This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

DPcode: Privacy-preserving Frequent Visual Patterns Publication on Cloud

Zhan Qin, *Student Member, IEEE,* Kui Ren, *Fellow, IEEE,* Ting Yu, *Member, IEEE*
Jian Weng *Member, IEEE*

Abstract

Nowadays, cloud has become a promising multimedia data processing and sharing platform. Many institutes and companies plan to outsource and share their large-scale video and image datasets on cloud for scientific research and public interest. Among various video applications, the discovery of frequent visual patterns over graphical data is an exploratory and important technique. However, the privacy concerns over leakage of sensitive information contained in the videos/images impedes the further implementation. Although the Frequent Visual Patterns Mining (FVPM) algorithm aggregates summary over individual frames and seems not to pose privacy threat, the private information contained in individual frames still may be leaked from the statistical result. In this paper, we study the problem of privacy-preserving publishing of graphical data FVPM on cloud. We propose the first differentially private frequent visual patterns mining algorithm for graphical data, named DPcode. We propose a novel mechanism that integrates the privacy-preserving visual word conversion with the differentially private mechanism under the noise allocation strategy of sparse vector technique. The optimized algorithms properly allocate the privacy budgets among different phases in FPM algorithm over images and reduce the corresponding data distortion. Extensive experiments are conducted based on datasets commonly used in visual mining algorithms. The results show that our approach achieves high utility while satisfying practical privacy requirement.

Index Terms

cloud, data publication, differential privacy, frequent pattern mining.

I. Introduction

Frequent Pattern Mining (FPM) is one important and fundamental concept in data mining [1]. In FPM, the patterns are defined as local data structures that capture correlations among a number of attributes or variables in a dataset (e.g., itemsets in transactions data [3], subimages in graphical data [2]). The goal of FPM is to discover patterns that appear frequently. Among many data mining tasks and theories stem from the concept of FPM, Frequent Visual Pattern Mining (FVPM) is adopted for graphical dataset, which also serves as an essential building block for many compelling applications (e.g., trajectory tracking and

Z. Qin, and K. Ren are with SUNY at Buffalo, USA (email: zhanqin@buffalo.edu, kuiren@buffalo.edu)

T. Yu is with Qatar Computing Research Institute, Doha, Qatar (email: tyu@qf.org.qa)

J. Weng is with Jinan University, Guangzhou, China (email: cryptjweng@gmail.com)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

behavior identification etc.) on graphical data from different fields like surveillance [33], meteorology [5], social network [6]. However, the graphical data that contains valuable information is usually sensitive and inappropriate to share on cloud. For example, sociologists may conduct promising research from a video surveillance database. Meanwhile, the people's identity and location information could also be leaked from the publication of these videos [4].

A few preliminary works are proposed targeting at solving privacy-preserving FPM problem instead of FVPM problem [8]. However, none of these techniques can be adapted for FVPM problem. The patterns in FVPM are composed of detected image features, i.e., visual primitives, instead of item sets in FPM. These visual primitives consist of real-valued data with high-dimensional attributes in contrast to transaction items in FPM, which is set-valued data with limited dimensional attributes. Obviously, it is more challenging to mine frequent patterns over real-valued data [19]. Feasible solutions require to convert a graphical dataset into a set-valued dataset. And one popular technique is to utilize the Bag-of-words (BoW) model, which helps convert real-valued visual primitives into set-valued visual words with respect to a pre-defined codebook through clustering [16]. In other words, image/video data can be effectively converted into transaction dataset through the BoW model. Now, to achieve differential privacy, the straightforward solution is to apply existing private FPM algorithms over the converted visual data. However, the high-dimensionality of visual words (millions) and the small size of frequent patterns (dozens or hundreds) in graphical data make it very sensitive to the information loss induced by the dimensionality reduction techniques, which are commonly used in existing FPM algorithms. Thus, the straightforward solutions inevitably lead to poor system utility. For example, the solutions proposed in [10], [11], both project the data into a lower-dimension space to reduce the dimensionality. The utility of mining results, however, are substantially affected, as the data are distorted in the dimensionality reduction procedures. On the other side, resorting to the Exponential mechanism will always lead to a very high complexity, i.e., traversing the output space with exponential complexity. Compared with Exponential mechanism, Laplace mechanism is a most common way that requires less computational complexity: It only needs adding noise to output ($O(1)$) or intermediates ($O(n)$), while Exponential mechanism requires traversing the whole output space, whose complexity is at least $O(2^M)$. Here, n denotes the number of items and M is the size of results. In a nutshell, it remains a challenge to design a differentially private FVPM algorithm with desirable system utility.

In this paper, we propose DPcode, a differentially private FVPM mechanism tailored for graphical data to address the above challenges. DPcode achieves both desirable utility and efficiency. To mitigate data distortion, it introduces randomness into the BoW model during the conversion as opposed to after the conversion in existing solutions. A spatial Laplace mechanism and a fine-grained truncation algorithm are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

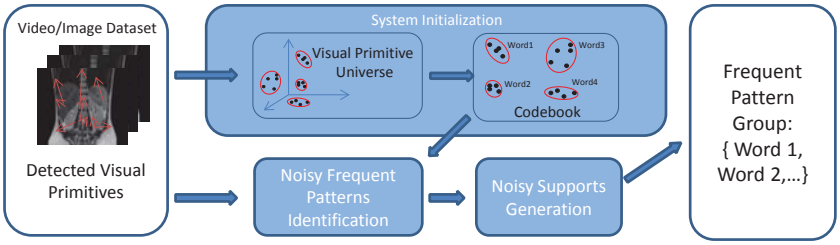


Fig. 1. **The system workflow of DPcode**

further proposed to effectively calibrate the noise perturbed into the visual primitives in the conversion procedure with $O(n)$ complexity. DPcode then utilizes a FP-tree structure to effectively minimize system sensitivity. The resulted privacy budget is further distributed properly between these two phases to achieve differential privacy with improved utility.

The contributions of DPcode are summarized as follows:

1. We present the design of DPcode, a differentially private algorithm for FVPM with high utility. It integrates the conversion from visual primitive universe to visual word space (codebook) with the differentially private mechanism to achieve low data distortion. Meanwhile, it also uses the low sensitivity of frequency/support query in FP-tree structure to enforce differential privacy.

2. We give a rigorous analysis and a theoretical bound on privacy and utility of the proposed mechanism. A formal proof according to the definition of differential privacy is provided. Meanwhile, the probability bound of utility is also modeled and proved.

3. We conduct extensive experiments on the utility of the proposed algorithms. A number of benchmark graphical datasets are utilized to study the privacy and utility effectiveness of the mechanism. DPcode shows a better utility performance than other state-of-the-art techniques of privacy-preserving FPM algorithms.

The rest of this paper is organized as follows: The next section gives a brief introduction to the preliminaries. Section III presents the design of DPcode. The analysis of the privacy of the proposed mechanism in described in Section IV. Finally, we present related work and conclude in Sections VI and VII, respectively.

II. The Preliminaries

A. Frequent Visual Pattern Mining and Notations

FVPM aims at discovering visual patterns that frequently appear in graphical data. Denote $D = \{\mathcal{I}_1, ..., \mathcal{I}_s\}$ as a graphical data, where $\mathcal{I}_i$ represents a frame and $|D| = s$ is the number of frames in D. A FVPM algorithm is performed as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

Firstly, a feature detection algorithm is performed to detect visual primitives in each frame. A frame $\mathcal{I}_i$ can be characterized by its visual primitives, i.e., $\mathcal{I}_i = \{v_1, ..., v_{n_i}\}$, where n_i is the number of visual primitives contained in frame $\mathcal{I}_i$ and $v \in \mathbf{V}$. A visual primitive v is a d-dimensional feature vector, which describes a visual feature in an image/video dataset. Normally an image/frame may contain hundreds of such visual primitives based on the size of the data. In this paper, we apply the SIFT feature vector [14] as the visual primitives, which is one of the most popular local features in image processing area.

After detecting all visual primitives, all of them are converted into codewords based on the BoW model for further utilization in image matching and classification. Each codeword or visual word represents the existence of a class of visual primitives. All these visual words together form a codebook, i.e., $\Omega = \{W_1, ..., W_M\}$, where W_j represents a visual word and M is the size of the codebook. Each visual word has a corresponding clustering center w_j in primitive universe $\mathbf{V}$. A codebook can be considered as a quantization function that converts real-valued high-dimensional visual primitives into low dimensional visual words. It is usually generated from a benchmark dataset regarding a specific type of graphical data. In this paper, we utilize a data independent codebook generation approach from [19]. According to the codebook Ω, for a visual primitive v, it is converted into word W_j, which has the nearest visual word clustering center w_j to v in primitive universe. Each frame $\mathcal{I}_i$ in a video can be converted into a set of visual words as $\mathcal{T}_i \subseteq \Omega$. A word-based graphical dataset $T = \{\mathcal{T}_1, ..., \mathcal{T}_s\}$ is converted from D.

Finally, a FPM algorithm can be performed over the converted graphical data T. Formally, a visual pattern is defined as a set of visual words from codebook, i.e., $\mathcal{P} \subseteq \Omega$. The support of $\mathcal{P}$, denoted as $\rho(\mathcal{P})$, is the number of frames in D that contains $\mathcal{P}$ as a subimage. The targets of FVPM algorithm is to output frequent patterns $\{\mathcal{P}\}_k$ and the corresponding supports, if its support is larger than a threshold ρ_k, or top-k frequent patterns. One form of FVPM problem can be converted into another form easily.

B. Differential Privacy

Differential privacy bounds the 'influence' that any individual presence in a dataset has on the distribution of the published statistical result [9]. Define D and D' as two neighboring datasets if one of them can be obtained by adding or removing one entry (individual record) to another.

Definition 1. *(ϵ-differential privacy): A randomized mechanism $\mathcal{M}$ is ϵ-differentially private if for all neighboring datasets D and D', $\forall S \subseteq Range(\mathcal{M})$:*

$$Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in S]$$

The parameter ϵ is called 'privacy budget', which indicates the privacy leakage from each answered query. A smaller ϵ means a stronger privacy guarantee the mechanism $\mathcal{M}$ can provide. In other words, ϵ represents the strength of privacy. A common technique to achieve ϵ-differential privacy is the perturbation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

of Laplace noise to the query results. It is much simpler to implement compared with other mechanisms (e.g., Exponential mechanism requires enumerating all possible outputs). Intuitively, the magnitude of added noise should be large enough to hide the influence that is made by any possible individual record in the dataset. This idea is reflected by the concept of the $sensitivity$ of the query function, which is defined as the maximum difference between the results of the query function on any pair of neighboring datasets: $\triangle f = max_{D,D'} \|f(D) - f(D')\|$.

One common approach to achieve ϵ-differential privacy is through the Laplace mechanism [17]. The statistical result of a query function f is perturbed by adding noise from the Laplace distribution with probability density function $pdf(x) = \frac{1}{2b}e^{-\frac{|x|}{b}}$, where $b = \frac{\triangle f}{\epsilon}$. The probabilistic mechanism $\mathcal{M}$ satisfies ϵ-differential privacy:

$$\mathcal{M}_{Laplace}(D) = f(D) + Lap(\frac{\triangle f}{\epsilon})$$

where $Lap(b)$ denotes a random variable sampled from the Laplace distribution with scale parameter b.

III. DPcode Design

DPcode publishes frequent visual patterns together with their supports (frequencies) for graphical data, while achieving differential privacy. It takes an integer k and a graphical dataset D as inputs and outputs the top-k most frequent visual patterns with their supports. To achieve differential privacy, a Laplace mechanism based design is proposed. Moreover, based on the property of query for FVPM, we distribute the privacy budget into two different phases of DPcode and achieve an improved utility performance. The privacy budget allocating strategy follows the idea behind sparse vector technique [20]. The system allocates the privacy budget between the set of frequent patterns $\{\mathcal{P}\}_k$ and noisy supports of these frequent patterns, instead of infrequent patterns with supports below the threshold. The system's workflow is divided into three phases: System Initialization, Noisy Frequent Patterns Identification, and Noisy Supports Generation, as shown in Fig. 1.

A. Noisy Frequent Patterns Identification

In the first phase of differentially privacy enforcing mechanism in DPcode, instead of directly adding noise to the supports, we introduce Laplace noise into visual word conversion procedure to get noisy word-based graphical dataset, and then use it to identify frequent patterns.

1) Noise Calibration: We first show the noise calibration for frequent pattern set. FVPM algorithm first compares the support of each pattern with the threshold to get a set of frequent patterns. We model these identified frequent patterns through a set of queries shown below (Maximal frequent patterns can be easily deduced from frequent patterns by eliminating overlapped patterns):

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

Definition 2. *(frequent pattern queries): The result of a set of frequent pattern query is an answer vector, which is composed of a set of query answers for all candidate patterns from dataset:*

$$FPQ = \{q_1, ..., q_{|C|}\}$$

where $q_i = 1$ *if* $\rho(\mathcal{P}_i) \geq \rho_k$*, otherwise* 0*, and* C *is the set of candidate patterns.*

It is easy to find that the sensitivity of a set of frequent pattern queries is very high: $\triangle FPQ = |C| - k$, where $k \ll |C|$. The proof is simple: Assume a frame/image contains every pattern except the frequent patterns, i.e., $|C| - k$ "infrequent" patterns, and the support of every infrequent pattern is $\rho_k - 1$. Hence, the involvement of this frame/image will change $|C| - k$ query answers by 1 at most. That is, the sensitivity of query set is $|C| - k$. Since the size of candidate pattern set is very large, i.e., $|C| \approx 2^M$, a straightforward implementation of the Laplace mechanism to frequent patterns identification will introduce undesirable noise into the result.Intuitively, to achieve differential privacy, all results disclosed should be computed through differentially private procedures.

Based on the sparse vector technique, the noise in the frequent pattern identification comes from the noise in both the threshold support and the supports of candidate patterns (sequential composition). The analysis of the noise perturbed to the threshold support ρ_k is simple. Intuitively, adding or removing a frame in a video can only change the threshold by at most 1, i.e., $\triangle f = 1$. Hence, based on the Laplace mechanism described previously, the noise added to the threshold is supposed to be $Lap(\frac{1}{\epsilon})$. The algorithm for phase 1 starts by calculating the noisy threshold $\hat{\rho}_k = \rho_k + Lap(\frac{2}{\epsilon_1})$ with half of the privacy budget. After that, the noise introduced to every support of each candidate pattern leads to sever data distortion: Though the sensitivity of each candidate pattern is 1, which is relatively small in differential privacy implementations, there are $|C|$ candidate patterns. Hence, if we assume the privacy budget for each candidate pattern is ϵ, the accumulated total privacy budget is $|C| \cdot \epsilon$, which will lead to an unacceptable total privacy budget to the whole system.

To solve this problem, we resort to the following observation on the visual words (and the corresponding visual primitives) that are infrequent. That is, while having little impact on identifying frequent patterns, they contribute significantly to data distortion. On one side, the information leakage is mostly due to those frequent visual words as they are the ones being released with overwhelming probability. On the other side, the infrequent visual words have significantly smaller chances to be released in outputs. This very observation leads to the idea of dividing the candidate group into two sets, i.e., a frequent pattern set and an infrequent pattern set, and treat them differently: For frequent pattern set, we map the differential private noises over pattern supports into the visual primitive space and limit the visual primitive perturbation only to the set of frequent visual words $\{W\}_f$. That is, only visual primitives of frequent visual patterns/words will be perturbed with random noise. For infrequent pattern set, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

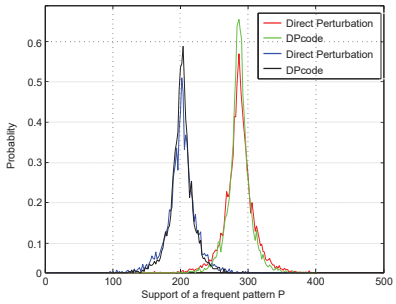


Fig. 2. **The distribution of noisy supports of DPcode in comparison with direct perturbation method**

still directly add noise to the support of each pattern for efficient implementation. We call the resulted algorithm a truncation algorithm as will be elaborated in Sec. III-A2 and III-A3. Note that the set of frequent visual word $\{W\}_f$ can be easily obtained by modifying a non-private FVPM algorithm [18]. $\{W\}_f$ can be obtained through a preprocessing procedure during system initialization and used as system parameters. The experimental results show that after a careful truncation processing, DPcode can achieve much improved system utility.

2) Frequent Pattern Set Perturbation: The advantage of mapping the noise from visual word to visual primitive space over the straightforward approach is based on the following observation in FVPM problem: If a visual primitive (and the corresponding visual word) is "frequent" in a video/image dataset, there are usually multiple similar visual primitives (which map to the same visual word) appear in the same frame. This is due to the symmetrical characteristic and self-repetitiveness of the image features. A "frequent" visual primitive usually means an effective description of a feature point in an image. And these feature points are often symmetric or repeat multiple times around the characteristic/interesting area in frames. The statistics in benchmark databases support such an observation [29], [30], [31], [32]. In this case, compared to directly adding noise to the count of visual words, we can improve the accuracy by perturbing noise to frequent visual primitives in visual primitive universe with relative low data distortion. Consequently, the corresponding distribution of visual word's (and pattern's) noisy support will be different from directly adding noise to it. The specific improvements depend on the graphical characteristics of graphical data, including the correlation between image features and their distributions in frames. For illustration, Fig. 2 shows the distributions of the noisy supports resulted by our method and a direct perturbation method (directly adding noise to support): We select the most frequent visual word from two typical benchmark databases: Caltech [29] and SCR [31] and simulate their noisy supports distributions through both methods ($\epsilon_1 = 1$).

Now we analyze how to allocate privacy budget to each frame. Assume that after getting the threshold

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

support ρ_k, candidate pattern set C is divided into two sets: frequent pattern set $\mathbf{S} = \{\mathcal{P} \mid \rho(\mathcal{P}) \geq \rho_{\mathbf{k}}\}$ and infrequent pattern set $\bar{\mathbf{S}}$ that contains rest patterns. The visual words in $\mathbf{S}$ are defined as frequent visual words and forms a set of visual words $\{W\}_f$. For frequent pattern set $\mathbf{S}$, we can consider the support of candidate patterns as aggregation results of counts of frequent visual words in frames. Assume each frame can be represented as $\mathcal{T}_i = \{x_1, ..., x_{|\{W\}_f|}\}$, where $x_j = 1$, if W_j exists in frame $\mathcal{I}_i$, otherwise $x_j = 0$. Thus, assume the privacy budget for each frequent visual word in $\mathcal{T}_i$ is ϵ, the accumulated privacy budget for each frame is $|\{W\}_f|\epsilon$. Moreover, we utilize the infinite divisibility of Laplace distribution [22]: a random variable with such distribution can be computed by aggregating s random variables.

$$Lap(b) = \sqrt{B_{s-1}} \sum_{i=1}^{s} Lap(b) \tag{1}$$

where B_{s-1} is a random variable drawn from Beta distribution with parameters 1 and $(s-1)$, and all frames use the same value. Hence, each frame only needs to be perturbed with a share of noise, such that the aggregated noise follows the Laplace distribution, which is enough to guarantee differential privacy: Assume that in phase 1, the privacy budget for frequent pattern set is $\epsilon_{1,f}$. For each visual word, the perturbed noise should be $\frac{1}{\sqrt{B_{s-1}}} Lap(\frac{2|\{W\}_f|}{\epsilon_{1,f}})$. Following the reasoning in [17], [23], a truncated distributed Laplace mechanism achieves the differential privacy. It is easy to observe that if we truncate the perturbed element as $\hat{x}_j = 1$, if $\hat{x}_j > 0.5$, otherwise $\hat{x}_j = 0$, it still satisfies differential privacy.

The detailed deduction of the frequent pattern set perturbation in primitive space is shown as follows: Assume the distance between the clustering center w_i and a visual primitive v is r, where the cluster radius of w_i is r_i, and the maximum cluster radius is r_{max}. The areas above 1 and below 0 can be truncated. Mapping the noise added on the existence of a visual word ($x_j = 0$ or 1) in a frame to the spatial location of a visual primitive in universe $\mathbf{V}$, a simple analysis suggests that the generalized scale factor to map the noise perturbed to the location of a primitive point is $2r_{max}$.

Before showing how to generate noisy visual primitive in universe $\mathbf{V}$ with high dimensionality, we first describe its simple version on a 2-dimensional universe, which utilizes the planar Laplace distribution. Given the privacy budget ϵ_1 for this phase, and the actual location $l = (x, y)$ in primitive universe, the density function of the differentially private approximate location $\hat{l} = (\hat{x}, \hat{y})$ in primitive universe should be:

$$pdf(\hat{l}) = \frac{\epsilon^2}{2\pi} e^{-\epsilon_1 \sqrt{(\hat{x}-x)^2 + (\hat{y}-y)^2}} \tag{2}$$

where $\frac{\epsilon^2}{2\pi}$ is a normalization factor. The above function of $\hat{l}$ is a Laplace distribution centered at l in a 2-dimensional space.

According to the definition of multivariate Laplace distribution, the projection of a planar Laplace distribution on any vertical plane passing by the center gives a scaled linear Laplace distribution, and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

corresponding mechanism satisfies differential privacy [24]. There are several definitions of multivariate Laplace distribution. We adopt the one used for bivariate Laplace distribution in [26] and expend it to multivariate according to [22], which is applicable for implementation. It is easy to see that the probability density in Laplace distribution only depends on its distance from l. Hence, for a clear and convenient expression, we convert the coordinates of primitive universe into a polar coordinate system with the origin in l.

In polar coordinates, a visual primitive at location $\hat{l}$ can be represented by a vector $(r, \theta_1, ..., \theta_{d-1})$, where r is the distance from the origin l to $\hat{l}$, and θ_i is the angle of elevation of the line in the cartesian coordinate system for a d-dimensional visual primitive universe. According the converting equations, the symmetric multivariate Laplace distribution in the polar coordinates centered at the origin l is:

$$pdf(\hat{l}) = pdf(r, \{\theta_i\}) \propto \frac{\epsilon^d}{2^{d-1}\pi^{\frac{d-1}{2}}\Gamma(\frac{d+1}{2})} r^{\frac{d}{2}} e^{-\epsilon r} \tag{3}$$

where $d \geq 2$ and Γ is a Gamma function. Since the multivariate that represent the radius and angles are independent, its distribution can also be expressed as the product of the multiple variates.

We can deduce the marginal pdf of these variates, according to radius r and angles θ:

$$pdf_r(r) = \epsilon^d r^{\frac{d}{2}} e^{-\epsilon r}, \quad pdf_\theta(\theta_i) = \frac{1}{2\pi} \tag{4}$$

and it can be easily deduced that

$$pdf(r, \{\theta_i\}) = \alpha \cdot \beta \cdot pdf_r(r) \prod_{1}^{d-1} pdf_\theta(\theta_i) \tag{5}$$

where α is a normalization factor in multivariate Laplace distribution, and the scale factor $\beta = \frac{2r_{max}}{\sqrt{B_{s-1}}}$. Since the density of each angle is constant, it is generated by drawing as a random number in the interval $[0, 2\pi)$ with uniform distribution.

Following Alg. 1, for each visual primitive, we first convert it into the corresponding visual word W_i based on the codebook Ω. If the corresponding visual word is frequent, i.e., $W_i \in \{W\}_f$, the truncation algorithm is performed. Otherwise, it is mapped to an infrequent visual word as any non-private mapping algorithm. In truncation algorithm, the visual primitive is first perturbed by adding Laplace noise to its original location in primitive universe. And then, it is mapped to a visual word. if the noisy visual word is a frequent visual word, i.e., $W_i \in \{W\}_f$, it is returned as truncation result. Otherwise, the truncation algorithm returns $\perp$. On the one side, the truncation algorithm truncates the area outside frequent visual word clusters by setting the corresponding probabilities to be 0. On the other side, the probabilities inside the clusters of frequent visual words still follows Laplace distribution. In other words, this truncation algorithm can be considered as a spatial version of truncated Laplace mechanism that only adds noise to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

Algorithm 1 Noisy Visual Word Conversion

Input: Graphical data $D = \{\mathcal{I}_1, ..., \mathcal{I}_s\}$, codebook $\Omega = \{W_i..., W_M\}$, frequent visual word set $\{W\}_f$, privacy budget $\epsilon_{1,f}$;
Output: Word-based graphical data $T = \{\mathcal{T}_i\}_{i=1}^{s}$;

$T \leftarrow \emptyset$
for each $\mathcal{I}_i \in D$ **do**
  $\mathcal{T}_i \leftarrow \emptyset$
  for each $v \in \mathcal{I}_i$ **do**
    $W_i \leftarrow Mapping(v, \mathcal{T}_i, \Omega)$
    if $W_i \in \{W\}_f$ **then**
      $W_j \leftarrow Truncation(v, \Omega, \frac{\epsilon_1}{2|\{W\}_f|}, \{W\}_f, \mathcal{T}_i)$
    else $W_j \leftarrow W_i$
    $\mathcal{T}_i \leftarrow \mathcal{T}_i \cup \{W_j\}$
  Insert $\mathcal{T}_i$ **to** T
return T
function $Mapping(v, \mathcal{T}_i, \Omega)$
  for $j = 1$ to M **do**
    if $dist(v, w_j) < r_j$ **and** $\mathcal{T}_i \cap \{W_j\} = \emptyset$ **then**
      $u \leftarrow W_j$
    else $u \leftarrow \perp$
return u
function $Truncation(v, \Omega, \epsilon, \{W\}_f, \mathcal{T}_i)$
    $v' \leftarrow PrimitivePerturbation(v, \Omega, \epsilon)$
    $W_i \leftarrow Mapping(v', \mathcal{T}_i, \Omega)$
    if $W_i \notin \{W\}_f$ **then**
      $W_j \leftarrow \perp$
    else $W_j \leftarrow W_i$
return W_j
function $PrimitivePerturbation(v, \Omega, \epsilon)$
  for $i = 1$ **to** $d - 1$
    $pdf(r, \{\theta_i\}) = \alpha \cdot \beta \cdot \epsilon^d r^{\frac{d}{2}} e^{-\epsilon r} \prod_1^{d-1} \frac{1}{2\pi}$
  locate v' from v and $(r, \{\theta_i\})$
return v'

frequent visual words. Apparently, the algorithm achieves differential privacy regarding frequent visual words.

3) Infrequent Pattern Set Perturbation: After converting graphical data D into word-based graphical data T, the system can employ an efficient non-private FPM algorithm to get noisy supports of frequent patterns, i.e. the frequent pattern set $\mathbf{S}$.

However, there is still one issue: the supports of infrequent patterns, i.e. patterns that contain infrequent visual words, does not achieve differential privacy. Note that the output of this phase is not the noisy supports but the existence of certain pattern in frequent pattern set. Hence, for an efficient implementation, we directly perturb noise to the support of infrequent patterns in $\bar{\mathbf{S}}$. Notice that the infrequent visual patterns have significantly smaller chances to be released in outputs. Assume the privacy budget for infrequent patter set is $\epsilon_{1,\bar{f}}$. For each pattern in the infrequent pattern set, the system computes the noisy

Algorithm 2 Noisy FP-tree Generation

Input: A perturbed word-based graphical data T, privacy budget ϵ;
Output: FP-tree t_{fp};

```
1: t_fp ← root node
2: for each pattern i ∈ 2^||P|| do
3:     k ← t_fp
4:     for each word j ∈ i do
5:         if ∄n ∈ k.child ∩ n.item = j then
6:             create n under k
7:             n.item ← j, n.count ← Lap(1/ε)
8:         k ← n
9: return t_fp
```

support by perturbing Laplace noise to its support:

$$\hat{\rho}(\mathcal{P}) = \rho(\mathcal{P}) + Lap(2\frac{|C| - k}{\epsilon_{1,\bar{f}}}) \tag{6}$$

where $\mathcal{P} \in \bar{\mathbf{S}}$. After that, the system can use the noisy threshold $\hat{\rho}_k$ to get a set of frequent patterns $\{\mathcal{P}\}_f$ from $\mathbf{S} \cup \mathbf{S}'$, where $\mathbf{S}' = \{\mathcal{P} \in \bar{\mathbf{S}} \mid \hat{\rho}(\mathcal{P}) \geq \rho_{\mathbf{k}}\}$.

B. Noisy Supports Generation

In this phase, a modified FP-growth algorithm is performed to derive noisy supports for the frequent pattern mining results [1]. Note that maximal frequent patterns $\{\mathcal{P}\}_M$ can be easily deduced from a set of frequent patterns $\{\mathcal{P}\}_f$ by eliminating overlapped patterns Compared with Apriori algorithm it is much faster with only two passes over the whole database. It first compresses the input database T creating an FP-tree instance to represent frequent patterns. After that, it divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately.

1) Noisy FP-tree Generation: Given a graphical data T, it is easy to use FP-tree construction algorithm to generate the frequent pattern tree of the database. This can be performed by following the intuition of FP-growth algorithm by scanning the database and collecting the set of frequent items attached with the corresponding supports, as shown in Alg. 2. Note that a node in the FP-tree has three attributes: $.item$, $.count$, and $.child$, where $item$ denotes which item represents, $count$ is the number of patterns represented by the path from the root to the node, and $child$ is an array of child nodes. The detailed algorithm is shown in Algorithm 2. If a node has a child such that $.item = child.item$, then increment $.count$ by 1; otherwise create a new node with initialized $.count = Lap(\frac{k}{\epsilon_2})$. Then the system can sort the collected set in descending order to get the perturbed noisy threshold of k-th frequent pattern ρ_k. By performing this algorithm, a FP-tree can be constructed in two scans of the whole database. In the first

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

round, the frequent items can be collected and sorted, and the top-k frequent patterns can be found. In the second round, the FP-tree is constructed.

2) Noisy Supports Generation: Utilizing the noisy FP-tree generation as a building block, the system performs Alg. 3 to generate noisy supports of frequent patterns. The algorithm takes the database T, phase 2 privacy budget ϵ_2 and maximal frequent patterns $\mathcal{P}_M$ as inputs and outputs a complete set of frequent patterns with the corresponding noisy supports. As shown in Alg. 3, if the FP-tree contains a single prefix path, then the system first mines single prefix path FP-tree: for each combination of the nodes in the single prefix-path part of the tree, the pattern with support equals the minimum support of nodes in the node combination is generated. If the tree contains no single prefix path, the system mines the multipath FP-tree: for each node n in the multipath tree, it inserts n into the patterns with the support of n. Then it constructs the conditional pattern-base of pattern $\mathcal{P}$ and its conditional FP-tree by calling function FP-tree generation with input $\mathcal{P}$.

We can calculate the noisy support of any frequent patterns $\mathcal{P}$ by counting the FP-trees that contains it and aggregating them together: $n_i.count + \sum n_i.child.count$. Start from the leaf nodes going up toward the root node, the count of each node is added to its parent to get the noisy count. The aggregation procedure is equivalent to constructing a tree from the differentially private set of maximal frequent patterns, then filling in supports using a noisy count query for each node. When the FP-tree contains a single prefix-path, it is generated by the enumerations of the corresponding subpaths that have the minimum support. Then, the multipath can be generated for each combination of the nodes in the path. After that, the combined results can be returned as the frequent patterns by generating the combination of all of the above results.

As the results of DPcode, the frequent patterns with the corresponding support is the output of the system and satisfies differential privacy. The detailed analysis of its privacy is shown in the next section.

IV. Privacy Analysis

We first show both phases of differentially private enforcing mechanism in DPcode are differentially private. After that, we will use the sequential composition property of differential privacy to prove that the whole mechanism achieves the differential privacy guarantee. Due to the space limitation, we only describe the basic idea and reasoning behind the security proof in this paper. The full version of paper can be found in the project website [21].

Based on the proposed method, the privacy budget ϵ is allocated between two phases. In phase 1, the outputs are a noisy k-th most frequent pattern's support ρ_k and the corresponding maximal frequent patterns $\{\mathcal{P}\}_M$. The randomness in these outputs comes from the noise perturbed to the supports of

Algorithm 3 Noisy Supports Generation

Input: Maximal frequent patterns $\{\mathcal{P}\}_M$, phase 2 privacy budget ϵ_2, word-based graphical data T, Top k;
Output: Top-k frequent patterns $\{\mathcal{P}\}_k$;

for each pattern $\mathcal{P} \in \{\mathcal{P}\}_M$ **do**
  $t_{fp} \leftarrow$ Noisy FP-Tree Generation($\mathcal{P}$, $\frac{\epsilon_2}{k}$)
  for each $\mathcal{T}_i \in T$ **do**
    $\mathcal{T}_i \leftarrow \mathcal{T}_i \cap \mathcal{P}$, $t \leftarrow t_{fp}$
    for each word $j \in \mathcal{T}_i$ **do**
      for each t's child n **do**
        if $\exists n.item = j$ **then**
          $t \leftarrow n,\ \ t.count = t.count + 1$
  for each node $n \in t_{fp}$ **do**
    $n.count \leftarrow \sum n.child.count + n.count$
  $\{\mathcal{P}\} \leftarrow$ *FP-growth*(t_{fp}, $\emptyset$);
$\{\mathcal{P}\}_k \leftarrow$ Top-k frequent patterns in $\{\mathcal{P}\}$
return $\{\mathcal{P}\}_k$
function *FP-growth*(t_{fp}, α)
  if t_{fp} contains a single path S **then**
    for each node combination $\beta \in S$ **do**
      $(\alpha \bigcup \beta).count \leftarrow$ min($n.count$) where $n \in \alpha$
  else
    for each $n \in$ multipah M **do**
      $(n \bigcup \alpha).count \leftarrow n.count$
      construct β's conditional FP-tree r_β
      if $r_\beta \neq \emptyset$ **then**
        FP-growth($r_\beta, beta$)
return $S \cup M \cup (S \times M)$

candidate patterns: noise introduced in visual word conversion, and the noise directly added to the threshold support of infrequent patterns. Here, we need to notice that the parallel composition theorem is not applied to the proposed mechanism, since the partition of two groups of candidate patterns depends on the visual primitive itself. However, the both mechanism on two groups of candidate patterns follows Laplace mechanisms. It is not difficult to find that the information leakage from the outputs still follows the definition of ϵ-differential privacy, as shown in theorem 3 (formal proof in [21]).

Theorem 1. *The mechanism $\mathcal{M}_1$ that generates ρ_k and $\{\mathcal{P}\}_M$ is ϵ_1-differentially private.*

Now we analyze the privacy of the second phase. The mechanism of phase 2 that outputs top-k frequent patterns $\{\mathcal{P}\}_k$ and its corresponding supports is denoted as $\mathcal{M}_2$. The generation of the FP-tree can be viewed as a query function $q(D)$ which returns an FP-tree built from database D. The count of the corresponding pattern is contained in the nodes of FP-tree. Hence, assume a vector $\mathbf{x}$= $\{x_1, ..., x_h\}$, where x_i represents the count of a node and h is the number of nodes in the tree. We will show that it suffices to prove that the function $n.child.count \leftarrow Lap(\frac{k}{\epsilon_2})$ satisfies differential privacy since this is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

the only place in the algorithm that requires access to T.

Theorem 2. *The mechanism $\mathcal{M}_2$ that generates top-k frequent patterns $\{\mathcal{P}_k\} and supports is ϵ_2-differentially private.*

$$\frac{Pr(\mathcal{M}_2(T) = \delta)}{Pr(\mathcal{M}_2(T') = \delta)} \leq e^{\epsilon_2} \tag{7}$$

where $\mathcal{P}_k$ and $\mathcal{P'}_k$ are neighboring frequent pattern that are different in one pattern, and $\delta \subseteq Range(\rho)$.

Proof. Since the count of one node is updated for each $\mathcal{T}_i$ in T in the algorithm. It is easy to see that changing a visual word in $\mathcal{T}_i$ or an entry in T can only affect the count of a node in FP-tree by at most 1. Assume the pattern $\mathcal{P}$ corresponding to x_i is removed from T. It will only decreases x_i by 1, while other nodes remains the same. Hence, for all T and its neighboring database T', The sensitivity of query function q is 1. According to Alg. 3, the system adds independent Laplace noise drawn from $Lap(\frac{k}{\epsilon_2})$ to each pattern achieves $\frac{\epsilon_2}{k}$-differential privacy. Hence, we can deduce that the k outputs achieve ϵ_2-differential privacy according to the composition theorem. □

Altogether, we can deduce that the DPcode mechanism achieves differential privacy according to the above theorems and sequential composition property of differential privacy mechanism:

Theorem 3. *The* DPcode *is ϵ_M-differentially private.*

where the total privacy budget ϵ_M is the sum of privacy budget ϵ_1 and ϵ_2 for phase 1 and 2.

V. Evaluation

A. Experiment Setup

We compare the performance of Dpcode with three existing private FPM algorithms over a set of video datasets: PrivBasis (PB) [10], SmartTruncation (ST) [11], and NoisyCut (NC) [12]. All experiments are conducted on an Intel Core i5 3.3GHz desktop with 32GB of physical memory. To obtain an average performance estimate, each algorithm is performed 30 times. Without loss of generality, we perform the original non-private algorithms to generate codebook with the same clustering configurations. The corresponding parameters for different algorithms are set to be comparable. The minimum support for ST is set to the support of the k-th most frequent pattern. We evaluate the utility of DPcode over a variety of benchmark video datasets, which are described as follows: 50 Salads dataset [25] contains 25 people preparing 2 mixed salads each and contains over four hours of annotated accelerometer and RGB-D video data. HMDB51 dataset [27] contains a total of 7,000 clips distributed in 51 action classes. The experiment utilizes 20,000 random frames drawn from 30 videos (15 videos per dataset). In addition, we also conduct experiments over professional image dataset, i.e., SCR dataset [31], which contains 247 PA chest radiographs. The images were scanned from films with a size of 2048 by 2048 pixels. We deliberately choose these three datasets to reflect the utility performance of the proposed mechanism in different types of visual datasets (normal, motion, and professional image datasets), so that the performance evaluation could be a more general. Note that the features that describe background is firstly detected in initialization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

Database	#img	Max$\|\mathbf{v}\|$	Avg$\|\mathbf{v}\|$
SALAD [25]	10,000	348	214
HMDB [27]	10,000	431	326
SCR [31]	247	6451	5681

TABLE I
DATASET DESCRIPTION

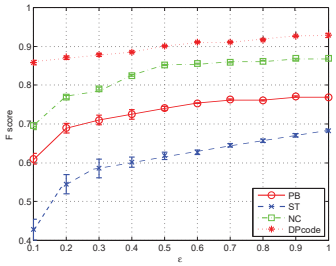


Fig. 3. F-score in SALAD

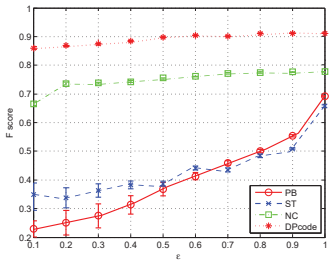


Fig. 4. F-score in HMDB

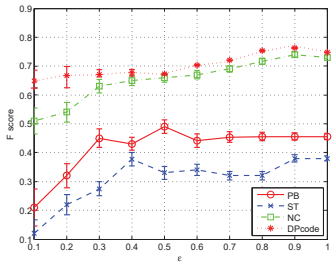


Fig. 5. F-score in SCR

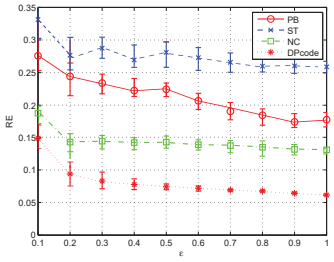


Fig. 6. RE in SALAD

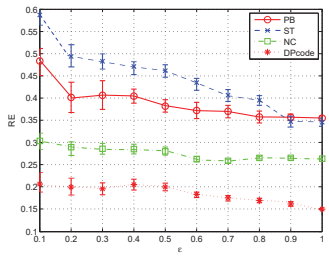


Fig. 7. RE in HMDB

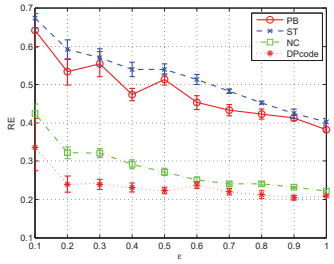


Fig. 8. RE in SCR

process and omitted in the rest of experiments. The detected frequent visual pattern is a building block for the corresponding utilization in applications like motion analysis etc. Note that, the characteristics of the databases are summarized in Table 1, where #img is the number of images in the dataset, Max$|\mathbf{v}|$ is the maximum number of SIFT features in an image and Avg$|\mathbf{v}|$ is the average number.

B. Performance Evaluation

1) Evaluation Metric: To compare the performance of algorithms, we evaluate the utility by employing $\mathcal{F}$-score and Relative Error (RE) as utility metrics [12].

Definition 3. *($\mathcal{F}$-score). Let $\mathcal{F}$ and $\hat{\mathcal{F}}$ be the set of correct and published frequent pattern, respectively. The $\mathcal{F}$ score is defined as:*

$$\mathcal{F}\ score = 2 \times \frac{precision * recall}{precision + recall} \tag{8}$$

where $precision = \frac{|\{\mathcal{F} \bigcap \hat{\mathcal{F}}\}|}{|\hat{\mathcal{F}}|}$, $recall = \frac{|\{\hat{\mathcal{F}} \bigcap \mathcal{F}\}|}{|\hat{\mathcal{F}}|}$.

For top-k FPM algorithms $\|\mathcal{F}\| = \|\hat{\mathcal{F}}\| = k$, hence $precision = recall = \mathcal{F} = 1 - FNR$, where FNR is the fraction of false negatives in the released frequent patterns, used in [10] for utility measure.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

Definition 4. *(Relative Error). The relative error of published frequent pattern $\hat{\mathcal{F}}$ is defined as*

$$RE = median_{\mathcal{P}\in\hat{\mathcal{F}}}\frac{\hat{\rho}(\mathcal{P}) - \rho(\mathcal{P})}{\rho(\mathcal{P})} \tag{9}$$

Notice that NC, PB, and ST algorithms have various parameter that controls the tradeoff between information loss and sensitivity reduction or data distortion. Their performance heavily depends on the proper setting of these parameters. However, there is no systematic way to estimate the optimal parameter settings. We will compare the maximal accuracy that can be obtained by our implementation or the results from the experimental results shown in the corresponding papers.

2) Experimental Result: Fig. 3 to Fig. 8 show the $\mathcal{F}$ scores of each algorithm by different values of ϵ when $k = 100$. Note that, the experiments with different k values are also conducted ($k = 1, 5, 10, 50, 100, 200$). The results show that as k increases, the utility performance drops. Following parameter settings in existing works, the utility curves with $k = 100$ is selected to be presented and analyzed in this section due to the space limit. The proposed DPcode consistently outperforms NC, PB and ST algorithms and shows stable performance for different privacy budget. When the ϵ is small, which means a stronger privacy guarantee, we observe significant advantage of our algorithm. For all of the datasets, the proposed system achieves the best performance of F-score and RE among all four algorithms. For datasets with relatively fewer visual primitives, the performance of the proposed system is closer to the existing work. When the visual primitives contained in each frame increases (HMDB) the utility of most existing works becomes unacceptable (F-score is lower than 0.35; RE is higher than 0.42). Meanwhile, the performance of DPcode is still stable and achieves high utility (F-score is higher than 0.64; RE is lower than 0.32). Moreover, the effect of adjusting weight parameter in allocating privacy budget between two phases is analyzed through experiments: When weight is small, less privacy budget is allocated to noisy frequent patterns identification phase, vice versa. It can be observed that the accuracy of the system can be effectively improved if the weight parameter is tuned well. Basically, allocating the privacy budget mainly on the first phase ($60\% - 70\%$) can achieve the best utility performance in DPcode. This is because when weight is relatively high, more privacy budget is allocated to the noisy visual word conversion. It could effectively generate a more 'comprehensive' noisy threshold and spend the rest privacy budget on noisy supports generation. However, if we spend all privacy budget on the first phase, then the utility of the proposed algorithm would be very poor. Since the privacy budget is uniformly distributed to all visual primitives in this phase.

VI. Related Work

FPM and its FVPM extensions [1], [34], [18] have been extensively studied in data mining community. It enables extracting informative and potentially interesting patterns from massive datasets by mining

meaningful patterns either through post-processing the FPM results or proposing new data mining criteria, including mining compressed patterns [35], approximate patterns [28] etc.

The privacy concerns that data mining results may lead to privacy leakage has been studied in a few works. Before the proposing of differential privacy [9], one popular solution is k-anonymity, which proposes to hide mining result to prevent inference attack on sensitive information [36]. However, it suffers to the modeling of the background knowledge of the adversary and vulnerable to inference attacks [34]. More recent work has focused on specific applications that achieve differential privacy, such as regression analysis [37] etc. However, this kind of technique significantly distorts the patterns distribution and is specialized for particular computations (e.g., mean), which makes it hard to be utilized for more complicated problem like FPM.

The problem of differentially privately releasing top k frequent itemsets of length m is first studied in [7]. Given the mining result of a non-private algorithm, it selects top k itemsets to release using traditional Laplace or exponential mechanism and adds noise to each itemset's support value. In [10], the authors propose a technique utilize the concept called basis. The itemsets in transaction database are viewed as tabular data and projected onto each basis. However, both two techniques may be very inaccurate and tend to generate more data distortion than practical implementation can tolerate. In [13], the process of frequent graph mining is integrated into a Markov Chain Monte Carlo framework to improve its efficiency under the consideration of the structural information of graph data. In [12], Apriori and FP-growth algorithm [20] are utilized to generate both noisy data and noisy threshold for FPM problem. However, the proposed method may lead to inaccurate results for small frequent pattern sizes. These existing data perturbation algorithms are restricted to the property of set-valued database, which can be considered to be treating the FPM problem as an inferior version of multiple counting problem and is hard to be generalized for real-valued database. In addition, these algorithms utilize dimensionality reduction based methods and have a substantial cost in the utility of mining results. However, there has been little work on differentially private data mining for graphical data. The complexity of graphical data structure and various corresponding mining algorithms impedes the adapting of existing techniques.

VII. Conclusion and future work

In this paper, we have presented a novel mechanism DPcode for differentially private mining frequent patterns publication for cloud-based graphical data . The proposed solution effectively integrate the process of the BoW model in FVPM, spatial Laplace noise perturbation, and FP-tree structure. The data distortion in the result of a differentially private FVPM algorithm is effectively reduced by the proposed mechanism. Moreover, the privacy budget is carefully balanced between the two phases: noisy frequent patterns

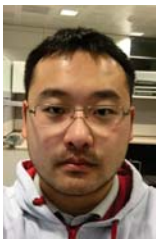
identification and noisy supports generation, in differential private mechanism to improve the accuracy of the results. In addition, we have also established the theoretical privacy guarantee and utility bounds of the proposed algorithms. The experiments on three visual datasets show a competitive accuracy performance with reasonable privacy budget. The comparison with existing techniques show that the proposed system outperforms the current state of the art. As a promising research direction, we will continue to investigate the privacy-preserving mechanisms for other important video data mining techniques, including secure video processing, sensitive contents detection, and trajectory detection etc.

REFERENCES

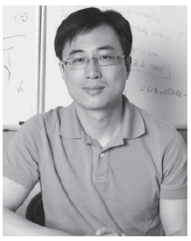
[1] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of ACM SIGMOD*, 2000.
[2] T. Washio, and H. Motoda, "State of the art of graph-based data mining," in *Proceedings of ACM SIGKDD*, 2003.
[3] J. Han, and M. Kamber, *Data Mining, Southeast Asia Edition: Concepts and Techniques*. Morgan kaufmann, 2006.
[4] T. M. Lehmann, et al. "Automatic categorization of medical images for content-based retrieval and data mining," *Computerized Medical Imaging and Graphics*, vol. 29, no. 2, pp. 143–155, 2005.
[5] P. Punitha, and D. Guru, "An effective and efficient exact match retrieval scheme for symbolic image database systems based on spatial reasoning: A logarithmic search time approach," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 10, pp. 1368–1381, 2006.
[6] J. Sabater and C. Sierra, "Reputation and social network analysis in multi-agent systems," in *Proceedings of ACM AAMAs*, 2002.
[7] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in *Proceedings of ACM SIGKDD*. 2010.
[8] M. Kantarciolu, J. Jin, and C. Clifton, "When do data mining results violate privacy?" in *Proceedings of ACM SIGKDD*, 2004.
[9] C. Dwork, "Differential privacy," in *Automata, languages and programming*. Springer, 2006, pp. 1–12.
[10] N. Li, W. Qardaji, D. Su, and J. Cao, "Privbasis: frequent itemset mining with differential privacy," in *Proceedings of IEEE VLDB Endowment*, 2012.
[11] C. Zeng, J. F. Naughton, and J.-Y. Cai, "On differentially private frequent itemset mining," in *Proceedings of the VLDB Endowment*, 2012.
[12] J. Lee and C. W. Clifton, "Top-k frequent itemsets via differentially private fp-trees," in *Proceedings of ACM SIGKDD*, 2014.
[13] E. Shen and T. Yu, "Mining frequent graph patterns with differential privacy," in *Proceedings of ACM SIGKDD*, 2013.
[14] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proceedings of IEEE CVPR*, 2004.
[15] C. B. Akgül, et. al. "Content-based image retrieval in radiology: current status and future directions," *Journal of Digital Imaging*, pp. 208–222, 2011.
[16] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, 2004.
[17] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proceedings of IEEE FOCS*, 2010.
[18] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007.
[19] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.
[20] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *Proceedings of IEEE FOCS*, 2010.
[21] Z. Qin, et al. "DPcode: Mining Frequent Visual Patterns with differential Privacy Guarantee," http://ubisec.cse.buffalo.edu/files/dpcode.pdf.
[22] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace distribution and generalizations: a revisit with applications to communications, exonomics, engineering, and finance*. Springer Science & Business Media, 2001.
[23] S. Goryczka, L. Xiong, and V. Sunderam, "Secure multiparty aggregation with differential privacy: a comparative study," in *Proceedings of ACM Joint EDBT/ICDT 2013 Workshops*, pp. 155–163, 2013.
[24] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, Springer, pp. 265–284, 2006.
[25] S. Stein, and S. J. McKenna, "Combining Embedded Accelerometers with Computer Vision for Recognizing Food Preparation Activities" in *Proceedings of ACM UbiComp*, 2013.
[26] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of ACM CCS*, 2013.
[27] H. Kuehne, et al. "HMDB: A Large Video Database for Human Motion Recognition." in *Proceedings of ICCV*, 2011.
[28] J. Liu, S. Paulsen, X. Sun, W. Wang, A. B. Nobel, and J. Prins, "Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis." in *Proceedings of SIAM SDM*, 2006.
[29] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVIU*, vol. 106, no. 1, pp. 59–70, 2007.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

[30] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proceedings of CVPR*. 2006.
[31] B. van Ginneken, M. Stegmann, and M. Loog, "Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database," *Medical Image Analysis*, vol. 10, no. 1, pp. 19–40, 2006.
[32] M. Heath, K. Bowyer, D. Kopans, R. Moore, and P. Kegelmeyer, "The digital database for screening mammography," in *Proceedings of the 5th international workshop on digital mammography*. Citeseer, pp. 212–218, 2000.
[33] U. Turdukulov, et al. "Visual mining of moving flock patterns in large spatio-temporal data sets using a frequent pattern approach." in *International Journal of Geographical Information Science*. pp. 2013–2029, 2014.
[34] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of ACM SIGMOD*. 1993.
[35] T. Calders and B. Goethals, "Depth-first non-derivable itemset mining." in *Proceedings of SIAM SDM*, 2005.
[36] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.
[37] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Advances in Neural Information Processing Systems*, pp. 289–296, 2009.



Zhan Qin is currently working toward his Ph.D. degree at the director of the Ubiquitous Security and Privacy Research Laboratory (UbiSeC) in the Computer Science and Engineering Department of the State University of New York at Buffalo. His research interests are in the areas of cloud computing and security, with focus on differential privacy data collection and publication, cybersecurity in smart grid. He is a student member of the IEEE, IEEE COMSOC, and ACM.



Kui Ren received his B.Eng. (first class honours) and M.Phil. degrees from the University of Hong Kong, and M.S. and Ph.D. degrees from Courant Institute of Mathematical Sciences, New York University. He is now a research fellow in the Department of Combinatorics & Optimization, University of Waterloo. During his doctoral study, he has interned at NTT Research and Development (Tokyo), Microsoft Research (Redmond) and Fuji Xerox Palo Alto Laboratory, and visited the University of Texas at Austin, Massachusetts Institute of Technology and Queensland University of Technology, resulted not only in research publications at major conferences such as ACM Conference on Computer and Communications Security and IACR Conference on Public-Key Cryptography, but also US patent applications. His research interests are applied cryptography, privacy and distributed systems security in general. He has been serving on the program committees of several international conferences on these topics such as Asiacrypt 2012 and ACNS 2012.

Ting Yu received the BS degree from Peking University in 1997, the MS degree from the University of Minnesota in 1998, and the PhD degree from the University of Illinois at Urbana-Champaign in 2003, all in computer science. He is currently a senior scientist in Qatar Computing Research Institute. His research interests include security, with a focus on data security and privacy, trust management, and security policies. He is a recipient of the US National Science Foundation (NSF) CAREER Award. He is a member of the IEEE.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2535729, IEEE Transactions on Multimedia

Jian Weng received the M.S. and B.S. degrees in computer science and engineering from South China University of Technology, in 2004 and 2000, respectively, and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, in 2008. From April 2008 to March 2010, he was a postdoc in the School of Information Systems, Singapore Management University. Currently, he is a professor and vice dean with the School of Information Technology, Jinan University. He has published more than 40 papers in cryptography conferences and journals, such as PKC, CT-RSA, ACSAC, SCN, Designs, Codes and Cryptography, Algorithmica, etc. He served as PC cochairs or PC member for more than 10 international conferences, such as ISPEC 2011, RFIDsec 2013 Asia, ISC 2011, IWSEC 2012, etc.