intuitive way to generate the protected stream $S_1^\varepsilon, \ldots, S_l^\varepsilon$ with $\delta = 1$ is collecting an ε/l-DP response to each $I_r(\{S_i\})$ for $r = 1, \ldots, l$. Since we allow record updates, it follows from the sequential composition property that $S_1^\varepsilon, \ldots, S_l^\varepsilon$ also meets the desired ε-DP requirement. In short, with this methodology, the protected database stream is generated by providing a DP response to the queries asking for the values of all attributes in l records of the original sequence.

Although this record-level perturbation methodology does not make any assumptions on the uses of the output data, unfortunately it may come at the expense of a huge information loss. Throughout this paper, we shall assume each attribute j takes on values in the interval $[0, \Lambda_j]$, and denote by Λ the column vector $(\Lambda_1, \ldots, \Lambda_d)$. Since each query I_r refers to a *single* individual, its L1-sensitivity is as large as $\sum_{k=1}^{d} \Lambda_k$, which implies a huge distortion to attain ε-DP. The result is a database stream $S_1^\varepsilon, \ldots, S_l^\varepsilon$ with very limited utility.

To make record-level masking viable to generate DP data sets, there is an evident need to reduce the sensitivity of the query function/s to be used. In the following subsections, we shall describe a method that protects, at a time, groups of tuples conveniently sorted, and exploits statistical dependencies among releases.

4.1 Overview

We propose a protection method that relies on the DPCM, which is closely related to the concept of closed-loop predictive quantization. The basic structure of our method is illustrated in Fig. 1 and described succinctly next. Each individual module is analyzed in greater detail in the following subsections.

In our protection method, the database stream tuples $\{S_i\}$ are not directly processed, but buffered at a preprocessing module, where records are appropriately sorted. In particular, as soon as m records are available at this module, groups of $n < m$ consecutive records are removed from the buffer and input into the closed loop successively, i.e., one after another. We shall assume that $m = n\,b$ for some integer $b > 1$, and that groups are processed in order of their records. Following the terminology of numerous image and video compression formats, we shall refer to this processing unit as *block*. In analogy to video coding, the set of b of such blocks will be called a *frame*.

From a notational point of view, note that, while i indexes *individual* records within the original database stream, j indexes *blocks* of n records within the closed loop. On the other hand, since all modules inside the loop operate at the block level, for mathematical convenience we shall model such blocks as random matrices of dimension $n \times d$. Hence the notation of Fig. 1.

Essentially, each block $\mathbf{X}_j$ is predicted based on the previous protected blocks $\tilde{\mathbf{X}}_{j-1}, \tilde{\mathbf{X}}_{j-2}, \ldots, \tilde{\mathbf{X}}_{j-\pi}$, for some integer π. The prediction block $\hat{\mathbf{X}}_j$ is subtracted from the preprocessed input block $\mathbf{X}_j$, thereby yielding a prediction error $\mathbf{E}_j = \mathbf{X}_j - \hat{\mathbf{X}}_j$. The block $\mathbf{E}_j$ is then transformed, quantized and protected with ε_j-DP, respectively by the modules analysis and quantization. The synthesis module afterwards reverses the previous transformation and the upshot is a protected and reconstructed block $\tilde{\mathbf{E}}_j$ for the prediction error $\mathbf{E}_j$. Then, $\tilde{\mathbf{E}}_j$ is added to the predictor $\hat{\mathbf{X}}_j$, resulting in the reconstructed *output* block $\tilde{\mathbf{X}}_j$. Releasing $\tilde{\mathbf{X}}_j$ in a single batch yields n consecutive records of the protected database stream $S_1^\varepsilon, \ldots, S_l^\varepsilon$.

The fundamental principle upon which the above methodology relies is *difference quantization*. One simple but important result that follows from the fact that

$$\begin{aligned} \mathbf{E}_j &= \mathbf{X}_j - \hat{\mathbf{X}}_j, \\ \tilde{\mathbf{E}}_j &= \tilde{\mathbf{X}}_j - \hat{\mathbf{X}}_j, \end{aligned} \tag{1}$$

is that the overall MSE in releasing $\tilde{\mathbf{X}}_j$ instead of $\mathbf{X}_j$ is equal to the MSE incurred in quantizing $\mathbf{E}_j$. Formally,

$$\mathrm{E}\,\|\mathbf{X}_j - \tilde{\mathbf{X}}_j\|_{\mathrm{F}}^2 = \mathrm{E}\,\|\mathbf{E}_j - \tilde{\mathbf{E}}_j\|_{\mathrm{F}}^2, \tag{2}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

When $\hat{\mathbf{X}}_j$ in (1) is a prediction of $\mathbf{X}_j$ based on some information about the past of $\mathbf{X}_j$, Eq. (2) is called the fundamental theorem of *predictive quantization* [30]. Note, however, that (2) holds for any $\hat{\mathbf{X}}_j$ regardless of whether it is a prediction of $\mathbf{X}_j$ or not. When it is, in the context of image and video compression, algorithms can be more efficient. In our context of database stream, we shall show that privacy protection can be provided with less distortion, drawing an analogy between these two fields.

We have mentioned that the analysis module applies a transformation on the prediction error block $\mathbf{E}_j$. Although multiple transformations are possible, here we use the most popular one in image and video compression, the discrete cosine transform (DCT), as well as the discrete sine transform (DST) and the discrete Hartley transform (DHT). Apart from variety, the reason for our choice is as follows. They are all orthogonal, two-dimensional separable and data-independent, and they all exhibit high-energy compaction, meaning that information, after being transformed, tends to be concentrated in a few, low-frequency transform coefficients.

As we shall describe in Sec. 4.3, the quantization module will be in charge of selecting which coefficients are retained and perturbed with the Laplace mechanism, and which ones are removed. Regardless of the selection criterion, however, predicting $\mathbf{X}_j$ from the reconstructed (and protected) past has two immediate advantages. On the one hand, the variance of the error block $\mathbf{E}_j$ will in principle[5] be less than the variance of the original block $\mathbf{X}_j$, so that a reduced range of values will be transformed and protected. In image coding, predictive quantization (without transform coding) has the ability to increase the accuracy of the quantized values without increasing the number of coding bits. In our case (where we additionally consider transform coding), a smaller variance of the elements of $\mathbf{E}_j$ will intuitively translate into a smaller number of high-frequency transform coefficients. As a result, the same privacy budget ε_j will be distributed among less coefficients, thus yielding less distortion.

On the other hand, predicting $\mathbf{X}_j$ from reconstructed blocks has an evident advantage both in video coding and in database streams. In the former application, it allows both an encoder and decoder to generate the same block $\hat{\mathbf{X}}_j$ without transmitting any additional information from the former to the latter. In our case, due to the post-processing property [31] of DP, we shall generate each prediction block without consuming any privacy budget.

With the proposed method, we shall therefore be able to output ε_j-DP blocks. At the frame level, since $\alpha \geqslant \delta \geqslant m > n$, each of the blocks of a same frame will contain records belonging to different subjects[6]. The result is that each protected frame will also satisfy ε_j-DP by the parallel composition property of DP [31]. To meet the requirement of protecting l input records, it will suffice to set $\varepsilon_j = \varepsilon\, m/l$ for all j.

4.2 Transform Coding

The aim of transform coding is to apply an adequate linear transformation on each input block, so that the transform coefficients are much less correlated than the original samples and the information is more "compact" in the sense of being concentrated in only a few of the transform coefficients[7]. It is important to note that transform coding exploits only dependencies among the samples of a single block. For additionally utilizing dependencies among transform blocks and frames, intra-picture and inter-picture prediction techniques can be used.

Transform codes are popular because they provide an attractive compromise between computational complexity and performance. As mentioned in Sec. 4.1, we shall use, among others, the DCT, a data-independent transform that is employed in all practical video coding schemes. Although there are several DCTs, the DCT-II is probably the most commonly used form and is often simply referred to as "the DCT". In addition to the DCT, our scheme also incorporates the DST-I and the DHT.

For notational simplicity, in this subsection we shall drop the subindex j of the r.v.'s represented in Fig. 1. In addition, we shall assume realizations of these variables.

Let $\mathbf{a}^n = [a_{ij}^n]$ denote the $n \times n$ transformation matrix of any of the three transforms employed by the analysis and synthesis modules. In the case of the DCT, the entries of $\mathbf{a}^n$ are

$$a_{ij}^n = \begin{cases} \dfrac{1}{\sqrt{n}}, & \text{if } i = 1 \\ \sqrt{\dfrac{2}{n}} \cos\Big(\dfrac{\pi}{2n}\,(i-1)\,(2j-1)\Big), & \text{if } i > 1. \end{cases}$$

In the case of the DST and DHT, the entries of the corresponding matrices are respectively

$$a_{ij}^n = \sqrt{\frac{2}{n+1}} \sin\left(\frac{\pi}{n+1} ij\right)$$

and

$$a_{ij}^n = \sqrt{\frac{2}{n}} \cos\left(\frac{2\pi}{n}(i-1)(j-1) - \frac{\pi}{4}\right).$$

Recall [32] that, given a matrix $\mathbf{x}$ of dimensions $n \times d$, the forward and inverse transform of a separable, two-dimensional transformation is given respectively by

$$\mathbf{y} = \mathbf{a}^n\, \mathbf{x}\, \mathbf{a}^{d^{\mathrm{T}}}, \quad \mathbf{x} = \mathbf{a}^{n^{\mathrm{T}}}\, \mathbf{y}\, \mathbf{a}^d. \tag{3}$$

Our next result, Lemma 1, derives the global sensitivity of the transformed coefficients of a separable, two-dimensional transformation, when a prediction block is subtracted from an input block. The strength of this result lies in that it is not restricted to the transforms contemplated in this work.

Lemma 1 (Sensitivity of transform coefficients). *For any $i = 1, \ldots, n$, denote by $r^*(i)$ the index that maximizes $|a_{ir}^n|$. Let $\mathbf{x}$ be an observed block of $n \geqslant 2$ records and d attributes, $\hat{\mathbf{x}}$ a prediction block, and $\mathbf{e}$ the corresponding error. Denote by $f_{c_{ij}}$ the query function that returns the element (i,j) of the transform block $\mathbf{c} = \mathbf{a}^n\, \mathbf{e}\, \mathbf{a}^{d^{\mathrm{T}}}$. The L1-sensitivity of this function is*

$$\mathrm{GS}(f_{c_{ij}}) = |a_{i,r^*(i)}^n| \sum_{k=1}^{d} \Lambda_k |a_{jk}^d|.$$

Proof: Consider two neighbor input blocks $\mathbf{x}$ and $\mathbf{x}'$, and their corresponding transformed error blocks $\mathbf{c}$ and $\mathbf{c}'$. For any $r \in \{1, \ldots, n\}$, denote by $x_r = (x_{r1}, \ldots, x_{rd})$ and $x_r' = (x_{r1}', \ldots, x_{rd}')$ the respective values of the different record in either input block. Clearly, since $\hat{\mathbf{x}}$ does not depend on $\mathbf{x}$ or $\mathbf{x}'$, but on previous reconstructed blocks,

$$\mathbf{c} - \mathbf{c}' = \mathbf{a}^n\,(\mathbf{x} - \mathbf{x}')\,\mathbf{a}^{d^{\mathrm{T}}}.$$

[5] As long as the prediction is good enough.

[6] Said otherwise, the sets of subjects protected in those blocks will be non-overlapping.

[7] We emphasize that there is no general theoretical result that states that uncorrelated quantities can be more efficiently quantized than can correlated variables.

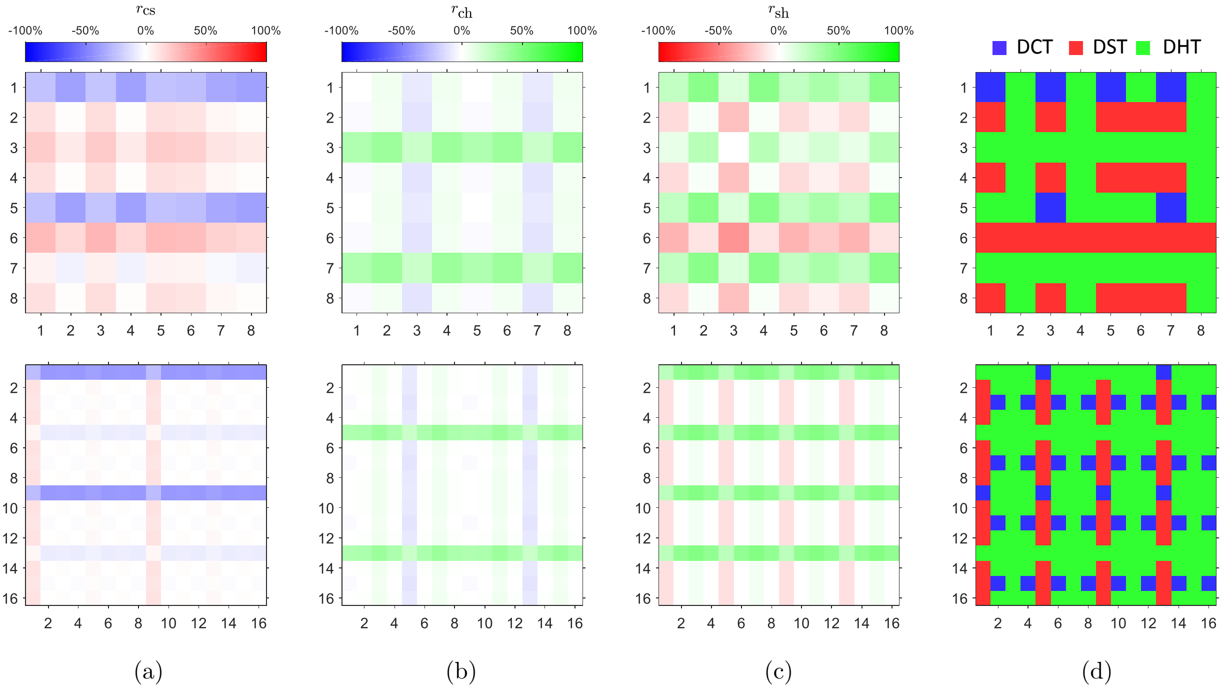


Fig. 2: (a-c) Relative difference in L1-sensitivity, as defined in (6), among the discrete cosine, sine and Hartley transforms for two block sizes, $n = d = 8$ (top-row figures) and $n = d = 16$ (bottom-row figures). (d) Transform with the minimum sensitivity value for each coefficient, for $n = d = 8$ (top figure) and $n = d = 16$ (bottom figure). The results have been computed for $\Lambda = \mathbf{1}$.

From (3), simple algebraic manipulation then shows

$$c_{ij} - c'_{ij} = \sum_{k=1}^{d} a^n_{ir}\, a^d_{jk}(x_{rk} - x'_{rk}). \qquad (4)$$

Accordingly,

$$\begin{aligned} \mathrm{GS}(f_{c_{ij}}) &= \max_{\mathbf{x},\mathbf{x}'} |c_{ij} - c'_{ij}| \\ &= \max_{x_r, x'_r} \left\{ |a^n_{ir}| \left| \sum_{k=1}^{d} a^d_{jk}(x_{rk} - x'_{rk}) \right| \right\} \\ &\overset{\text{(a)}}{=} \max_{r} |a^n_{ir}| \max_{x_r, x'_r} \left| \sum_{k=1}^{d} a^d_{jk}(x_{rk} - x'_{rk}) \right| \\ &\overset{\text{(b)}}{=} |a^n_{i,r^*(i)}| \sum_{k=1}^{d} \Lambda_k |a^d_{jk}|, \end{aligned}$$

where

(a) reflects that the maximization of $|a^n_{ri}|$ with respect to all x_r and x'_r depends just on the position index r; and

(b) holds with equality since the components of x_r and x'_r can be chosen so that all terms $a^d_{kj}(x_{rk} - x'_{rk})$ have the same sign. ■

An important conclusion that follows from Lemma 1 is that the sensitivity of any coefficient c_{ij} (regardless of the particular transformation used) depends on the sensitivity of each and every attribute, rather than on a single Λ_k. In other words, there is no a one-to-one correspondence between the sensitivity of the attribute value of a record in $\mathbf{x}$, and that of the transform coefficients, which in principle may limit the benefits of transform coding. Our next result, Corollary 1, shows that this limitation is, fortunately, compensated in part by an averaging effect of Λ. Before proceeding, we first prove an interesting property of the DCT transform matrix, used in the corollary.

Proposition 1 (Property of the DCT transformation matrix)**.** *For any $i = 2, \ldots, n$, any $j = 1, \ldots, d$, and any $n \geqslant 2$, $|a^n_{ij}| \geqslant a^n_{1j}$.*

Proof: Assume $2j - 1$ and $2n$ are mutually prime. By Bézout's identity, there exist then integers α and β such that

$$\alpha(2j - 1) + \beta 2n = 1. \qquad (5)$$

Note that, for an arbitrary integer k, $\alpha_k = \alpha + k2n$ and $\beta_k = \beta - k(2j-1)$ satisfy (5), but $\alpha_k = kn$ does not. Consequently, we may restrict the set to which α belongs to be $\{1, \ldots, n-1, n+1, \ldots, 2n-1\}$.

Define $i = \alpha + 1$ if $\alpha < n$ and $i = 2n + 1 - \alpha$ otherwise, and verify that $i \in \{2, \ldots, n\}$ in either case and that $(i - 1)(2j - 1) = \pm 1 - \beta 2n$ for any $\beta \in \mathbb{Z}$. Hence,

$$\begin{aligned} |a^n_{ij}| &= \sqrt{\frac{2}{n}} \left| \cos\left(\frac{\pi}{2n} (i-1)(2j-1) \right) \right| \\ &= \sqrt{\frac{2}{n}} \left| \cos\left(\pm \frac{\pi}{2n} - \beta\pi \right) \right| \\ &= \sqrt{\frac{2}{n}} \cos\left(\frac{\pi}{2n} \right) \\ &\geqslant \frac{1}{\sqrt{n}} \\ &= a^n_{1j}, \end{aligned}$$

by virtue of $n \geqslant 2$.

Next, assume $2j - 1$ and $2n$ are not coprime integers. Denote by d their greatest common divisor and verify that

$d \geqslant 3$. Define $i = 2n/d + 1$ and check i may take values on $\{2, \ldots, n\}$. Since $2j - 1 = \beta d$ for some $\beta \in \mathbb{Z}$, it follows that

$$\left|\cos\left(\frac{\pi}{2n}\,(i-1)\,(2j-1)\right)\right| = |\cos(\beta\pi)| = 1,$$

and therefore $|a^n_{ij}| > a^n_{1j}$. ■

The following result, Corollary 1, compares the sensitivity of the coefficients of the DCT, DST and DHT, with that of I_r, the identity function used by the naive record-perturbation approach, which we described at the beginning of this section. Also, the corollary shows the low sensitivity of the DCT coefficients of the first row.

Corollary 1. *Let* $\mathrm{GS}(I_r)$ *denote the L1-sensitivity of* I_r*, and* $f^{\mathrm{c}}_{c_{ij}}$ *the query function that returns the element* (i, j) *of the DCT. For any* $i = 1, \ldots, n$ *and any* $j = 1, \ldots, d$*,*

(i) $\mathrm{GS}(f^{\mathrm{c}}_{c_{1j}}) \leqslant \mathrm{GS}(f^{\mathrm{c}}_{c_{ij}})$,

(ii) $\mathrm{GS}(f_{c_{ij}}) \leqslant \frac{2}{\sqrt{nd}}\mathrm{GS}(I_r)$.

Proof: The first claim is immediate from Proposition 1 and Lemma 1, by noting that

$$a^n_{1j} \leqslant |a^n_{ij}| \leqslant |a^n_{i,r^*(i)}|$$

for the DCT matrix $\mathbf{a}^n$. For the same transform and for $i \geqslant 2$, it follows that

$$\begin{aligned}
\mathrm{GS}(f^{\mathrm{c}}_{c_{ij}}) &= \sqrt{\frac{2}{n}}\left|\cos\left(\frac{\pi}{2n}\,(i-1)\,(2r^*(i)-1)\right)\right| \times \\
&\quad \left(\frac{\Lambda_1}{\sqrt{d}} + \sum_{k=2}^{d}\Lambda_k\sqrt{\frac{2}{d}}\left|\cos\left(\frac{\pi}{2d}\,(j-1)\,(2k-1)\right)\right|\right) \\
&\leqslant \sqrt{\frac{2}{n}}\sqrt{\frac{2}{d}}\sum_{k=1}^{d}\Lambda_k \\
&= \frac{2}{\sqrt{nd}}\mathrm{GS}(I_r).
\end{aligned}$$

In the case of a DST and a DHT, an entirely analogous derivation leads to $\mathrm{GS}(f^{\mathrm{s}}_{c_{ij}}) \leqslant 2\mathrm{GS}(I_r)/\sqrt{nd}$ and $\mathrm{GS}(f^{\mathrm{h}}_{c_{ij}}) \leqslant 2\mathrm{GS}(I_r)/\sqrt{(n+1)(d+1)}$, respectively. Since $\mathrm{GS}(f^{\mathrm{c}}_{c_{1j}}) \leqslant \mathrm{GS}(f^{\mathrm{c}}_{c_{ij}})$ from claim (i), we prove the second statement. ■

Corollary 1 tells us that the sensitivity values of the transform coefficients are significantly lower, compared to that of the baseline identity function. Specifically, for $n = d$, $\mathrm{GS}(f_{c_{ij}})$ can be interpreted roughly as averaging Λ by the number of records (attributes).

Direct application of Lemma 1 allows us to examine the differences in terms of sensitivity among the cosine, sine and Hartley transforms. For ease of comparison, we define the *sensitivity relative difference* between transforms σ and ρ as

$$r_{\sigma\rho} = \frac{\mathrm{GS}(f^{\sigma}_{c_{ij}}) - \mathrm{GS}(f^{\rho}_{c_{ij}})}{\min\{\mathrm{GS}(f^{\sigma}_{c_{ij}}), \mathrm{GS}(f^{\rho}_{c_{ij}})\}}, \qquad (6)$$

where $\sigma, \rho \in \{\mathrm{c}, \mathrm{s}, \mathrm{h}\}$. Fig. 2 shows the percentage values of the quantities r_{cs}, r_{ch} and r_{sh} for two square block sizes, namely, $n = 8$ and $n = 16$.

Several remarks are in order from this figure. First, we observe that the DST is preferable to the DCT except for roughly two rows, $i = 1$ and $i = 5$ for $n = 8$, and similarly for $n = 16$ (Fig. 2(a)); this observation is consistent with the first claim of Corollary 1. When compared to the DHT, however, the sensitivities of the DST coefficients are observed to be much larger in odd rows for $n = 8$. In contrast, this latter transform exhibits smaller sensitivities in columns $1, 5, 9$ and 11 for $n = 16$.

In general, the sine and Hartley transforms seem to be more suitable, as reflected in Fig. 2(d), where for each coefficient we represent the transformation with the least sensitivity. This is evident for $n = 8$, where, for all but 6 coefficients, these transforms outperform the DCT. The case $n = 16$ is less clear although it still shows the DHT as the transformation with the largest number of coefficients with least sensitivity. We would like to stress that this does not signify the other two transforms are inappropriate. In fact, the suitability of any transform will hinge upon the block size, the individual attribute sensitivities Λ, and more importantly, the specific coefficients to be protected as well as the ability of the transform to compact energy.

4.3 Quantization

In source coding, lossy systems are characterized by the fact that the reconstructed signal is not identical to the source signal. The process that introduces the corresponding loss of information is called *quantization*, and the algorithm that performs the quantization process is referred to as *quantizer*. Although in image and video coding the information loss is due to analog-to-digital conversion, in a mild abuse of terminology we refer to quantization more generally as the process whereby distortion is introduced. In this subsection, we shall omit the block index j and therefore subindexes will denote elements of the corresponding matrices. For simplicity, we shall also drop the subindex of ε_j.

The purpose of introducing distortion is to satisfy a DP requirement. As we shall show next, our quantizer will be designed to cause the least possible loss of information while meeting this requirement. Although we shall be looking at the overall MSE in releasing $\tilde{\mathbf{X}}_j$ (instead of $\mathbf{X}_j$), a typical measure of performance for the quantizer is the *coding gain* [30], defined as the ratio

$$G_{\mathrm{Q}} = \frac{\mathrm{E}\,\|\mathbf{C}\|^2_{\mathrm{F}}}{\mathrm{E}\,\|\mathbf{C} - \tilde{\mathbf{C}}\|^2_{\mathrm{F}}}, \qquad (7)$$

which is simply the signal-to-noise (SNR) power ratio achieved by the quantizer.

Our quantizer aims to appropriately select a subset of transform coefficients of $\mathbf{C}$, protect them through the Laplace mechanism, and eliminate the remaining ones. Let t be the number of retained coefficients, and $\boldsymbol{\varepsilon} \in \mathbb{R}^{n \times d}_{+}$ a matrix with the privacy budget ε_{ij} assigned to each of them. We consider implicitly that $\varepsilon_{ij} = 0$ if the transform coefficient C_{ij} is not selected. On the other hand, we assume $\|\boldsymbol{\varepsilon}\|_1 = \varepsilon_{Q_L} < \varepsilon$. Accordingly, the quantization module outputs

$$\tilde{C}_{ij} = \begin{cases} C_{ij} + L\left(0, \mathrm{GS}(f_{c_{ij}})/\varepsilon_{ij}\right), & \text{if } C_{ij} \text{ is selected} \\ 0, & \text{otherwise,} \end{cases}$$

where L is a zero-mean Laplacian r.v. with scale $\mathrm{GS}(f_{c_{ij}})/\varepsilon_{ij}$.

Quantization therefore incurs two sources of error: first, the error due to eliminating $nd - t$ coefficients, and secondly,

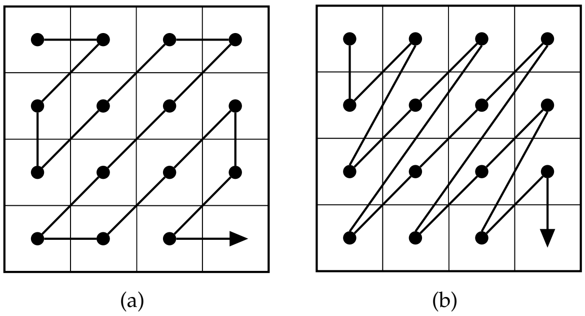


Fig. 3: (a) Zig-zag and (b) diagonal orders for scanning a transform coefficient matrix with $n = d = 4$. In this figure, the sequence of matrix indexes specified by the zig-zag order is $\mathcal{O} = ((1,1),(1,2),(2,1),\ldots,(4,4))$.

the noise added to the remaining t coefficients to attain ε_{Q_L}-DP. We shall refer to these two errors as *coefficients-removal and Laplace errors*, respectively.

Clearly, there is a trade-off between such two errors. For a fixed ε_{Q_L}, if t approaches nd, the coefficients-removal error will likely be small or even negligible, but the privacy budget will need to be distributed among a significant number of coefficients, thereby causing the Laplace error to be large. The opposite occurs if t is small compared to nd. The fundamental questions that we address next are (i) how to choose t; and (ii) given t, which coefficients of $\mathbf{C}$ need to be protected, so that these two decisions cause the minimum overall distortion. We tackle these two questions in reverse order.

4.3.1 Selection and Protection of Transform Coefficients

Intuitively, in the choice of transform coefficients, their global sensitivities as well as the possible values they may take on will play an important role. Let $\nu_{ij} = \Pr\{C_{ij} > 0\}$. In video coding, it is typically advantageous to arrange the transform coefficients C_{ij} of a block in the order of decreasing probabilities ν_{ij}. However, the transform coefficients of a block have to be transmitted in a certain order that is also known to the decoder. Making this order data-dependent is clearly inefficient, since it would need to be conveyed on a per block basis.

Most video coding standards adopt a predefined, signal-independent approach by leveraging the fact that, in transformed error blocks, ν_{ij} usually decreases with increasing frequency indexes i and j. A signal-independent scan in video coding that approximately arranges the transform coefficient values in the desired order is the zig-zag scan. This scan, which is illustrated in Fig. 3(a) for the example of a 4 × 4 block, is used in most video coding standards. H.265, also known as MPEG-H Part 2 or high efficiency video coding (HEVC), may operate with the diagonal scan depicted in Fig. 3(b). The two scans have similar properties but the latter provides some benefits for certain implementations.

In our case, arranging the coefficients of $\mathbf{C}$ according to ν_{ij} is a data-dependent operation and, as such, would not satisfy DP. To cope with this, we follow an approach entirely analogous to that of video coding and assume the coefficients of $\mathbf{C}$ are arranged in an order defined by a *coefficients order* $\mathcal{O}$. Accordingly, given such an order and a number t of coefficients to protect, our quantizer proceeds just by selecting the first t coefficients in the given order. Next, we examine how these coefficients are protected.

Denote by $\xi_{\tilde{X}}(\varepsilon, \mathcal{O}, t)$ the MSE incurred in outputting $\tilde{\mathbf{X}}$ instead of $\mathbf{X}$, where conveniently we make explicit its dependency with the assignment of the privacy budget to the t selected coefficients, and with the parameters specifying which concrete coefficients are to be protected. Our next result shows that this error consists in the sum of the MSEs due to the removal of coefficients and DP protection at the quantizer.

Lemma 2 (Laplace and coefficients-removal errors)**.** *Given* ε, $\mathcal{O}$ *and* t, *the MSE in releasing* $\tilde{\mathbf{X}}$ *rather than* $\mathbf{X}$ *is*

$$\xi_{\tilde{X}}(\varepsilon, \mathcal{O}, t) = 2\sum_{k=1}^{t} \mathrm{GS}(f_{c_{\mathcal{O}(k)}})^2/\varepsilon^2_{\mathcal{O}(k)} + \sum_{k=t+1}^{nd} \mathrm{E}\, C^2_{\mathcal{O}(k)}.$$

Proof: From (2), we know that $\mathrm{E}\,||\mathbf{X} - \tilde{\mathbf{X}}||^2_{\mathrm{F}} = \mathrm{E}\,||\mathbf{E} - \tilde{\mathbf{E}}||^2_{\mathrm{F}}$. On the other hand,

$$\begin{aligned}
\mathrm{E}\,||\mathbf{E} - \tilde{\mathbf{E}}||^2_{\mathrm{F}} &= \mathrm{E}\,\mathrm{tr}\left((\mathbf{E} - \tilde{\mathbf{E}})^{\mathrm{T}}(\mathbf{E} - \tilde{\mathbf{E}})\right) \\
&= \mathrm{E}\,\mathrm{tr}\left((\mathbf{a}^{n^{\mathrm{T}}}(\mathbf{C} - \tilde{\mathbf{C}})\mathbf{a}^d)^{\mathrm{T}}(\mathbf{a}^{n^{\mathrm{T}}}(\mathbf{C} - \tilde{\mathbf{C}})\mathbf{a}^d)\right) \\
&= \mathrm{E}\,\mathrm{tr}\left(\mathbf{a}^{d^{\mathrm{T}}}(\mathbf{C} - \tilde{\mathbf{C}})^{\mathrm{T}}\mathbf{a}^n\mathbf{a}^{n^{\mathrm{T}}}(\mathbf{C} - \tilde{\mathbf{C}})\mathbf{a}^d\right) \\
&\overset{\text{(a)}}{=} \mathrm{E}\,\mathrm{tr}\left(\mathbf{a}^{d^{\mathrm{T}}}(\mathbf{C} - \tilde{\mathbf{C}})^{\mathrm{T}}(\mathbf{C} - \tilde{\mathbf{C}})\mathbf{a}^d\right) \\
&\overset{\text{(b)}}{=} \mathrm{E}\,\mathrm{tr}\left((\mathbf{C} - \tilde{\mathbf{C}})^{\mathrm{T}}(\mathbf{C} - \tilde{\mathbf{C}})\mathbf{a}^d\mathbf{a}^{d^{\mathrm{T}}}\right) \\
&\overset{\text{(c)}}{=} \mathrm{E}\,||\mathbf{C} - \tilde{\mathbf{C}}||^2_{\mathrm{F}},
\end{aligned}$$

where

(a) and (c) follow from the orthogonality of $\mathbf{a}^n$ and $\mathbf{a}^d$, respectively; and

(b) uses the fact that the trace is invariant under cyclic permutations.

Using the matrix indexes given by $\mathcal{O}$, it follows that

$$\begin{aligned}
\mathrm{E}\,||\mathbf{C} - \tilde{\mathbf{C}}||^2_{\mathrm{F}} &= \mathrm{E}\sum_{i=1}^{n}\sum_{j=1}^{d}(C_{ij} - \tilde{C}_{ij})^2 \\
&= \sum_{k=1}^{nd} \mathrm{E}(C_{\mathcal{O}(k)} - \tilde{C}_{\mathcal{O}(k)})^2 \\
&= \sum_{k=1}^{t} \mathrm{E}\left(L(0, \mathrm{GS}(f_{c_{\mathcal{O}(k)}})/\varepsilon_{\mathcal{O}(k)})\right)^2 \\
&\quad + \sum_{k=t+1}^{nd} \mathrm{E}\, C^2_{\mathcal{O}(k)}.
\end{aligned}$$

Finally, we derive the expression claimed in the statement by recalling that the variance of a Laplacian r.v. of scale parameter b is $2b^2$. ■

Lemma 2 provides the MSE incurred by quantization, and shows that the Laplace and coefficients-removals errors are strictly increasing and non-increasing with t, respectively. We explore next how to distribute the privacy budget among the selected coefficients so that the total error is minimized.

Denote by $\boldsymbol{\varepsilon}^*$ the optimal assignment of ε_{Q_L},

$$\boldsymbol{\varepsilon}^* = \underset{\substack{\boldsymbol{\varepsilon} \\ \varepsilon_{\mathcal{O}(k)}>0,\, k=1,\dots,t \\ \sum_k \varepsilon_{\mathcal{O}(k)}=\varepsilon_{Q_L}}}{\arg\min} \quad \xi_{\tilde{X}}(\boldsymbol{\varepsilon}, \mathcal{O}, t). \tag{8}$$

Theorem 1 (Optimal assignment of ε_{Q_L})**.** *For any given $\mathcal{O}$ and any $t \in \{1, \dots, nd\}$, the optimal assignment $\boldsymbol{\varepsilon}^*$ is*

$$\varepsilon^*_{\mathcal{O}(i)} = \frac{\mathrm{GS}(f_{c_{\mathcal{O}(i)}})^{2/3}}{\sum_{k=1}^{t} \mathrm{GS}(f_{c_{\mathcal{O}(k)}})^{2/3}} \varepsilon_{Q_L}$$

for $i = 1, \dots, t$, and the corresponding minimum MSE yields

$$\xi_{\tilde{X}}(\boldsymbol{\varepsilon}^*, \mathcal{O}, t) = \frac{2}{\varepsilon^2_{Q_L}} \left(\sum_{k=1}^{t} \mathrm{GS}(f_{c_{\mathcal{O}(k)}})^{2/3} \right)^3 + \sum_{k=t+1}^{nd} \mathrm{E}\, C^2_{\mathcal{O}(k)}.$$

Proof: The proof is organized in two steps. First, we show that the optimization problem implicit in (8) is convex. Secondly, we use Karush-Kuhn-Tucker (KKT) conditions to solve the problem.

For notational conciseness, we denote $\varepsilon_{\mathcal{O}(1)}, \dots, \varepsilon_{\mathcal{O}(t)}$ by $\varepsilon_1, \dots, \varepsilon_t$, and define

$$\gamma_k = 2\mathrm{GS}(f_{c_{\mathcal{O}(k)}})^2 \quad \text{and} \quad f_k(\varepsilon_k) = \gamma_k / \varepsilon_k^2.$$

To show that the problem is convex, note that, from Lemma 2,

$$\xi_{\tilde{X}}(\boldsymbol{\varepsilon}, \mathcal{O}, t) - \sum_{k=t+1}^{nd} \mathrm{E}\, C^2_{\mathcal{O}(k)}$$

is the sum of strictly convex functions f_k, and observe that the inequality and equality constraint functions are linear and affine. Since the objective and constraint functions are also differentiable and Slater's constraint qualification holds, KKT conditions are necessary and sufficient conditions for optimality [26, §5]. The application of these optimality conditions leads to the following Lagrangian cost,

$$\mathcal{L} = \sum f_k(\varepsilon_k) - \sum \lambda_k \varepsilon_k - \mu \left(\sum \varepsilon_k - \varepsilon_{Q_L} \right),$$

and finally to the conditions

$$\begin{aligned}
&f_k'(\varepsilon_k) + \lambda_k - \mu = 0 && \text{(dual optimality)},\\
&\lambda_k\, \varepsilon_k = 0, && \text{(complementary slackness)},\\
&\lambda_k \geqslant 0 && \text{(dual feasibility)},\\
&\varepsilon_k > 0,\ \textstyle\sum \varepsilon_k = \varepsilon_{Q_L}, && \text{(primal feasibility)}.
\end{aligned}$$

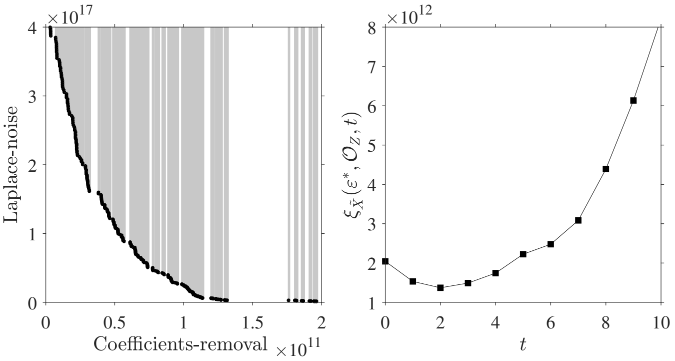
Since $f_k''(\varepsilon_k) = 6\gamma_k/\varepsilon_k^4 > 0$, f_k' is strictly increasing, and, interpreted as a function from $(0, \varepsilon_{Q_L})$ to $f_k'((0, \varepsilon_{Q_L}))$, invertible. Denote the inverse by $f_k'^{-1}$. Since $\varepsilon_k > 0$, it follows from the complementary slackness condition that $\lambda_k = 0$, which, by the dual optimality condition, implies $f_k'(\varepsilon_k) = \mu$, or equivalently, $\varepsilon_k = f_k'^{-1}(\mu)$.

From the primal equality constraint,

$$\sum_{k=1}^{t} f_k'^{-1}(\mu) = \sum_{k=1}^{t} \sqrt[3]{-2\gamma_k/\mu} = \varepsilon_{Q_L},$$

and hence

$$\mu = -\frac{2}{\varepsilon^3_{Q_L}} \left(\sum_{k=1}^{t} \sqrt[3]{\gamma_k} \right)^3.$$



(a) (b)

Fig. 4: (a) Trade-off between the Laplace-noise error and the coefficients-removal error, and (b) minimum MSE due to quantization. Each black point in (a) corresponds to one of the 33×14 possible values of t for which the Laplace-noise error is minimized. These points constitute the *optimal* trade-off. The points in gray, on the other hand, reflect a non-optimal assignment of $\boldsymbol{\varepsilon}$. In (b), we observe that $t = 2$ minimizes the minimum MSE. In this example, $\varepsilon_{Q_L} = 1$ and Λ is the maximum value of each attribute within the block.

Substituting the above expression for μ into $f_k'^{-1}(\mu)$ leads to the expression of the optimal $\boldsymbol{\varepsilon}$ given in the theorem. Then, the MSE follows by substituting the solution into $\xi_{\tilde{X}}(\boldsymbol{\varepsilon}, \mathcal{O}, t)$. ■

A couple of remarks follow from Theorem 1. On the one hand, the optimal assignment of ε_{Q_L} conforms to intuition, as those coefficients with smaller sensitivities are assigned smaller $\varepsilon_{\mathcal{O}(i)}$. On the other hand, we observe that the MSE due to the Laplace error is proportional to the inverse of the square of ε_{Q_L}. This means, for example, that increasing ε_{Q_L} from 1 to 2 implies a reduction by 75 percent in MSE.

4.3.2 *Choice of t and Transform*

For a given transform and $\mathcal{O}$, the trade-off between the Laplace and the coefficients-removal errors is determined by t. In Fig. 4, we provide an example of this trade-off in the case of (i) a 32×13 input block $\mathbf{X}$ corresponding to the first 32 records of the "Census" data set [33]; (ii) the DCT; (iii) a zig-zag order, and (iv) no prediction.

In this particular example we show that there exists a value of t minimizing the sum of the two errors above for the DCT. This subsection aims to compute, in a DP manner, this value of t and the transform $\sigma \in \{\mathrm{c}, \mathrm{s}, \mathrm{h}\}$ that jointly minimize such total error[8]. Since this computation is a data-dependent operation, we resort to the exponential mechanism [34] of DP. Henceforth, we shall denote the optimal values of those two parameters by t^* and σ^*. For notational compactness, we shall use $\boldsymbol{\kappa}$ to refer to the tuple of quantization parameters $(\mathcal{O}, t, \sigma)$.

The exponential mechanism requires designing a proper *scoring function*. To investigate the impact of this design decision on our quantizer, we consider a parametrized family ω_θ of such functions, where θ denotes the exponent of both the Laplace-noise and the coefficients-removal errors in $\xi_{\tilde{X}}(\boldsymbol{\varepsilon}^*, \boldsymbol{\kappa})$.

[8] Note that minimizing $\mathrm{E}\, ||\mathbf{X} - \tilde{\mathbf{X}}||^2_{\mathrm{F}}$ implies maximizing the coding gain of the quantizer, since $\mathrm{E}\, ||\mathbf{C} - \tilde{\mathbf{C}}||^2_{\mathrm{F}} = \mathrm{E}\, ||\mathbf{X} - \tilde{\mathbf{X}}||^2_{\mathrm{F}}$ from the proof of Lemma 2.

Intuitively, the purpose of using these error-based functions is for the exponential mechanism to favor values of t and σ causing less MSE. Let T and Σ be the r.v.'s modeling the response of this mechanism, and ε_{Q_E} the desired level of protection of said mechanism. Ideally, we would like the joint PMF $p_{T\Sigma}$ to be as large as possible for $T = t^*$ and $\Sigma = \sigma^*$, and as small as possible for the rest of values. Since $p_{T\Sigma}(t, \sigma; \theta)$ is proportional to $\varepsilon_{Q_E}\, \omega_\theta(\mathbf{c}, \boldsymbol{\kappa})/\mathrm{GS}(\omega_\theta(\mathbf{c}, \boldsymbol{\kappa}))$, one might be tempted to choose $\theta >> 1$. However, this may not be an appropriate choice since the sensitivity of the corresponding function is likely to increase accordingly.

For conciseness, our analysis only contemplates the cases $\theta = 1/2$ and $\theta = 1$, and for simplicity the scoring functions operate with current observed values rather than expected values. Accordingly, the respective scoring functions are

$$\omega_{1/2}(\mathbf{c}, \boldsymbol{\kappa}) = -\frac{\sqrt{2}}{\varepsilon_{Q_L}} \left(\sum_{k=1}^{t} \mathrm{GS}(f^{\sigma}_{c_{\mathcal{O}(k)}})^{2/3} \right)^{3/2} - \sqrt{\sum_{k=t+1}^{nd} c^2_{\mathcal{O}(k)}},$$

and

$$\omega_1(\mathbf{c}, \boldsymbol{\kappa}) = -\xi_{\tilde{X}}(\varepsilon^*, \boldsymbol{\kappa}).$$

Our next result computes upper bounds on the sensitivities of these two functions. Before proceeding, however, we introduce some notation. Denote by $\boldsymbol{\Lambda}$ a matrix of dimension $n \times d$ with all rows being Λ^{T}, and by $f_{\mathbf{c}}$ the query function that returns all elements of the transform block $\mathbf{c}$. Accordingly, define

$$\bar{\sigma} = \operatorname*{arg\,max}_{\sigma \in \{\mathrm{c,s,h}\}} \|\mathrm{GS}(f^{\sigma}_{\mathbf{c}})\|_{\mathrm{F}}.$$

Furthermore, the absolute value function, when applied to a matrix, will denote the element-wise absolute value of such matrix.

Lemma 3 (Sensitivities of $\omega_{1/2}$ and ω_1). *Under the assumptions of Lemma 1, and for a given prediction block $\hat{\mathbf{x}}$, the L1-sensitivities of the scoring functions $\omega_{1/2}$ and ω_1 satisfy*

(i) $\mathrm{GS}(\omega_{1/2}(\mathbf{c}, \boldsymbol{\kappa})) \leqslant \|\mathrm{GS}(f^{\bar{\sigma}}_{\mathbf{c}})\|_{\mathrm{F}}$,

(ii) $\mathrm{GS}(\omega_1(\mathbf{c}, \boldsymbol{\kappa})) < 2\|\mathrm{GS}(f^{\bar{\sigma}}_{\mathbf{c}})\|_{\mathrm{F}} \sqrt{\sum_{ij} \max\{\Lambda_j - \hat{x}_{ij}, \hat{x}_{ij}\}^2}$.

Proof: Let $\mathbf{x}$ and $\mathbf{x}'$ be two neighboring input blocks, and $\mathbf{c}$ and $\mathbf{c}'$ their corresponding transformed error blocks. For any $r \in \{1, \ldots, n\}$, denote by $x_r = (x_{r1}, \ldots, x_{rd})$ and $x'_r = (x'_{r1}, \ldots, x'_{rd})$ the respective values of the different record in either input block.

Let $\mathcal{O}$ be any order. For $k \in \{1, \ldots, nd\}$, let $\mathcal{O}(k, 1)$ and $\mathcal{O}(k, 2)$ denote the first and the second index of $\mathcal{O}$, respectively. Accordingly, define

$$J_{\mathcal{O}(k)}(x_r, x'_r) = \sum_{l=1}^{d} a^n_{\mathcal{O}(k,1),r}\, a^d_{\mathcal{O}(k,2),l} (x_{rl} - x'_{rl}),$$

where $\mathbf{a}^n$ and $\mathbf{a}^d$ are transformation matrices of dimensions $n \times n$ and $d \times d$, as specified in (3).

From the definition of L1-sensitivity, we have that

$$\begin{aligned}
\mathrm{GS}(\omega_{1/2}(\mathbf{c}, \boldsymbol{\kappa})) &= \max_{\mathbf{x}, \mathbf{x}', \boldsymbol{\kappa}} \left| \sqrt{\sum_{k=t+1}^{nd} c^2_{\mathcal{O}(k)}} - \sqrt{\sum_{k=t+1}^{nd} c'^2_{\mathcal{O}(k)}} \right| \\
&\leqslant \max_{\mathbf{x}, \mathbf{x}', \boldsymbol{\kappa}} \sqrt{\sum_{k=t+1}^{nd} (c_{\mathcal{O}(k)} - c'_{\mathcal{O}(k)})^2} && (9) \\
&= \sqrt{\max_{x_r, x'_r, \boldsymbol{\kappa}} \sum_{k=t+1}^{nd} J_{\mathcal{O}(k)}(x_r, x'_r)^2} && (10) \\
&\leqslant \sqrt{\max_{\boldsymbol{\kappa}} \sum_{k=t+1}^{nd} \max_{x_r, x'_r} J_{\mathcal{O}(k)}(x_r, x'_r)^2} && (11) \\
&= \sqrt{\max_{\boldsymbol{\kappa}} \sum_{k=t+1}^{nd} \left(\max_{x_r, x'_r} |J_{\mathcal{O}(k)}(x_r, x'_r)| \right)^2} && (12) \\
&= \sqrt{\max_{\boldsymbol{\kappa}} \sum_{k=t+1}^{nd} \mathrm{GS}(f^{\sigma}_{c_{\mathcal{O}(k)}})^2}, && (13)
\end{aligned}$$

where (9) follows from the reverse triangle inequality and does not depend on $\hat{\mathbf{x}}$; (10) results from (4) and from the strict monotonicity of the square root function; (11) follows from the fact that the maximum of a sum is at most the sum of maxima; (12) holds since the squaring function preserves the order of nonnegative numbers; (13) follows from Lemma 1; and from (13) we immediately verify claim (i) in the lemma, as it is maximized for $t = 0$ (and hence for any $\mathcal{O}$) and $\sigma = \bar{\sigma}$.

To prove the second claim, we use $\|\mathbf{c}\|^2_{2,t}$ to denote $\sum_{k=t+1}^{nd} c^2_{\mathcal{O}(k)}$, and $\|\mathbf{c}'\|^2_{2,t}$ analogously. Note that this notation uses the Euclidean norm instead of the Frobenius norm since we interpret $\mathbf{c}$ and $\mathbf{c}'$ as vectors, indexed by $\mathcal{O}$.

That being said, observe that

$$\left|\, \|\mathbf{c}\|^2_{2,t} - \|\mathbf{c}'\|^2_{2,t} \,\right| = \left|\, \|\mathbf{c}\|_{2,t} - \|\mathbf{c}'\|_{2,t} \,\right| (\|\mathbf{c}\|_{2,t} + \|\mathbf{c}'\|_{2,t}),$$

and that

$$\max_{x,y} \{g(x,y) h(x,y)\} \leqslant \max_{x,y} g(x,y) \max_{x,y} h(x,y)$$

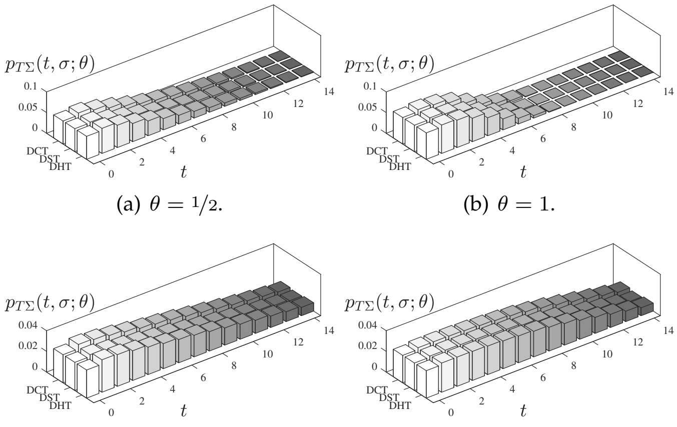
for any x, y and any positive real-valued functions g, h. Accordingly, it follows that

$$\begin{aligned}
\mathrm{GS}(\omega_1(\mathbf{c}, \boldsymbol{\kappa})) \leqslant \max_{\mathbf{x}, \mathbf{x}', \boldsymbol{\kappa}} \left|\, \|\mathbf{c}\|_{2,t} - \|\mathbf{c}'\|_{2,t} \,\right| \times \\
\times \max_{\mathbf{x}, \mathbf{x}', \boldsymbol{\kappa}} \|\mathbf{c}\|_{2,t} + \|\mathbf{c}'\|_{2,t}.
\end{aligned}$$

We know from claim (i) that the maximum on the left-hand side is upper bounded by $\|\mathrm{GS}(f^{\bar{\sigma}}_{\mathbf{c}})\|_{\mathrm{F}}$. On the other hand, we have that

$$\begin{aligned}
\max_{\mathbf{x}, \mathbf{x}', \boldsymbol{\kappa}} \|\mathbf{c}\|_{2,t} + \|\mathbf{c}'\|_{2,t} &= \max_{\mathbf{x}, \mathbf{x}', \sigma} \|\mathbf{c}\|_2 + \|\mathbf{c}'\|_2 \\
&= \max_{\mathbf{x}, \mathbf{x}'} \|\mathbf{x} - \hat{\mathbf{x}}\|_{\mathrm{F}} + \|\mathbf{x}' - \hat{\mathbf{x}}\|_{\mathrm{F}}, && (14)
\end{aligned}$$

where (14) follows from the orthogonality of the three transforms under consideration. To complete the proof, note that each summand in (14) is maximized for either $x_{ij} = \Lambda_j$ or $x_{ij} = 0$, depending on the largest absolute difference



(a) $\theta = 1/2$.

(b) $\theta = 1$.

(c) $\theta = 1/2$.

(d) $\theta = 1$.

Fig. 5: PMF $p_{T\Sigma}(\theta)$ of the exponential mechanism for $\theta = 1/2$ and $\theta = 1$. We have used the zig-zag order, $\varepsilon_{Q_L} = \varepsilon_{Q_E} = 1$, and $\hat{\mathbf{x}} = 0$. The input data are an 8×8 block corresponding to the last 8 records and first 8 attributes of the "Census" data set (a,b); and a 48×13 block corresponding to the last 48 records and all attributes of the same data set (c,d). The pairs (t, σ) that minimize $\xi_{\tilde{X}}(\boldsymbol{\varepsilon}^*, \boldsymbol{\kappa})$ are $(1, \mathrm{c})$ for the former block and $(2, \mathrm{h})$ for the latter.

between x_{ij} and $\hat{x}_{ij}$. The strict inequality in claim (ii) is due to the fact that $\mathbf{x}$ and $\mathbf{x}'$ must differ in one record. ■

Several conclusions follow from Lemma 3. First, and most evident, the upper bounds on the sensitivities of $\omega_{1/2}$ and ω_1 do not depend on $\mathcal{O}$. The reason lies in that the bounds are maximized for $t = 0$, which means all terms $\mathrm{GS}(f^{\sigma}_{c_{\mathcal{O}(k)}})^2$ in (13) must be added up. Likewise, the upper bound on the sensitivity of $\omega_{1/2}$ does not hinge on $\hat{\mathbf{x}}$ either, as the difference $c_{\mathcal{O}(k)} - c'_{\mathcal{O}(k)}$ in (9) does not. However, this is not the case for $\theta = 1$, which requires that the prediction module share $\hat{\mathbf{x}}$ with the quantization module.

In this latter case, we can observe the straightforward effect that prediction may have on the obtained bound. Specifically, it is immediate to verify that

$$\frac{1}{4}\|\boldsymbol{\Lambda}\|_{\mathrm{F}}^2 \leqslant \sum_{ij} \max\{\Lambda_j - \hat{x}_{ij}, \hat{x}_{ij}\}^2 \leqslant \|\boldsymbol{\Lambda}\|_{\mathrm{F}}^2,$$

which indicates that, to reduce the sensitivity bound of ω_1 and thus obtain more accurate results from the exponential mechanism, the predictions $\hat{\mathbf{x}} = 0$ (right inequality) and $\hat{\mathbf{x}} = \mathbf{x} - \boldsymbol{\Lambda}/2$ (left inequality) represent worst and best-case scenarios. We note that this latter prediction simply reduces the domain of each attribute to be $[0, \Lambda_j/2]$.

Another interesting conclusion is that the sensitivity results are valid for any set of orthogonal, separable, two-dimensional transforms, which extends the scope of our selection algorithm to include the vast majority of transform-coding techniques.

Finally, we observe that squaring the error terms in $\omega_{1/2}$ (i.e., moving from $\theta = 1/2$ to $\theta = 1$) has a significant impact on L1-sensitivity. While the resulting function may yield larger scores for (t^*, σ^*) (which may help the exponential mechanism choose the optimal number of coefficients and transform), we note its sensitivity may in the worst case become $2\|\boldsymbol{\Lambda}\|_{\mathrm{F}}$ times larger than that of $\omega_{1/2}$, which may lose out the benefits of such an exponentiation.

Despite this latter observation, we would like to stress that determining which function will cause the least distortion is not possible *a priori*, since one would need to know $\mathbf{c}$ in advance. The appropriateness of $\omega_{1/2}$ and ω_1 will therefore depend on the actual data. Fig. 5 reflects this situation by comparing the PMFs $p_{T\Sigma}(\theta)$ for $\theta = 1/2$ and $\theta = 1$, and for two different input blocks. In Figs. 5(a,b), the smaller dispersion of $\theta = 1$ and the fact that $\mathrm{E}_{p_{T\Sigma}(1)}[T|\sigma]$ is close to t^* for all σ, makes this function more suitable. In Figs. 5(c,d), however, $\theta = 1/2$ seems to be more appropriate: the PMF exhibits a smaller dispersion than $\theta = 1$, and it attains its maximum value exactly at $t^* = 2$ for the three transforms.

Algorithm 1: Transform coding and quantization.

```
   Input: An input block $\mathbf{X}$; a prediction block $\hat{\mathbf{X}}$; a
          coefficients order $\mathcal{O}$; the respective privacy
          parameters $\varepsilon_{Q_L}$ and $\varepsilon_{Q_E}$ of the Laplace and
          the exponential mechanisms; the
          scoring-function parameter $\theta$
   Output: A protected error block $\tilde{\mathbf{E}}$ satisfying
           $(\varepsilon_{Q_L} + \varepsilon_{Q_E})$-DP
 1 Compute $\omega_\theta(\mathbf{C}, \boldsymbol{\kappa})$ for the given order, all
     $t = 0 \ldots, nd$ and all $\sigma \in \{\mathrm{c}, \mathrm{s}, \mathrm{h}\}$
 2 Calculate the upper bounds[9]on the L1-sensitivity of
     $\omega_\theta(\mathbf{C}, \boldsymbol{\kappa})$ from Lemma 3
 3 Calculate $p_{T\Sigma}(\theta)$, being $p_{T\Sigma}(t, \sigma; \theta)$ proportional to
     $\exp\left(\varepsilon_{Q_E}\, \omega_\theta(\mathbf{c}, \boldsymbol{\kappa})/2\mathrm{GS}(\omega_\theta(\mathbf{c}, \boldsymbol{\kappa}))\right)$
 4 Generate a random draw $(T, \Sigma)$ from $p_{T\Sigma}(\theta)$
 5 Compute $\mathbf{C}$ as the $\Sigma$ transform of $\mathbf{E}$
 6 From Theorem 1, compute $\boldsymbol{\varepsilon}^*$ for the selected $\Sigma$
     transform so that $\sum_k \varepsilon^*_{\mathcal{O}(k)} = \varepsilon_{Q_L}$
 7 for $k = 1, \ldots, T$ do
 8     Generate a random draw $L$ from a zero-mean
         Laplace distribution and scale $\mathrm{GS}(f^{\Sigma}_{c_{\mathcal{O}(k)}})/\varepsilon^*_{\mathcal{O}(k)}$
 9     Set $\tilde{C}_{\mathcal{O}(k)} = C_{\mathcal{O}(k)} + L$
10 end
11 for $k = T + 1, \ldots, nd$ do
12     Set $\tilde{C}_{\mathcal{O}(k)} = 0$
13 end
14 Compute $\tilde{\mathbf{E}}$ as the inverse $\Sigma$ transform of $\tilde{\mathbf{C}}$
15 return $\tilde{\mathbf{E}}$.
```

The joint operation of the modules analysis, quantization and synthesis is summarized in Algorithm 1. The interaction among the three modules is reflected in lines 5 and 14, where quantization decides on the transform to be used by the transform-coding modules. Since quantization also requires the prediction block to compute $\mathrm{GS}(\omega_1(\mathbf{c}, \boldsymbol{\kappa}))$, the algorithm is input $\mathbf{X}$ and $\hat{\mathbf{X}}$, rather than just $\mathbf{E}$.

Returning to the notation of block subindexes, we also note that a decision must be made with regard to the distribution of the privacy budget ε_j available for each block and frame. In Algorithm 1 we make no assumption, apart from the fact that the budget devoted to the Laplace and to the exponential mechanism must satisfy $\varepsilon_{Q_L} + \varepsilon_{Q_E} \leqslant \varepsilon_j$.

[9] The bounds of Lemma 3 are for $\theta = 1/2$ and $\theta = 1$.

4.4 Prediction

Transform coding is a simple albeit efficient technique for utilizing statistical dependencies among the records within a single transform block. For additionally exploiting dependencies among transform blocks within a same or different frame, image and video coding relies on prediction techniques.

In video compression, there exist two classical prediction modes, *intra-prediction* and *inter-prediction*. In the former mode, the transform coefficients or original samples of a transform block are predicted using already coded samples of neighboring blocks. That is to say, intra-prediction only leverages statistical dependencies inside frames. However, as video sequences usually contain significant temporal redundancies, the additional exploitation of dependencies among the different frames of a video sequence can notably enhance coding efficiency. This later approach is referred to as inter-prediction.

In this work, we propose a *hybrid* video coding scheme to protect database streams, meaning that the protection algorithm is a hybrid of three fundamental techniques, namely, transform coding for dependencies within blocks, and the two prediction modes above. However, unlike video compression, these modes will be applied in a more general sense: we shall allow both intra-prediction and inter-prediction to generate $\hat{\mathbf{X}}$ from reconstructed blocks of the *same* frame and from reconstructed blocks of *different* frames.

Intuitively, the better the future of an input block (modeled as a vector process) is predicted from its past output blocks and the more redundancy the input block contains, the less new information is contributed by each successive block of the database stream [35]; for a fixed privacy budget, if less information needs to be protected, less distortion is introduced.

Next we recover the subindex notation for blocks. A measure of prediction performance is the *closed-loop prediction gain ratio* [30], which is defined as

$$G_{\text{clp}} = \frac{\mathrm{E}\,||\mathbf{X}_j||_{\mathrm{F}}^2}{\mathrm{E}\,||\mathbf{E}_j||_{\mathrm{F}}^2}. \qquad (15)$$

From (2), (7), and (15), the overall SNR power ratio of the DPCM system can be expressed as

$$\text{SNR}_{\text{sys}} = \frac{\mathrm{E}\,||\mathbf{X}_j||_{\mathrm{F}}^2}{\mathrm{E}\,||\mathbf{X}_j - \tilde{\mathbf{X}}_j||_{\mathrm{F}}^2} = G_{\text{clp}}\, G_{\mathrm{Q}}. \qquad (16)$$

We shall adopt the most commonly used criterion for the optimality of a predictor [36], [30], the minimization of the denominator of (15), which implies the minimization of the variance and the mean of the prediction error.

We shall denote by Φ the *set of modes and types of prediction* of the video coding standards available to the module at hand. Accordingly, each $\phi \in \Phi$ will represent a unique configuration of the prediction module, e.g., the intra-mode of H.264 with horizontal prediction, the latter being the prediction type.

We shall consider *spatial* prediction modes[10], which operate with original samples, in contrast to those that estimate $\hat{\mathbf{X}}$ from transform coefficients. Formally,

$$\hat{\mathbf{X}}_j = f(\tilde{\mathbf{X}}_{j-1}, \tilde{\mathbf{X}}_{j-2}, \ldots, \tilde{\mathbf{X}}_{j-\pi}),$$

where the function f is chosen adequately to generate a good estimate of $\hat{\mathbf{X}}_j$ from the π past values of the reproduced process $\{S_i^\varepsilon\}$. Although a variety of "standard" functions will be considered for intra-prediction in our evaluation (a couple of examples are shown in Fig. 6), we shall only contemplate *block matching* [37] as inter-prediction technique. In our case, when applying block matching we will be selecting the reconstructed block that minimizes the denominator of (16). The reason for restricting to block matching is that we expect small inter-frame redundancies, in contrast to video sequences.

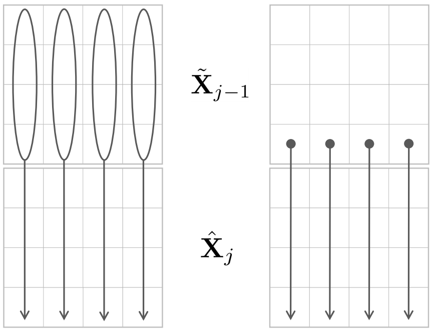


Fig. 6: Vertical intra-prediction modes of the standards H.263 (left) and H.264/MPEG-4 AVC (right). The former estimates $\hat{\mathbf{X}}_j$ from the column averages of the previously reproduced block $\tilde{\mathbf{X}}_{j-1}$. The latter uses directly adjacent samples of already protected blocks.

4.5 Preprocessing

Recall that a permutation matrix is a square $(0,1)$-matrix in which each row and each column has exactly one entry of 1 and zeros elsewhere. Let Ψ denote the set of permutation matrices. For any $\boldsymbol{\psi} \in \Psi$, notice that the product $\boldsymbol{\psi}\mathbf{X}$ is a permutation of the rows of $\mathbf{X}$.

Informally speaking, the goal of the preprocessing module is to find a permutation of the rows of $\mathbf{X}$ that helps the predictor generate a better prediction $\hat{\mathbf{X}}$ of $\mathbf{X}$. Since the actual $\hat{\mathbf{X}}$ is not available to the preprocessing module at the time when it is to permute $\mathbf{X}$, the module will be devised to find the permutation that minimizes the prediction error for *all* $\phi \in \Phi$. We shall see in Sec. 4.6 that this operation is conducted jointly with the encoder-control module.

The minimization of the prediction error, however, is not without constraints, since the cost of permuting must be kept to an acceptable level. In this work, we quantify this cost with the *Spearman's footrule* distance [38][11], given by

$$F(\boldsymbol{\psi}) = \|(\boldsymbol{\psi} - \mathbf{i}_n)(1, \ldots, n)\|_1,$$

which measures the total element-wise displacement from the original order, denoted by the identity matrix $\mathbf{i}_n$.

Formally, for a given set Φ of prediction modes and types, the preprocessing module is designed to compute the solution to the optimization problem

$$\min_{\substack{\phi \in \Phi \\ \boldsymbol{\psi} \in \Psi}} \|\boldsymbol{\psi}\mathbf{X} - \hat{\mathbf{X}}(\phi)\|_{\mathrm{F}}^2 \quad \text{subject to } F(\boldsymbol{\psi}) \leqslant c_{\mathcal{R}}, \qquad (17)$$

[10] Predictions in the sample domain have the advantage that predictor blocks can be generated for arbitrary prediction directions [35].

[11] The Spearman's footrule is the most popular metric to evaluate distances between permutations.

which describes the optimal trade-off between prediction error on the one hand, and on the other permutation or reordering cost. Intuitively, the larger the maximum acceptable cost, the smaller the prediction error and vice versa.

Let $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^{n \times d}$ and $z \in \mathbb{R}$ be the parameters of an *assignment problem with side constraints* (APSC) [39][12]. Recall that the formulation of an APSC in standard form is given by

$$\min_{\boldsymbol{\psi} \in \Psi} \sum_{ij} v_{ij} \psi_{ij} \quad \text{subject to} \sum_{ij} w_{ij} \psi_{ij} \leqslant z. \tag{18}$$

Our next results shows the equivalence of the problems (17) and (18).

Lemma 4. *For a fixed ϕ, the optimization problem* (17) *is an APSC.*

Proof: For brevity, we write $\hat{\mathbf{X}}$ instead of $\hat{\mathbf{X}}(\boldsymbol{\phi})$. Recall that the Frobenius inner product of two matrices $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{n \times d}$ is defined as $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathrm{F}} = \operatorname{tr}\left(\mathbf{a}^{\mathrm{T}} \mathbf{b}\right)$ and induces the corresponding Frobenius norm $\|\mathbf{a}\|_{\mathrm{F}} = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle_{\mathrm{F}}}$. Accordingly, we have that

$$\begin{aligned}
\|\boldsymbol{\psi}\mathbf{X} - \hat{\mathbf{X}}\|_{\mathrm{F}}^2 &= \|\boldsymbol{\psi}\mathbf{X}\|_{\mathrm{F}}^2 + \|\hat{\mathbf{X}}\|_{\mathrm{F}}^2 - 2\langle \hat{\mathbf{X}}, \boldsymbol{\psi}\mathbf{X} \rangle_{\mathrm{F}} \\
&= \operatorname{tr}\left(\mathbf{X}^{\mathrm{T}} \boldsymbol{\psi}^{\mathrm{T}} \boldsymbol{\psi} \mathbf{X}\right) + \|\hat{\mathbf{X}}\|_{\mathrm{F}}^2 - 2\langle \hat{\mathbf{X}}, \boldsymbol{\psi}\mathbf{X} \rangle_{\mathrm{F}} \\
&= \|\mathbf{X}\|_{\mathrm{F}}^2 + \|\hat{\mathbf{X}}\|_{\mathrm{F}}^2 - 2\langle \hat{\mathbf{X}}, \boldsymbol{\psi}\mathbf{X} \rangle_{\mathrm{F}} \qquad (19) \\
&= \|\mathbf{X}\|_{\mathrm{F}}^2 + \|\hat{\mathbf{X}}\|_{\mathrm{F}}^2 - 2\operatorname{tr}\left(\boldsymbol{\psi}\mathbf{X}\hat{\mathbf{X}}^{\mathrm{T}}\right), \qquad (20)
\end{aligned}$$

where (19) is due to the orthogonality of the permutation matrices, and (20) follows from the invariance of the trace under cyclic permutations.

Eq. (20) implies that minimizing $\|\boldsymbol{\psi}\mathbf{X} - \hat{\mathbf{X}}\|_{\mathrm{F}}^2$ for a given ϕ is equivalent to the problem of finding the permutation of the rows of $\mathbf{X}\hat{\mathbf{X}}^{\mathrm{T}}$ that maximizes the trace. The equivalence of problems (17) and (18) in terms of their objective functions is verified immediately by noting that (i) the objective function of Eq. (18) can be recast as the trace of $\boldsymbol{\psi}\boldsymbol{v}^{\mathrm{T}}$ and (ii) a problem in which the objective function is to be maximized can be converted into a minimization problem just by multiplying $\boldsymbol{v}$ by -1. To check the equivalence of the inequality constraint functions simply observe that $F(\boldsymbol{\psi}) = \operatorname{tr}(\boldsymbol{w}\boldsymbol{\psi})$ for $w_{ij} = |i - j|$. This completes the proof. ■

The strength of recasting (17) as an APSC lies in that it allows us to resort to efficient methods [39], [40] to compute the optimal permutation $\boldsymbol{\psi}^*$. This is of a great practical relevance as an APSC is NP-complete and our scheme must satisfy the delay constraint δ, as specified in Definition 3.

4.5.1 Extreme Regions of the Trade-off Plane

Even though powerful methods are available to compute $\boldsymbol{\psi}^*$, the fact that (17) is a minimization over all $\phi \in \Phi$ means we need to solve an APSC for each available prediction mode and type, and each input block. This imposes an important computational burden on the preprocessing module and may compromise the fulfillment of the delay constraint δ. In the special cases when the system is designed to operate at the extreme regions of the trade-off, we may alleviate this burden as described below.

Low Prediction Error. It can be shown [41] that

$$\max_{\boldsymbol{\psi} \in \Psi} F(\boldsymbol{\psi}) = \left\lfloor \frac{n^2}{2} \right\rfloor. \tag{21}$$

This result implies that if we accept permutation costs larger than or equal to $\lfloor n^2/2 \rfloor$, then the optimization problem (17) becomes an *unconstrained* linear assignment problem. Optimization problems of this kind can be solved in polynomial time $\mathcal{O}(n^4)$ with the original Hungarian algorithm and more efficiently with a bunch of algorithms that achieve $\mathcal{O}(n^3)$. We refer the reader to [42] for further details on this topic.

Low Reordering Cost. In the case when there are stringent, tight constraints on the permutation cost, intuitively the feasible set of (17) will mostly include permutations of nearby records. We contemplate two strategies, S1 and S2, that exploit this fact for the sake of computational efficiency.

Recall that an assignment problem can be regarded as a minimum weight perfect matching problem. S1 decomposes the blocks to be matched (i.e., $\mathbf{X}$ and $\hat{\mathbf{X}}$) into blocks of smaller sizes, and finds the matching of each of those sub-blocks. More specifically, it computes the solution of r optimization problems of the form (17), where $\mathbf{X}$ and $\hat{\mathbf{X}}$ are now replaced with $\mathbf{X}^i$ and $\hat{\mathbf{X}}^i$ and denote sub-blocks of size $n/r \times d$ containing the records $\frac{n(i-1)}{r} + 1, \ldots, \frac{n\,i}{r}$ of $\mathbf{X}$ and $\hat{\mathbf{X}}$, respectively. Naturally, $\sum_i c_{\mathcal{R}}^i = c_{\mathcal{R}}$.

The strategy S2, on the other hand, tackles the original problem with a weight matrix that prevents the matching of records belonging to different sub-blocks. Specifically, we consider the matrix

$$w_{ij} = \begin{cases} |i - j|, & \text{if } \frac{k-1}{r} n < i, j \leqslant \frac{k}{r} n \text{ for } k = 1, \ldots, r \\ \infty, & \text{otherwise,} \end{cases}$$

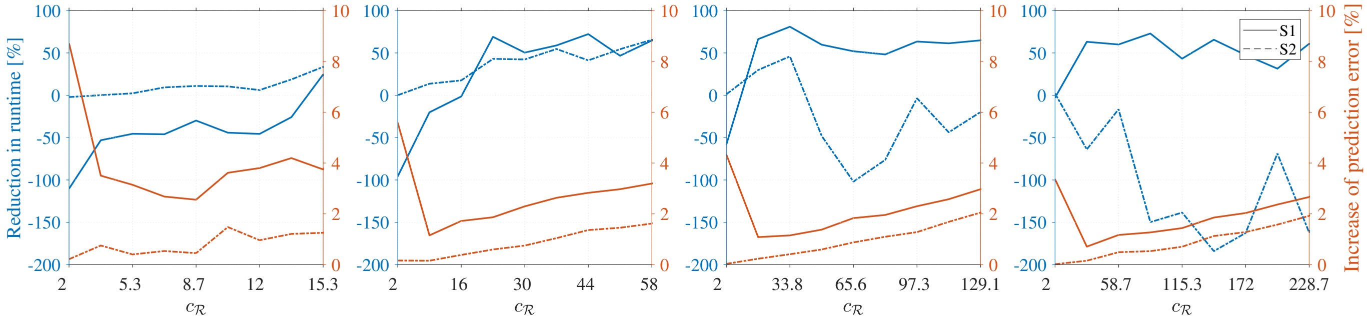
which produces the same effect as S1, but without having to split $c_{\mathcal{R}}$ up into the r sub-problems. This is precisely the reason why the minimum prediction error attained by S1 will never be smaller than that achieved by S2, and also the reason why S1 may be more efficient than S2.

Fig. 7 shows the performance of S1 and S2, expressed in relative terms with respect to the original optimization problem (17). We generated 100 instances of $\mathbf{X}$ and $\hat{\mathbf{X}}$ completely at random and computed the average runtime and prediction error for $d = 16$, $n = 8, 16, 24, 32$ and $r = 2$. Since we are assuming low reordering costs, the performance was assessed for values of $c_{\mathcal{R}}$ up to 1/5 of the maximum $F(\boldsymbol{\psi})$ (see Eq. (21)).

The results show that the proposed strategies may reduce the computational burden significantly, with the highest reduction being an 80% for $n = 24$ and $c_{\mathcal{R}} \simeq 33$. As for the differences between the two strategies, we note that S2 performed better than S1 in terms of runtime for $n = 8$, while the opposite was observed for $n = 24, 32$. An important consideration is that both S1 and S2 may exhibit, for certain values of n and $c_{\mathcal{R}}$, larger runtimes than those required to compute (17).

The results also seem to indicate that the price to pay is relatively small. In our experiments, the minimum error value was observed to be just 9% larger than that attained by the original problem. In short, although these results obviously depend on the data and thus we cannot draw conclusions on whether which strategy is more appropriate

[12] The problem has also been investigated in [40] where it is referred to as the *resource constraint minimum weight assignment problem.*



(a) $n = 8$.

(b) $n = 16$.

(c) $n = 24$.

(d) $n = 32$.

Fig. 7: Reduction in execution time and increase of the minimum prediction error provided by the two proposed strategies (S1 and S2), when the system is designed to operate at low permutation costs $c_{\mathcal{R}}$. The results have been obtained for different block lengths n and for $d = 16$ and $r = 2$.

for a given data, with them we show the potential benefits of operating at the region of low permutation costs.

4.6 Encoder Control

Coding efficiency describes the ability of a video codec to trade-off bit rate and reconstruction quality [43]. In video applications one typically wants the best possible reconstruction quality for a given available bit rate.

A multitude of parameters including coding modes and intra-prediction modes have to be selected on a per-block or per-frame basis. These selections determine the coding efficiency of a generated bitstream and are referred to as *encoder control*.

A larger set of coding and prediction modes is only advantageous in video coding if the reduction in bit rate that results from the improved prediction and transform coding outweighs the additional bit rate required for transmitting the selected modes to the decoder. In the case of database streams, we have an entirely analogous trade-off. Since such a selection is a data-dependent operation, re-distributing a fixed privacy budget to allow one more DP algorithm only makes sense if a larger set of prediction modes and types can effectively reduce the overall distortion.

We design the encoder control to decide, on a per *block basis*, the prediction mode (i.e., intra or inter) and the specific prediction type to be used (e.g., average vertical, horizontal). Consistently with the optimality criterion of the predictor module, we define the scoring function

$$\kappa(\mathbf{x}, \hat{\mathbf{x}}) = -\|\mathbf{x} - \hat{\mathbf{x}}(\phi)\|_{\mathrm{F}}^2 = -\|\mathbf{e}\|_{\mathrm{F}}^2.$$

Our next results computes the sensitivity of this function, which we shall use to design the exponential mechanism selecting the specific prediction mode and type.

Lemma 5 (Sensitivity of the scoring function for the selection of the encoding parameters)**.** *The L1-sensitivity of the scoring function* κ *is* $\mathrm{GS}(\kappa(\mathbf{x}, \hat{\mathbf{x}}(\phi))) = \|\Lambda\|_2^2$.

Proof: Let $\mathbf{x}$ and $\mathbf{x}'$ be two neighboring input blocks, and $x_r = (x_{r1}, \ldots, x_{rd})$ and $x'_r = (x'_{r1}, \ldots, x'_{rd})$ the records in which the two input blocks differ respectively. Direct application of the definition of L1-sensitivity leads to

$$\mathrm{GS}(\kappa(\mathbf{x}, \hat{\mathbf{x}}(\phi))) = \max_{\substack{\mathbf{x}, \mathbf{x}' \\ \hat{\mathbf{x}}, \phi}} \left| \sum_{ij} (x_{ij} - \hat{x}_{ij}(\phi))^2 - (x'_{ij} - \hat{x}_{ij}(\phi))^2 \right|.$$

Note that each of the summands above is maximized when $\hat{x}_{ij}(\phi) = x'_{ij}$, and that the minimum achievable value of each summand is zero. Accordingly,

$$\begin{aligned} \mathrm{GS}(\kappa(\mathbf{x}, \hat{\mathbf{x}}(\phi))) &= \max_{\mathbf{x}, \mathbf{x}'} \left| \sum_{ij} (x_{ij} - x'_{ij})^2 \right| \\ &= \max_{x_r, x'_r} \left| \sum_{j=1}^{d} (x_{rj} - x'_{rj})^2 \right|, \end{aligned}$$

where clearly the maximum is attained at the extreme values of x_r and x'_r. ■

Algorithm 2 shows how the modules preprocessing, encoder control and prediction interact to select, in a DP manner, a permutation ψ and a configuration ϕ that minimize the prediction error. Specifically, the predictor estimates $\hat{\mathbf{X}}$ for all possible configurations in line 1. All prediction blocks are then sent to the preprocessing module, which computes the permutations minimizing each of these blocks, as specified in (17) (lines 2 to 4). Lastly, the encoder control decides on the configuration of the predictor and the corresponding optimal permutation (line 5), which are conveyed to the predictor and the preprocessing modules, respectively.

By the sequential and parallel composition properties of DP, it is immediate to verify our DPCM-based protection method provides ε-DP frames, with $\varepsilon = \varepsilon_{Q_L} + \varepsilon_{Q_E} + \varepsilon_E$.

5 Experimental Evaluation

In this section, we evaluate experimentally the protection method proposed in Section 4. The aim of this section is to show that our approach, which builds on hybrid video encoding techniques to enhance data utility, may in fact diminish the amount of noise required to attain ε-DP. The empirical analysis provided in this section has been conducted in its entirety with Matlab 2019b, on a Ryzen 7 1800X at 4GHz.

5.1 Data Sets

To try to capture the voluminous and continuous characteristics of database streams, our experiments are targeted toward large datasets.

Our experimental evaluation will use two standardized data sets, known as "(Very) Large Census" and "Quant Forest", which are two of the largest datasets in the community

Algorithm 2: Preprocessing, encoder control and prediction.

```
Input: An input block $\mathbf{X}$; the reconstructed blocks
       $\tilde{\mathbf{X}}_{j-1}, \tilde{\mathbf{X}}_{j-2}, \ldots, \tilde{\mathbf{X}}_{j-\pi}$; the privacy
       parameter $\varepsilon_E$ of the exponential mechanism;
       the maximum desirable permutation cost $c_{\mathcal{R}}$
Output: A permutation $\psi$ and a prediction
        configuration $\phi$ satisfying both $\varepsilon_E$-DP
Compute $\hat{\mathbf{X}}(\phi)$ for all $\phi \in \Phi$
forall $\phi \in \Phi$ such that $\hat{\mathbf{X}}(\phi)$ has a least a
  non-constant column do
    Compute $\|\mathbf{E}(\phi)\|_{\mathrm{F}}^2$ as (17) and denote the
      minimizer by $\phi(\psi)$
end
Select $\phi(\psi)$ with probability proportional to
  $\exp\left(-\varepsilon_E \|\mathbf{E}(\phi)\|_{\mathrm{F}}^2 / 2\|\Lambda\|_2^2\right)$
return $\phi(\psi)$.
```

of statistical disclosure control. For brevity, we shall refer to them as `vlCensus` and `forest`, respectively.

The former data set contains 149 642 records and has 13 numerical attributes. It was previously documented and used in [44], [45], and has been chosen to adhere to the de facto convention in the area as well as for its large number of records.

The latter has 581 012 records and is based on the `Forest FCoverType` dataset available at the UCI KDD data repository [46]. Exactly as in [45], [47], we selected just the real-valued attributes, which reduced the number from 54 to 10, and for computational reasons we took the first 150 000 records. In our analysis, all attributes have been treated as quasi-identifiers and therefore all them have been the target of protection.

5.2 Baseline Method

As we mentioned in Sec. 2.2.2, only [10] has tackled the problem of publishing DP database streams in a continuous manner. However, since that work is limited to single-attribute databases, evaluating this protection method against ours is meaningless.

Consequently, we cannot but compare our solution just with the baseline approach described at the beginning of Sec. 4. The plain Laplace noise (PLN), as we shall call it, will add Laplace noise directly to the incoming records, without introducing delay nor reordering them. Although it is a rather naive strategy, it will allow us to assess the benefits of our method and derive worst-case bounds on distortion.

5.3 Configuration Parameters

Next, we specify the range of configuration parameters used in our experiments.

5.3.1 Coefficients Order

As explained in Sec.4.3.1, scans are designed following the empirical evidence that $\Pr\{C_{ij} > 0\}$ is typically decreasing with i and j. In our experiments we use the zig-zag scan and the diagonal orders shown in Fig. 3.

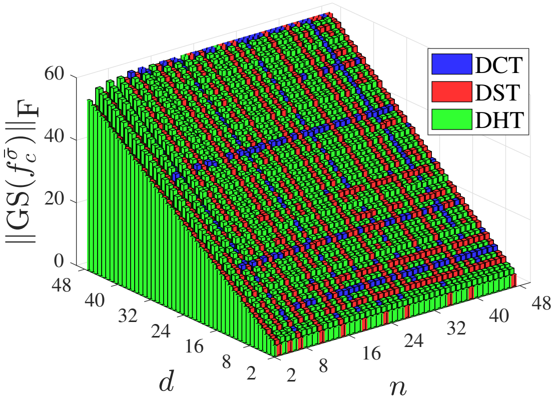


Fig. 8: Quantity $\|\mathrm{GS}(f_{\mathbf{c}}^{\bar{\sigma}})\|_{\mathrm{F}}$ for different block sizes and for the three transforms under study. From the figure, we note that this quantity does not vary significantly with the block size.

5.3.2 Intra-Prediction Functions

We use the intra-prediction functions specified in the video coding standards H.262 — MPEG-2 Video [48], H.263 [49] and MPEG-4 AVC [50]. This includes DC, horizontal, vertical and diagonal predictions types.

5.3.3 Block Sizes

In real practice, d will be given by the database stream to be protected, and thus is fixed, whereas n is a parameter of the scheme and needs to be chosen appropriately.

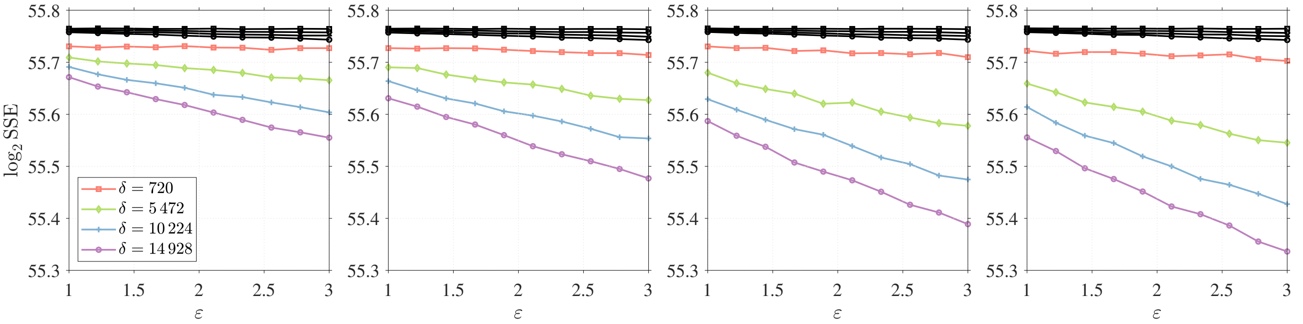
In Fig. 8, we have computed the quantity $\|\mathrm{GS}(f_{\mathbf{c}}^{\bar{\sigma}})\|_{\mathrm{F}}$ for different block sizes. This quantity is central to compute the sensitivities of the scoring functions $\theta = 1/2$ and $\theta = 1$, and therefore to choose t and σ; hence its importance.

The results have been obtained for $\Lambda = \mathbf{1}$, which is equivalent to dividing each attribute value by its maximum value and essentially indicate that the specific value of n will not have a large impact on the sensitivity of either scoring function. The number of attributes, however, does have a greater effect on $\|\mathrm{GS}(f_{\mathbf{c}}^{\bar{\sigma}})\|_{\mathrm{F}}$, and appears to be roughly linear with d.

It is worth emphasizing that the transforms shown for each block size are the ones maximizing the quantity at hand. In other words, they are the worst choice, among the three transforms under study, in terms of data distortion. However, as Fig. 8 shows, the differences in terms of $\|\mathrm{GS}(f_{\mathbf{c}}^{\bar{\sigma}})\|_{\mathrm{F}}$ among the DCT, DST and DHT are small.

Although n does not seem to have an impact on the sensitivities of the scoring functions $\theta = 1/2$ and $\theta = 1$, it does pose various trade-offs in our DPCM scheme. For example, the larger n, the larger the number of transformed coefficients, and the more Laplacian noise will be added to each of them, but the larger the coding efficiency of transform coding[13] Likewise, the smaller n, the less permutations will be available for the preprocessing module, and therefore the worse the prediction $\hat{\mathbf{X}}$ of $\mathbf{X}$. In order to capture the effect of n on the proposed scheme, our experiments will be conducted for block lengths of $8, 16, 32$ and 48 records.

[13] The coding efficiency of transform coding typically increases with the block size. Nonetheless, the potential gains may become insignificant beyond a certain block size [35].



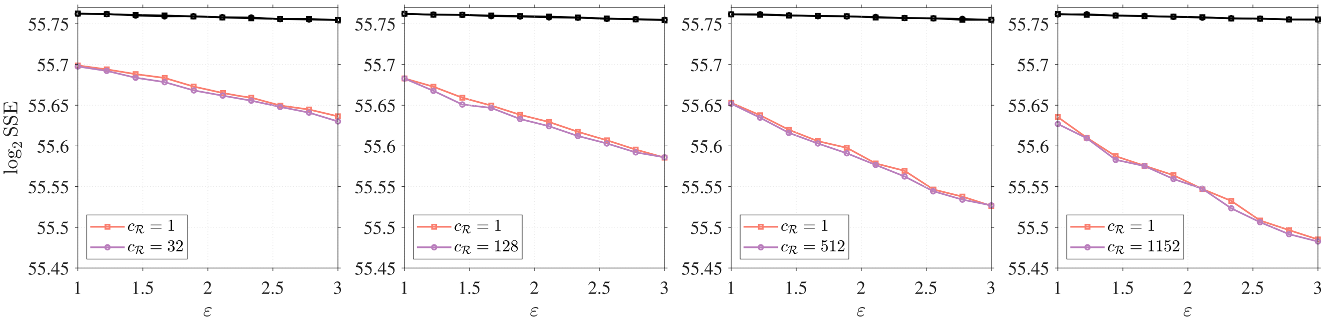
(a) $n = 8$, $c_{\mathcal{R}} = 16$.

(b) $n = 16$, $c_{\mathcal{R}} = 64$.

(c) $n = 32$, $c_{\mathcal{R}} = 256$.

(d) $n = 48$, $c_{\mathcal{R}} = 576$.

Fig. 9: Average distortion versus privacy protection for several values of record delay δ, block size n and maximum allowed reordering cost $c_{\mathcal{R}}$ in the data set `vlCensus`. The baseline approach and the proposed solution are represented with *black* and *coloured* lines and points, respectively.



(a) $n = 8$.

(b) $n = 16$.

(c) $n = 32$.

(d) $n = 48$.

Fig. 10: Average distortion versus privacy protection for a fixed delay $\delta = 7\,872$, several block sizes n and two allowed reordering costs $c_{\mathcal{R}} \in \{1, \lfloor n^2/2 \rfloor\}$ in the data set `vlCensus`. The baseline approach and the proposed solution are represented with *black* and *coloured* lines and points, respectively.

5.3.4 Preprocessing

In those cases when the processing module is to operate at the extreme regions of the prediction-reordering trade-off, we shall use the strategies described in Sec. 4.5.1 to alleviate the computational burden on the module. In the low-reordering case, we shall employ S1 for $n = 8, 16$ and S2 for $n = 32, 48$. In any case, we shall set a timeout of 2 seconds for the computation of either the original problem (17) or the strategies S1 and S2.

5.4 Distortion Metric and Privacy Parameters

We use the SSE to evaluate the impact on distortion caused by anonymization. The SSE is a measure of overall information loss that is frequently employed in the evaluation of statistical disclosure control methods.

On the other hand, we shall conduct our series of experiments for levels of privacy protection in the interval $\varepsilon \in [1, 3]$, which cover the usual range of values observed in the literature [51], [52], [15], [16], [17]. In this regard, we shall set $\varepsilon_{Q_L} = \varepsilon_{Q_E} = \varepsilon_E = \varepsilon_j/3$.

Lastly, note that the sensitivity values derived in Section 4 are essentially proportional to the length of the intervals in which these attributes take values. Since the attributes of our two data sets are not naturally upper-bounded, we need to delimit the domain of each attribute. For the sake of comparison, we follow the methodology described in [15], [53], [16], [54] and upper-bound the domain of an attribute to be 1.5 times the maximum value of this attribute in the data set.

5.5 Results

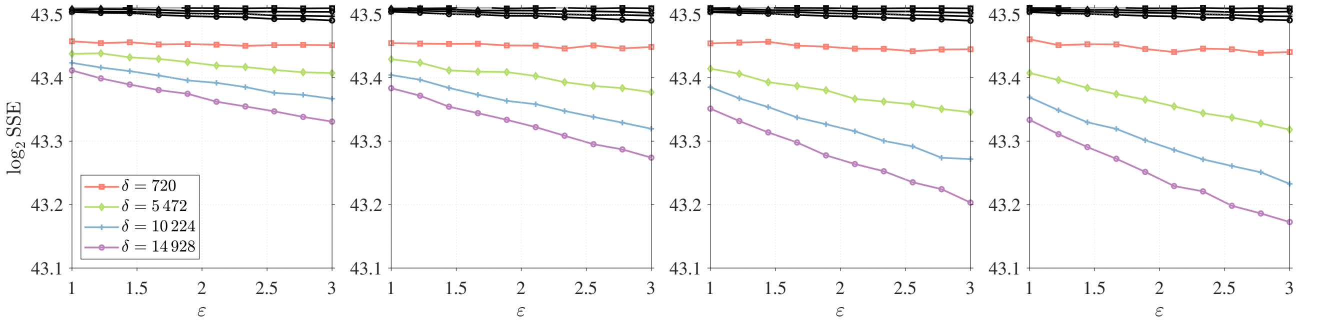
First of all, it should be noted that the series of experiments shown in the sequel have been conducted for the scoring functions $\omega_{1/2}$ and ω_1. However, since the observed differences are negligible, we just report on the results for one of them, namely, ω_1.

Fig. 9 shows average[14] distortion values for ten equally spaced values of ε within the interval $[1, 3]$. In our experiments, we set $\delta = m$, which means all records experienced a delay of m records, that is to say, a frame[15]; and evaluated the proposed system for four delay-constraint values (shown in the figure), which account for roughly 0.5%, 3.65%, 6.82% and 10% of the total length of the dataset. Furthermore, we allowed a reordering cost of half of the maximum acceptable cost, that is, $c_{\mathcal{R}} = \lfloor n^2/2 \rfloor/2$.

The log-distortion obviously decreases with ε, and does so in an almost linear way, both in our system (coloured lines) and in the baseline approach (black lines). In any of the four subfigures, Figs. 9(a-d), we can see that higher delays translate into lower distortion. This is not because there are more record blocks available for inter-prediction or block matching, as these parameters are fixed. This is simply because ε_j is larger, on account of the fact that $\varepsilon_j = \varepsilon\, m/l$ and $m = \delta$. Also, in the process of decreasing distortion, the

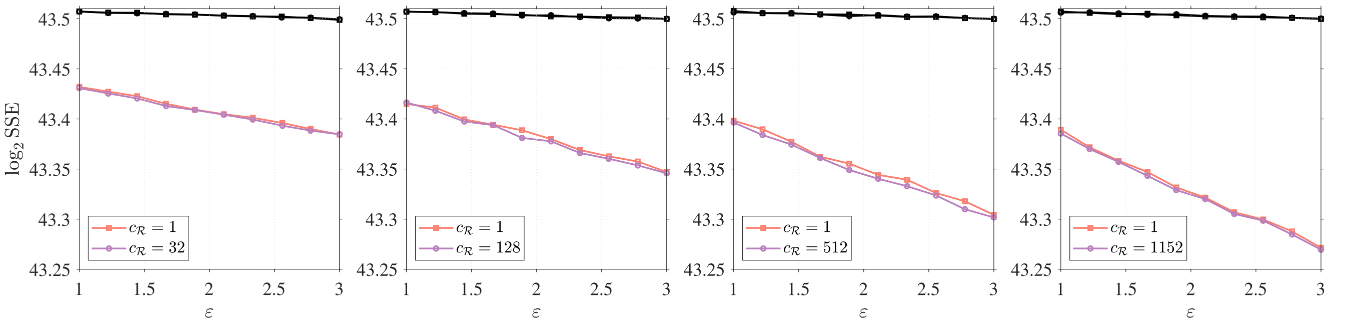
[14] Given the randomness of the DP mechanisms employed, we used one hundred repetitions for each combination of system parameters and averaged all them.

[15] Recall that m is the number of records within a *frame*.



(a) $n = 8$, $c_{\mathcal{R}} = 16$. (b) $n = 16$, $c_{\mathcal{R}} = 64$. (c) $n = 32$, $c_{\mathcal{R}} = 256$. (d) $n = 48$, $c_{\mathcal{R}} = 576$.

Fig. 11: Average distortion versus privacy protection for several values of record delay δ, block size n and maximum allowed reordering cost $c_{\mathcal{R}}$ in the data set `forest`. The baseline approach and the proposed solution are represented with *black* and *coloured* lines and points, respectively.



(a) $n = 8$. (b) $n = 16$. (c) $n = 32$. (d) $n = 48$.

Fig. 12: Average distortion versus privacy protection for a fixed delay $\delta = 7\,872$, several block sizes n and two allowed reordering costs $c_{\mathcal{R}} \in \{1, \lfloor n^2/2 \rfloor\}$ in the data set `forest`. The baseline approach and the proposed solution are represented with *black* and *coloured* lines and points, respectively.

effect of the delay is much more important in our system than in the baseline approach, essentially because the latter does not leverage the delay for anything else, other than increasing ε_j.

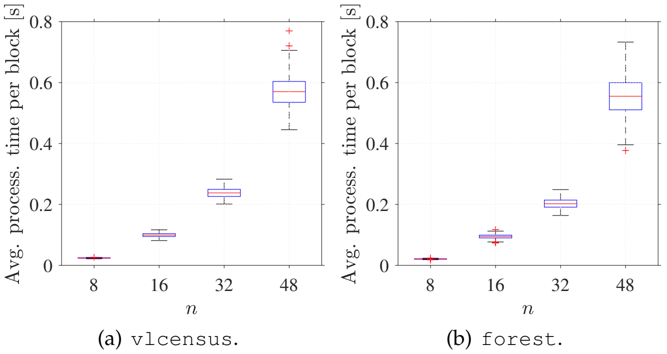
In comparative terms, it may seem that there is not a large difference in distortion between our solution and the baseline approach. However, indeed there is: a reduction of 0.3 or 0.4 in $\log_2$ SSE in fact represents a relative reduction of 23.11% or 31.95% in SSE. This is what we observe in Fig. 9(d): our approach yields 32% less distortion that the baseline solution for $\varepsilon = 3$, $n = 48$, $\delta = 14\,928$ and $c_{\mathcal{R}} = 576$. However, for the smallest delay value ($\delta = 720$), it appears that larger block sizes do not diminish distortion too much. It should be noted, though, that the observed gain margins are despite the low values of ε_j our system operates with, going approximately from 0.0048 (when $\delta = 720$ and $\varepsilon = 1$) to 0.2986 (when $\delta = 14\,928$ and $\varepsilon = 3$).

For a fixed delay, Fig. 9 shows how the distortion decreases with the pair $(n, c_{\mathcal{R}})$. This seems to indicate that the coding efficiency of transform coding increases with n (despite the fact that we may have potentially more coefficients and thus more noise added to them) and/or that a greater number of permutations available for the preprocessing module notably improves the prediction $\hat{\mathbf{X}}$ of $\mathbf{X}$.

Fig. 10 clarifies this latter point. Here we show the distortion for a fixed delay $m = 7\,872$ and two values of $c_{\mathcal{R}}$, namely, $c_{\mathcal{R}} = 1$ (no reordering allowed) and $c_{\mathcal{R}} = \lfloor n^2/2 \rfloor$ (no constraints on reordering). The results indicate that the gains due to allowing any reordering are not significant, which suggests that the block size has a greater impact on distortion. In short, it seems that, out of the main parameters controlling the trade-off among distortion, delay, reordering and privacy, n and δ have a greater effect on distortion than $c_{\mathcal{R}}$ —at least in this dataset.

Fig. 11 shows the same variables of Fig. 9 but for the data set `forest`. In general, we can observe a very similar behaviour than in `vlcensus`, including the same little impact of reordering on distortion reduction. There are some slight differences, however. First, the minimum difference in distortion between the baseline approach and our scheme, which is observed for $\delta = 720$, $n = 8$, $c_{\mathcal{R}} = 16$ and $\varepsilon = 1$), is 0.66%; while in `vlcensus` this yields 0.078%. And secondly, the maximum differences in distortion between our solution and the baseline approach are observed, analogously as in the data set `vlcensus`, in Fig. 11(d) and yield 31%.

Fig. 13 shows the average processing time per block we recorded in the computation of Figs. 9, 10, 11 and 12. We observe that the 75th percentile for `vlcensus` is 0.6038, 0.2495, 0.1039, 0.02456 seconds respectively for $n = 8, 16, 32, 48$ and 0.5997, 0.2141, 0.0991, 0.0214 seconds for `forest`. We also notice that the processing time is slightly greater for the `vlcensus` data set, which is consistent with its 3 additional numerical attributes. In this regard, we would like to emphasize that the efficiency of our method for large-dimensional stream databases (i.e., large d) will depend on the efficiency of the employed transforms. As a matter of fact, the computational burden on the analysis and synthesis



(a) vlcensus. (b) forest.

Fig. 13: Average processing time per block.

blocks represents, on average, the 41% of the time needed to protect a block.

6 Previous Work on Differentially-Private Transform Coding

As described in Secs. 2.2.2 and 5.2, the only work that has dealt with the problem of publishing DP database streams is [10]. Nonetheless, we could not compare our approach with that work experimentally since it just operates with single-attribute databases and does not allow record updates.

Although to the best of our knowledge there is just this work, the conceptual approach presented here, however, shares some similarities with two distinct protection methods, [55] and [56]. Although none of them are intended for database streams, for the sake of rigorousness we deem it appropriate to highlight the main differences between those two works and ours.

The former work, [55], aims to answer a fixed number of queries over time-series data under DP. To this end, the authors propose a protection method called sampling perturbation algorithm (SPA) that perturbs the one-dimensional discrete Fourier transform (DFT) of such query answers. In particular, the SPA chooses the number of such coefficients adaptively with the exponential mechanism by sampling a multidimensional hyperbolic distribution and then perturbing them. On the other hand, [56] aims to protect static histograms with DP. The proposed solution, called enhanced SPA (ESPA), essentially uses a different scoring function in the exponential mechanism of the SPA.

First and foremost, we would like to emphasize that SPA and ESPA are not aimed, nor can be trivially adapted, to database streams[16]. Secondly, their fundamental operation relies merely on a single, one-dimensional transform-coding scheme and the elimination of certain coefficients; but they do not address the problem through a hybrid video coding approach nor interprets the processing of those coefficients as a quantization step, nor considers prediction, encoding control or data permutations. Thirdly, [55] and [56] capitalize upon the DFT of the *original input data*, whereas our work operates with the two-dimensional DCT, DST and Hartley transforms of the *residual signal*. Fourthly, our approach distributes the privacy budget among the transformed coefficient in an optimal fashion, so as to minimize the MSE. Fifthly, we use a family of parameterized scoring functions in the exponential mechanism to select not only the number of transform coefficients but also the type of transform. Finally, our approach leverages the *exact* sensitivity of those coefficients, while SPA and ESPA operate with a sensitivity *bound*, whose mathematical derivation is flawed.

7 Conclusions and Future Research

With the advent of big-data analytics, complying with current data-protection frameworks in Europe and some Western countries has become very challenging. Our work focuses on the anonymization of database streams (a particular class of dynamic data), a technique whereby data controllers can legitimately circumvent such legal frameworks.

Among a variety of privacy notions, DP is one of the most popular among the scientific community working in data anonymization. In this work, we have tackled the protection of database streams with DP in the compelling case when the data controller wishes to publish those streams, rather than statistics derived from them.

We have proposed an anonymization method that can publish multiple numerical-attribute, finite database streams with DP guarantees and provide high protection as well as high utility in terms of data distortion, delay and record reordering.

The proposed method, which relies on the DPCM compression scheme, adapts techniques originally intended for hybrid video encoding, to favor and leverage dependencies among the blocks of the stream to be protected. In video coding, the exploitation of statistical dependencies can enhance coding efficiency and reduce the information contributed by image blocks and frames. In our context of database anonymization, we have shown the adapted techniques can help introduce significant less distortion.

We have designed our method to operate with blocks of records going through a series of modules analogous to those of the DPCM scheme, except for the preprocessing module. The design of our solution has been optimized in a number of different ways to minimize the MSE incurred in releasing the synthetized, protected block (instead of the original one).

Our extensive experimental evaluation demonstrates the suitability of utilizing hybrid video encoding to publish DP database streams. For the two data sets under study, we have shown our method can achieve a relative reduction of 32% and 31% less distortion in SSE than the baseline approach for the vlCensus and forest data sets, respectively. Remarkably enough, these results have been obtained for extremely low values of ε_j (i.e., for extremely high values of block protection), in the interval $[0.0048, 0.2986]$.

We have also observed that distortion decreases with the block size and the maximum acceptable reordering cost, which suggests that the coding efficiency of transform coding increases with the former parameter and/or that a larger number of permutations at preprocessing module significantly reduces the prediction error. Furthermore, our experimental results seem to confirm that the block size

[16] Note that [56] does not even address the case of continuous data.

and the delay have a greater effect on distortion than the maximum acceptable reordering cost.

With this work we have also shown the riveting interplay between the field of information privacy on the one hand, and on the other the fields of data compression and video encoding, while bridging the gap between the respective communities.

Finally, an interesting and necessary avenue for future research is to develop anonymization algorithms for database streams containing categorical attributes.

Acknowledgments

J. Parra-Arnau is the recipient of a Juan de la Cierva postdoctoral fellowship, IJCI-2016-28239, from the Spanish Ministry of Economy and Competitiveness, and an Alexander von Humboldt postdoctoral fellow. The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 100010434) and from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 847648. The fellowship code is LCF/BQ/PR20/11770009. This work was also supported by the Spanish Government under research project "Enhancing Communication Protocols with Machine Learning while Protecting Sensitive Data (COMPROMISE)" (PID2020-113795RB-C31/AEI/10.13039/501100011033), and by the German Federal Ministry of Education and Research through the project 16KIS1393K "PROPOLIS".

References

[1] S. P. Gangadharan, "Big data for the people: it's time to take it back from our tech overlords," May 2013, accessed on 2021-01-18. [Online]. Available: https://www.theguardian.com/sustainable-business/how-can-big-data-social-good

[2] S. Lohr, "Big data is opening doors, but maybe too many," Mar. 2013, accessed on 2021-01-18. [Online]. Available: https://www.nytimes.com/2013/03/24/technology/big-data-and-a-renewed-debate-over-privacy.html

[3] B. Tarnoff, "Big data for the people: it's time to take it back from our tech overlords," Mar. 2018, accessed on 2021-01-18. [Online]. Available: https://www.theguardian.com/technology/2018/mar/14/tech-big-data-capitalism-give-wealth-back-to-people

[4] S. Ovide, "Just collect less data, period." Jul. 2020, accessed on 2021-01-18. [Online]. Available: https://www.nytimes.com/2020/07/15/technology/just-collect-less-data-period.html

[5] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Lang., Program.* Springer-Verlag, 2006, pp. 1–12.

[6] J. Soria-Comas and J. Domingo-Ferrer, "Big data privacy: Challenges to privacy principles and models," *Data Sci. Eng.*, vol. 1, no. 1, pp. 21–28, 2016.

[7] B. Marr, "How much data do we create every day? the mind-blowing stats everyone should read," May 2018, accessed on 2021-09-04.

[8] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2094–2106, 2014.

[9] F. Fioretto and P. V. Hentenryck, "OptStream: Releasing time series privately," *J. Artif. Intell. Res.*, vol. 65, no. 1, pp. 423?–456, May 2019.

[10] B. C. Leal, I. C. Vidal, F. T. Brito, J. S. Nobre, and J. C. Machado, "δ-DOCA: Achieving privacy in data streams," in *Proc. Int. Workshop Data Priv. Manage. (DPM)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 11025, Barcelona, Spain, Sep. 2018, pp. 279–295.

[11] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, 2006, pp. 265–284.

[12] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 2339–2347. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999325.2999396

[13] G. Cormode, C. M. Procopiuc, D. Srivastava, and T. T. L. Tran, "Differentially private publication of sparse data," *CoRR*, vol. abs/1103.0825, 2011. [Online]. Available: http://arxiv.org/abs/1103.0825

[14] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: private data release via bayesian networks," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 1423–1434.

[15] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez, "Enhancing data utility in differential privacy via microaggregation-based k-anonymity," *VLDB J.*, vol. 23, no. 5, pp. 771–794, Oct. 2014.

[16] D. Sánchez, J. Domingo-Ferrer, S. Martínez, and J. Soria-Comas, "Utility-preserving differentially private data releases via individual ranking microaggregation," *Inform. Fusion*, vol. 30, pp. 1–14, 2016.

[17] J. Parra-Arnau, J. Domingo-Ferrer, and J. Soria-Comas, "Differentially private data publishing via cross-moment microaggregation,," *Inform. Fusion*, vol. 53, pp. 269–288, Jan. 2020.

[18] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 1, pp. 189–201, 2002.

[19] A. Trombetta, W. Jiang, and E. Bertino, *Privacy and Anonymity in Information Management Systems: New Techniques for New Practical Problems*. Springer-Verlag, 2011, ch. Advanced Privacy-Preserving Data Management and Analysis, pp. 7–30.

[20] H. Li, L. Xiong, X. Jiang, and J. Liu, "Differentially private histogram publication for dynamic datasets: An adaptive sampling approach," in *Proc. Int. Conf. Inform., Knowl. Manage. (CIKM)*. ACM, 2015, pp. 1001–1010.

[21] R. Chen, Y. Shen, and H. Jin, "Private analysis of infinite data streams via retroactive grouping," in *Proc. Int. Conf. Inform., Knowl. Manage. (CIKM)*. ACM, 2015, pp. 1061–1070.

[22] N. Li, W. Qardaji, and D. Su, "On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy," in *Proc. ACM Int. Symp. Inform. Comput. Commun. Secur. (AsiaCCS)*. ACM, 2012, pp. 32–33.

[23] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proc. ACM Int. Symp. Theory Comput. (STOC)*. ACM, 2010, pp. 715–724.

[24] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau, "Pegasus: Data-adaptive differentially private stream processing," in *Proc. ACM Conf. Comput., Commun. Secur. (CCS)*. ACM, 2017, p. 1375?1388.

[25] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *VLDB J.*, vol. 7, no. 12, p. 1155?1166, Aug. 2014.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[27] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. ACM, 2011, p. 193?204.

[28] J. Cao, B. Carminati, E. Ferrari, and K. Tan, "CASTLE: Continuously anonymizing data streams," *IEEE Trans. Depend., Secure Comput.*, vol. 99, 2009.

[29] F. D. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," ser. Proc. ACM SIGMOD Int. Conf. Manage. Data. ACM, 2009, pp. 19–30.

[30] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer Academic Publishers, 1992.

[31] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found., Trends Theor. Comput. Sci.*, vol. 9, no. 3.

[32] W. Chen, M. J. Er, and S. Wu, "PCA and LDA in DCT domain," *Pattern Recognit. Lett.*, vol. 26, pp. 2474–2482, 2005.

[33] R. Brand, J. Domingo-Ferrer, and J. M. Mateo-Sanz, "Computational Aspects of Statistical Confidentiality project," European project IST-2000-25069 CASC, Tech. Rep., 2001–2004. [Online]. Available: http://neon.vb.cbs.nl/casc/CASCtestsets.htm

[34] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. IEEE Annual Symp. Found. Comput. Sci. (FOCS)*. Washington, DC: IEEE Comput. Soc., 2007, pp. 94–103.

[35] T. Wiegand and H. Schwarz, *Source Coding: Part I of Fundamentals of Source and Video Coding*. San Diego, CA: Found. Trends., 2011, vol. 4, no. 1–2.

[36] J. Makhoul, "Linear prediction — A tutorial review," *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[37] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, no. 12, pp. 1799–1808, Dec. 1981.

[38] C. Spearman, "The proof and measurement of association between two things," *Amer. J. Psychol.*, vol. 15, pp. 88–103, 1904.

[39] J. B. Mazzola and A. W. Neebe, "Resource-constrained assignment scheduling," *Oper. Res.*, vol. 34, pp. 560–572, 1986.

[40] R. Aboudi and K. Jornsten, "Resource constrained assignment problems," *Discr. Appl. Math.*, vol. 26, pp. 175–191, 1990.

[41] P. Diaconis and R. Graham., "Spearman's footrule as a measure of disarray," *J. Royal Stat. Soc., Ser. B*, vol. 39, no. 2, pp. 262–268, 1977.

[42] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Soc. Ind., Appl. Math., 2009.

[43] A. C. Bovik, *Handbook of Image and Video Processing*, 2nd ed. Elsevier, 2005.

[44] J. Domingo-Ferrer, A. Martínez-Ballesté, J. M. Mateo-Sanz, and F. Sebé, "Efficient multivariate data-oriented microaggregation," *VLDB J.*, vol. 15, no. 4, pp. 355–369, 2006.

[45] V. M.-M. M. Solé and J. Nin, "Efficient microaggregation techniques for large numerical data volumes," *Int. J. Inform. Secur.*, vol. 11, p. 253?267, Apr. 2012.

[46] [Online]. Available: http://kdd.ics.uci.edu

[47] A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Parra-Arnau, and J. Forné, "The fast MDAV (F-MDAV) algorithm: An Algorithm for k-Anonymous Microaggregation in Big Data," *Eng. Appl. Artif. Intell.*, vol. 60, Apr.

[48] ITU-T and ISO/IEC, "Generic coding of moving pictures and associated audio information: Video. ITU-T Recommendation H.262 — ISO/IEC 13818-2," Tech. Rep., 2012.

[49] ITU-T, "Video coding for low bit rate communication. ITU-T Recommendation H.263," Tech. Rep., 2005.

[50] ITU-T and ISO/IEC, "Advanced video coding for generic audiovisual services. ITU-T Recommendation H.264 — ISO/IEC 14496-10," Tech. Rep., 2014.

[51] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc., Data Min. (KDD)*. ACM, 2010, pp. 503–512.

[52] J. Lee and C. Clifton, "How much is enough? choosing ϵ for differential privacy," in *Proc. Int. Inform. Secur. (ISC)*. Springer-Verlag, 2011, pp. 325–340.

[53] D. Sánchez, J. Domingo-Ferrer, and S. Martínez, "Improving the utility of differential privacy via univariate microaggregation," in *Priv. Stat. Databases (PSD)*, vol. 8744. Springer-Verlag, 2014, pp. 130–142.

[54] J. Soria-Comas and J. Domingo-Ferrer, "Differentially private data sets based on microaggregation and record perturbation," in *Proc. Int. Conf. Model. Decisions Artif. Intell.*, Oct. 2017, pp. 119–131.

[55] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. ACM, Jun. 2010, pp. 735–746.

[56] G. Acs, C. Castelluccia, and R. Chen, "Differentially private histogram publishing through lossy compression," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2012, pp. 1–10.