Privacy models for big data: a survey

Nancy Victor* and Daphne Lopez

School of Information Technology and Engineering,
VIT University,
Vellore, India
Email: nancyvictor@vit.ac.in
Email: daphnelopez@vit.ac.in
*Corresponding author

Jemal H. Abawajy

Faculty of Science, Engineering and Built Environment,
Deakin University,
Melbourne, Australia
Email: jemal@deakin.edu.au

Abstract: Big data is the next big thing in computing. As this data cannot be processed using traditional systems, it poses numerous challenges to the research community. Privacy is one of the important concerns with data, be it traditional data or big data. This paper gives an overview of big data, the challenges with big data and the privacy preserving data sharing and publishing scenario. We focus on the various privacy models that can be extended to big data domain. A description of each privacy model with its benefits and drawbacks is discussed in the review. This survey will contribute much to the benefit of researchers and industry players in uncovering the critical areas of big data privacy.

Keywords: anonymisation; big data challenges; big data characteristics; big data privacy; big data; differential privacy; *k*-anonymity; *l*-diversity; privacy models; privacy preserving data publishing; social network graph; streaming data; *t*-closeness.

Reference to this paper should be made as follows: Victor, N., Lopez, D. and Abawajy, J.H. (2016) 'Privacy models for big data: a survey', *Int. J. Big Data Intelligence*, Vol. 3, No. 1, pp.61–75.

Biographical notes: Nancy Victor is currently working as an Assistant Professor at VIT University, India. She is an active researcher in the field of big data privacy. Her research interests include privacy preserving data publishing, data anonymisation, big data privacy, etc.

Daphne Lopez is a Professor in the School of Information Technology and Engineering, Vellore Institute of Technology University. Her research spans the fields of grid and cloud computing, spatial and temporal data mining and big data. She has a vast experience in teaching and industry. Prior to this, she has worked in the software industry as a consultant in data warehouse and business intelligence.

Jemal H. Abawajy is a Full Professor at the School of Information Technology, Deakin University, Australia. He is currently the Director of the Distributing Computing and Security research group. He is a senior member of IEEE Computer Society. He has delivered more than 50 keynote addresses, invited seminars, and media briefings and has been actively involved in the organisation of more than 300 national and international conferences in various capacity. He has also served on the editorial-board of numerous international journals.

1 Introduction

Big data can be defined as large amount of data that cannot be processed using traditional database systems. Data from various sensors, hospitals and social networking sites are a rich source of information for big data. This rampant growth of data leads to various challenges in today's digital world where data publishing plays a major role in every aspect of health and economics. The type of big data ranges from unstructured text to highly structured data. This huge amount of data with diverse dimensionality raises two fundamental challenges in big data domain: storage and processing of raw data.

The data in its raw form may not be useful, but if we refine data, we can generate some profitable information out of it ('Tech giants may be huge, but nothing matches big data', 2014). This includes information that an industry can make use for improving their product quality and sales, health information that can help the analyst for finding out hidden patterns, etc.

There are many controversies regarding the standard definition of big data. Hashem et al. (2015) referred big data as a set of techniques and technologies that require new forms of integration to uncover large hidden values from large datasets that are diverse, complex and of a massive scale, whereas Dumbill (2012) defined big data as data that exceeds the processing capacity of conventional database systems.

The HACE theorem states a clear definition about big data: big data starts with large volume, heterogeneous, autonomous sources with distributed and decentralised control, and seeks to explore complex and evolving relationships among data (Wu et al., 2014). Gartner explained big data in terms of 3V's: volume, velocity and variety. The definition proposed in the Big Data Preliminary Report (2014) summarises big data as a dataset(s) with characteristics (e.g., volume, velocity, variety, variability, veracity, etc.) that for a particular problem domain at a given point in time cannot be efficiently processed using current/existing/established/traditional technologies and techniques in order to extract value.

The potential of big data lies with the integration of different types of data from various sources and generating value out of it. Big data integration can be used to build ecosystems that integrate structured, semi-structured and unstructured information from the published data. But, the major concern is with the privacy constraints in data publishing. Privacy can be defined as the right of individuals to determine how and to what extend information about them is communicated to others. The usual method of data sharing focuses on removing personally identifying information from the dataset that is published. As this mechanism does not prevent linkage attacks, various anonymisation mechanisms such as *k*-anonymity have been proposed for privacy preserving data sharing.

Various privacy models have been discussed in this paper which helps in effective data publishing while keeping the identity of the individual private. As we are dealing with privacy models for big data, traditional privacy models may not be efficient in terms of big data characteristics such as volume, velocity and variety. Big data privacy models can be well explained with the help of social network graphs and streaming data because of the characteristics it possess.

The main objectives of this survey focuses on providing an overview of the various big data privacy models, different methodologies followed to enhance privacy in data publishing and examining how effectiveness of privacy models are best assessed.

The rest of this paper is organised as follows. Section 2 presents big data characteristics while Section 3 focuses on the challenges with big data. Section 4 discusses about the importance of privacy preserving data publishing and Section 5 introduces the various privacy models. Section 6 gives an overview of how these privacy models can be extended to big data domain.

2 Big data characteristics

Big data can be well explained by examining its characteristics (Katal et al., 2013). The main characteristics of big data were identified as three V's: volume, velocity and variety. But, the recent update is that the characteristics of big data are not just limited to three V's, but on seven V's.

The characteristics of big data (seven V's) are explained as follows:

1 Volume: The amount of computed data generated is growing exponentially every second. A common data source to represent the volume of big data is that of social networks. The enormous amount of data in social networks is due to the image, video, music and text data that is uploaded by different users.

2 Velocity: This refers to the speed at which data arrives from different sources, the speed at which the data flows inside the system and how fast the data is processed.

E.g.: The main sources are sensor data, streaming data, social network data, etc. Nearly 100 hours of video are uploaded on YouTube every minute. More than 140 million tweets are published per day.

3 Variety: Data can be categorised as structured, semi-structured and unstructured. The data which is being produced from various sources may fall into any of these categories.

E.g.: A healthcare data source consists of patient's data which is structured, prescription by the doctor which is unstructured and the diagnostic report which is usually image data or graph data. The challenge is to gain insights by integrating this big data.

4 Veracity: This refers to the abnormality and uncertainties in data due to inconsistencies and incompleteness. There can be some noisy data, missing data or incorrectly entered data. This aspect is quite challenging because of the velocity and variety constraints of incoming data streams.

5 Value: This is one of the critical aspects in the case of big data. There is no meaning in collecting all the data unless we are able to generate some value out of it. Recommendations given by various sites based on user preferences and click stream data are one of the best examples to outline this characteristic of big data.

6 Variability: This can be defined as the data whose meaning is constantly changing. An example is the sentiment analysis of textual data. The words can take different meanings in different context. The challenge is to correctly identify the meaning of a word by understanding the context.

7 Visualisation: This focuses on the presentation of big data. Techniques should be adopted to represent this huge volume of data in an efficient manner. Dimensionality reduction plays a major role in visualisation of massive datasets.

By examining the characteristics of big data, the challenge it poses to the research community is very clear. Each characteristic of big data poses a new challenge.

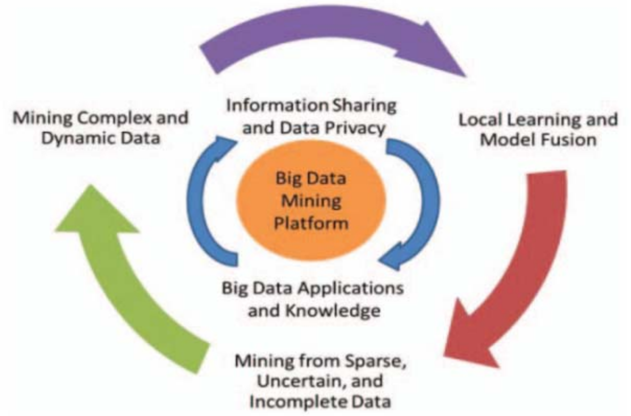
3 Challenges with big data

It is estimated that the data size will reach nearly 44 zettabytes, by the year 2020 ('Executive summary, data growth, business opportunities, and the IT imperatives', 2014). It is not the quantity of data that is posing challenges, but the improved methods in statistical and computational procedures makes it an issue ('Why 'big data' is a big deal', 2014).

3.1 Data processing framework

There are mainly three tiers for a big data processing framework (Wu et al., 2014) as shown in Figure 1: big data mining platform (tier 1); big data semantics/application knowledge (tier 2); and big data mining algorithms (tier 3).

Figure 1 Big data processing framework (see online version for colours)


Mining Complex and Dynamic Data
Information Sharing and Data Privacy
Local Learning and Model Fusion
Big Data Mining Platform
Big Data Applications and Knowledge
Mining from Sparse, Uncertain, and Incomplete Data

3.1.1 Big data mining platform

This tier focuses on data accessing and computing procedures. The amount of data is too large for a single PC or parallel computers to handle the data mining task assigned to them. Big data processing requires massively parallel software running on cluster computers. Some important technologies are discussed below.

Apache Hadoop is a framework for handling big data. This mainly consists of a file system, HDFS and the programming paradigm, MapReduce. Hadoop distributed file system is for storing and processing large files. MapReduce allows massive scalability. This model has two main tasks: map and reduce. The map function takes as input a key-value pair and produces a set of intermediate key-value pairs. A reduce function reduces the set of values with the same key to a smaller set of values. The reduce function will be invoked for every unique key. In place of MapReduce, Google is now moving with cloud data flow which is based on their internal technologies, flume and millwheel ('Google moves on from MapReduce, launches cloud dataflow', 2014).

3.1.2 Big data semantics and application knowledge

This tier deals with data sharing, information privacy and data modelling. In order to gain valuable insights from data, it should be analysed. But, the datasets could contain sensitive information that the user does not want others to know about. Sharing sensor data for weather analysis does not question the privacy in any manner, as it does not contain private information about any individual. But, at the same time, sharing the medical records of a patient will not be acceptable for many. So, based on the sensitivity of the data that is shared, privacy preserving mechanisms should be formulated. There are mainly two areas in this: data privacy and location privacy. This paper focuses mainly on the notion of data privacy.

3.1.3 Big data mining algorithms

This deals with algorithm designs for solving the difficulties raised by big data volume, variety, velocity and veracity. There are many tools which deal with solving some specific issues with big data. Many researches are still going on to solve the challenges with big data. Everyday, some new technologies are arriving in the market which claims that it could solve all the issues faced by big data. But, the fact is that there is no accepted efficient and best data model for handling big data so far.

3.2 Security and privacy challenges

The challenges in big data privacy and security domain can be organised into four aspects (Big Data Working Group, Cloud Security Alliance, 2013). They are:

- *Infrastructure security*: The distributed computations and data stores must be secured in order to secure big data systems. The MapReduce computation model allows parallel computation of data in distributed frameworks. The major attack prevention measures are focused on securing the mappers and securing the data in the presence of an untrusted mapper. Trust establishment and MAC ensure the trustworthiness of mappers.
- *Data privacy*: This is a challenging area in big data domain. Here, the focus is on securing the data itself. For this, information exchange and circulation should be privacy preserving and the sensitive and most

important data should be cryptographically secured. Privacy preserving data mining and sharing becomes an important area in today's digital world, as it provides maximum utility of published dataset without questioning individual's privacy.

- *Data management*: Managing massive datasets require efficient solutions for securing the data storage. Granular access control mechanisms prevent unauthorised users from accessing data elements. Audit information is yet another important aspect which helps in providing better security by employing query auditing mechanisms.
- *Integrity and reactive security*: This mainly includes performing real-time security monitoring and end point input validation and filtering. Real-time security monitoring is all about monitoring the big data infrastructure and its applications. As the amount of data generated by big data systems increases enormously, validation of input data poses a real challenge to the big data domain.

Hashem et al. (2015) have conducted a study on the 'Rise of big data on cloud computing' and identified various challenges in big data processing arena. A detailed review on the categories of big data with respect to data sources, content format, data stores, data staging and data processing is discussed in this paper. Several studies that deal with big data through the use of cloud computing technology are presented in this survey.

Chen et al. (2014) surveyed the various technical challenges on the four phases of big data value chain such as data generation, acquisition, storage and analysis. An introduction to cloud computing, IoT, data centre and Hadoop is given in this paper. A comparison on MPI, MapReduce and Dryad is also discussed.

A study on developing a holistic approach for big data is conducted by Sagiroglu and Sinanc (2013). This paper discusses about the various privacy and security issues in big data domain and explains the characteristics of a big data driven security model. Some of the unique privacy risks presented by big data such as incremental effect, automated decision making, predictive analysis and chilling effect is presented by Tene and Polonetsky (2013). This paper also reviews the legal framework challenges with respect to data minimisation and individual control of data, and the solutions that can be adopted for a data publishing scenario.

Emani et al. (2015) presented a survey on the various concepts of big data such as big data processing frameworks, challenges in big data management and architectural solutions for handling big data. This paper defines and characterises the concept of big data. Various technologies for big data management are also presented in this paper.

4 Privacy preserving data publishing

Privacy and security are often confusing terms. Data security ensures that data is available when those with authorised access need it. Data privacy ensures that the data is used appropriately. Privacy is defined as the right of individuals to determine how and to what extent information is communicated to others. Data privacy can be protected either by restricting access to the data by using access control methods or by anonymising the data.

The two main classes of data privacy are:

1 When the data is to be released or shared to third parties. This includes data collection, data integration and privacy preserving data publishing and sharing.

2 Privacy preserving data mining.

The current privacy protection practices focus on policies and guidelines to restrict the types of publishable data. Methods and tools need to be developed for publishing data to make it useful for researchers/analysts. The main idea is to publish sensitive data for gaining valuable insights without questioning individual's privacy. This approach is called as privacy preserving data publishing. A data publishing scenario usually consists of a data owner, who is the sole author of the data, the data holder, who stores the data about these data owners and the data recipient, who will be using the published dataset. There are two models of data publishers: trusted and un-trusted. The data publisher is trust worthy in the case of trusted model, whereas the data publisher is not trusted in the case of un-trusted model. This paper focuses on the trusted model of data publishers.

5 Privacy models

In the case of traditional model, two types of attack are common. In one type of attack, the attacker will be able to identify an individual from the published table. This can be done either by linking the record, attribute or the whole table itself. The second type of attack is the probabilistic attack. There are various privacy models which can efficiently deal with these types of attacks by providing better privacy for the data that is published in the released table. These privacy models ensure privacy either at the record level, attribute level, table level or at all levels.

Consider the in-patient (IP) dataset of a hospital given in Table 1.

There are four types of attributes in a table:

1 Explicit identifiers: These are the attributes used to identify an individual uniquely. Such details are always removed when a dataset is published. Explicit identifiers in this table include the IP number and name of the patient.

2 Quasi identifiers: This data seems to be harmless as it cannot reveal the identity of any individual explicitly. But, these attributes can be combined with some external information to uniquely identify an individual

in a population. The quasi identifiers in this table are age, gender and job.

3 Sensitive attributes consists of person specific information that are considered as sensitive with respect to the data owner. This is always released and this is the data that the researcher's need. The sensitive attribute in this table is the attribute 'diagnosis'.

4 Non-sensitive attributes include those attributes which are not sensitive.

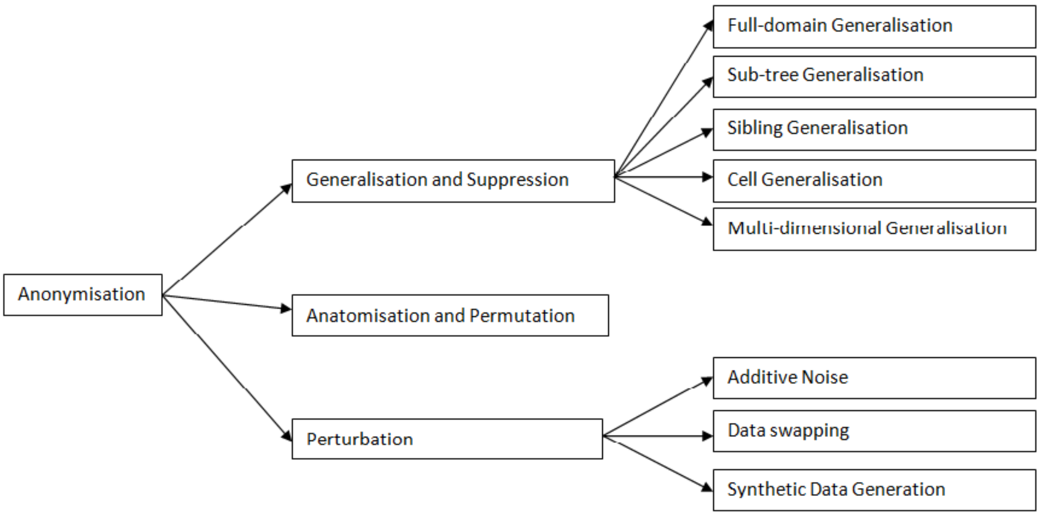
As mentioned above, all the explicit identifiers will be removed and only the quasi identifiers, sensitive attributes and non-sensitive attributes are published during the data publishing phase. The dataset should be modified before publishing the data. This is accomplished by performing a variety of anonymisation operations on the dataset. The various approaches for anonymisation (Fung et al., 2010) are given in Figure 2.

The approach of generalisation is used to replace specific values with more general ones. As a result of this approach, many tuples will be having the same set of values for quasi identifiers. The term equivalence class can be defined as the set of tuples that have the same value for quasi identifiers. Anatomisation and permutation operations aim in de-linking the relation between quasi-identifiers and the sensitive attributes. Perturbation works by adding some noise to the original data before presenting that to the user.

Table 1 Original dataset

Sl. no.	*IP no.*	*Name*	*Age*	*Gender*	*Job*	*Diagnosis*
1	140010	Ann	21	F	Dancer	Hepatitis
2	140011	Emil	32	M	Singer	Influenza
3	140012	Susanne	23	F	Keyboardist	Malaria
4	140013	David	34	M	Dancer	Malaria
5	140014	Jacob	38	M	Dancer	Hepatitis
6	140015	Carolina	27	F	Singer	Influenza
7	140016	Diana	29	F	Keyboardist	Hepatitis
8	140017	Nathaniel	39	M	Music director	Influenza
9	140018	John	42	M	Engineer	Malaria
10	140019	Mathew	46	M	Doctor	Influenza
11	140020	Paul	47	M	Lawyer	Hepatitis
12	140021	Robert	43	M	Engineer	Malaria

Figure 2 Anonymisation approaches


Full-domain Generalisation
Sub-tree Generalisation
Sibling Generalisation
Generalisation and Suppression
Cell Generalisation
Multi-dimensional Generalisation
Anonymisation
Anatomisation and Permutation
Additive Noise
Perturbation
Data swapping
Synthetic Data Generation

This section gives a detailed description about the various privacy models. Each privacy model makes use of any anonymisation operations for giving better results.

5.1 *k-anonymity*

The normal practice of publishing person specific data is by removing explicit identifiers from the table. As we have discussed, the explicit identifiers in Table 1 are the IP no. and name. The dataset after removing these explicit identifiers are given in Table 2.

As stated by Sweeney (2002), this may not be sufficient for preserving the privacy of data that is published. This was proved by uniquely identifying the record of a particular individual by linking two databases. One database had the medical data published in Massachusetts hospital dataset and the other database had the voter's list, which had three attributes in common: zip, date of birth and the sex. Sweeney was able to find the diagnosis and the medication provided for the then governor by identifying the record correctly. This was done by linking two databases and concluded that nearly 87% of the records can be linked using the same approach. *k*-anonymity technique solves this problem by preventing record linkage.

Table 2 Dataset after removing the explicit identifiers

Sl. no.	*Age*	*Gender*	*Job*	*Diagnosis*
1	21	F	Dancer	Hepatitis
2	32	M	Singer	Influenza
3	23	F	Keyboardist	Malaria
4	34	M	Dancer	Malaria
5	38	M	Dancer	Hepatitis
6	27	F	Singer	Influenza
7	29	F	Keyboardist	Hepatitis
8	39	M	Music director	Influenza
9	42	M	Engineer	Malaria
10	46	M	Doctor	Influenza
11	47	M	Lawyer	Hepatitis
12	43	M	Engineer	Malaria

Definition 1: Let $RT(A_1, \ldots, A_n)$ be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy *k*-anonymity if and only if each sequence of values in $RT[QI_{RT}]$ appears with at least k occurrences in $RT[QI_{RT}]$ (Sweeney, 2002).

For satisfying *k*-anonymity when $k = 2$, an equivalence class should have at least two rows. Generalisation operation will be first done on the dataset and then the *k*-anonymisation technique is applied.

The dataset after applying *k*-anonymisation is given in Table 3. The tuples 1 and 2 belongs to the same equivalence class as it contains same value for all the quasi-identifiers. Likewise, this table contains six equivalence classes.

Table 3 *k*-anonymised table with $k = 2$

Sl. no.	*Age*	*Gender*	*Job*	*Diagnosis*
1	20–25	F	Artist	Hepatitis
2	20–25	F	Artist	Malaria
3	25–30	F	Artist	Influenza
4	25–30	F	Artist	Hepatitis
5	30–35	M	Artist	Influenza
6	30–35	M	Artist	Malaria
7	35–40	M	Artist	Hepatitis
8	35–40	M	Artist	Influenza
9	40–45	M	Professional	Malaria
10	40–45	M	Professional	Malaria
11	45–50	M	Professional	Influenza
12	45–50	M	Professional	Hepatitis

One of the assumptions with this approach is that the data holder knows about the quasi-identifiers that should be taken care of in the table. An approach for automatic identification of quasi-identifiers is explained by Lodha and Thomas (2008). This gave way for the probabilistic notion of anonymity.

k-anonymity suffers from certain drawbacks. One of the main attacks with *k*-anonymity is the homogeneity attack. This attack happens when the sensitive attribute lacks diversity. Another popular attack with *k*-anonymity is the background attack. This attack occurs when the adversary has some background knowledge about the individual. *k*-anonymity approach does not help in preventing attribute disclosure. Attribute disclosure means that an adversary will be able to gain additional insights about an individual even without linking to any item in the published table.

5.2 *l-diversity*

The attacks on *k*-anonymity gave way for the improved notion of publishing the table on the basis of *l*-diversity. Consider the *k*-anonymised dataset given in Table 3.

From Table 3, it is very clear that at least 2 tuples are having the same values for quasi identifiers. But, when we examine the rows 9 and 10, we can conclude that all the male patients within the age limit 40–45 who are professionals, are affected with malaria. This seriously questions the privacy of the individual. This issue occurred because of less diversity in the sensitive attributes.

There are mainly two factors that can question the privacy of the individual in a published table. One factor is the lack of diversity of sensitive attributes in the table. Another factor is when the adversary has strong background knowledge about a particular individual. Positive disclosure and Negative disclosure are the two ways in which sensitive information can be leaked from a table. If the attacker can correctly identify the individual with high probability, it is called as positive disclosure. If the attacker can correctly eliminate possible values of the sensitive attribute, it is called as negative disclosure.

This privacy model not only prevents record linkage, but also prevents attribute linkage. Even if the data publisher has no idea about the back ground information that is possessed by the attacker about the individual, this model protects the privacy in an efficient manner using the concept of Bayes optimal privacy. The *l*-diversity principle can be explained as:

Definition 2: A Quasi-identifier block *Q* is *l*-diverse if it contains at least *l* well represented values for each sensitive attribute *S*. A table *T* is *l* diverse if every *Q* block (Equivalence class) is *l*-diverse (Machanavajjhala et al., 2007).

If there are at least *l* well represented values for sensitive attributes, the adversary needs to eliminate *l*-1 possibilities of sensitive attributes to gain a positive disclosure about the information of the individual.

Table 4 Three-diverse dataset

Sl. no.	*Age*	*Gender*	*Job*	*Diagnosis*
1	< 30	F	Artist	Influenza
2	< 30	F	Artist	Hepatitis
3	< 30	F	Artist	Malaria
4	< 30	F	Artist	Hepatitis
5	≥ 40	M	Professional	Malaria
6	≥ 40	M	Professional	Malaria
7	≥ 40	M	Professional	Influenza
8	≥ 40	M	Professional	Hepatitis
9	3*	M	Artist	Influenza
10	3*	M	Artist	Malaria
11	3*	M	Artist	Hepatitis
12	3*	M	Artist	Influenza

The three-diverse dataset is given in Table 4. In the table shown above, it satisfies the *l*-diversity principle for *l* = 3. This means that, in every block of quasi identifiers in the table, at least three different sensitive values will be recorded. Here, the tuples are grouped into blocks based on the values of Quasi Identifiers. Thus, each block consists of 4 tuples in the dataset. All the values of quasi identifiers in each block will be the same, which satisfies the *k*-anonymity approach for *k* = 4. But, we can see that each block is having three different values for sensitive attributes which shows that the dataset conforms to *l*-diversity principle.

l-diversity approach holds the property of monotonicity. Suppose we have a table T. It is generalised using some generalisation technique to get a new table T*. The monotonicity property states that if the table T* is preserving privacy, then all the generalisations of T* also preserves privacy.

There are various connotations for the term 'well represented' in *l*-diversity approach such as the distinct *l*-diversity, entropy *l*-diversity and the recursive (*c*, *l*) diversity.

l-diversity suffers from certain drawbacks. *l*-diversity is too difficult and unnecessary to achieve. If the sensitive attribute is just taking one of two values 'affected' or 'not affected' and if 90% of the people are in the category 'not affected', it may be acceptable for the individuals in that category to reveal their status. But, this will not be the case for individuals who were tested as positive. They always want to keep their information private.

There are mainly two types of attacks: skewness attack and similarity attack. Skewness attack occurs when each block of quasi identifiers (equivalence class) has equal probability for positive and negative values of sensitive attributes. Similarity attack happens when the values of sensitive attributes seems to be different, but are actually similar in meaning. This attack occurs because the principle considers the diversity of sensitive attributes, but does not consider the closeness of various values in the sensitive attributes.

5.3 *t-closeness*

As discussed in the previous section, the closeness of sensitive attributes can lead to a privacy breach. This is because even if the sensitive attributes will be considered while preserving *l*-diversity, it does not consider whether the values of these sensitive attributes will lead to the same conclusion. This can be explained with the help of an example. Suppose there are 4 tuples in an equivalence class. This class preserves the *l*-diversity privacy model, where *l* = 3. According to the *l*-diversity model for *l* = 3, there are three different values of sensitive attributes in each equivalence class. But, suppose the values are given as stomach pain, gastritis and stomach cancer, one can very well conclude that all the individuals in that class is suffering with some type of stomach-related problems. The approach of *t*-closeness takes this issue into account and proposes a novel method for finding out the distance between two equivalence classes. This privacy model helps to protect against attribute disclosure. This approach uses Earth mover distance (EMD) measure.

The *t*-closeness principle states that:

Definition 3: An equivalence class is said to have *t*-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold *t*. A table is said to have *t*-closeness if all equivalence classes have *t*-closeness (Li et al., 2007).

The idea of *t*-closeness incorporates the idea of *k*-anonymity and *l*-diversity. If the *t*-closeness principle holds for a dataset, it is known that the dataset obeys *k*-anonymity and *l*-diversity principles as well. The '*t*' parameter in *t*-closeness often compromises the notions of privacy and utility. It means that there is a trade-off between these measures. There are various methods to find the distance between probability distributions. The various approaches for finding out the distance are proposed by Ho and Yeung (2010), Cardoso (1997) and Andoni et al. (2008).

The main idea of *t*-closeness is information gain. This is defined as the difference between prior and posterior beliefs of an adversary about the sensitive attribute of a table before publishing it and after getting published. This information gain is found for the whole table and also for some set of individuals in the table. The distance between probability distributions is found out using EMD measure.

Consider two quasi-identifier blocks E_1 and E_2. Let the distribution of sensitive attribute for E_1 is P_1, E_2 is P_2 and E_1U E_2 is P. Then,

$$\mathrm{D}[\mathbf{P},\mathbf{Q}] \le \frac{|E_1|}{|E_1|+|E_2|}\mathrm{D}[\mathbf{P}_1,\mathbf{Q}] + \frac{|E_2|}{|E_1|+|E_2|}\mathrm{D}[\mathbf{P}_2,\mathbf{Q}]$$

Different approaches should be devised for finding the EMD of numerical attributes and categorical attributes. The dataset which satisfies *t*-closeness is given in Table 5. From this table, it is clear that the distribution of sensitive attributes influenza, hepatitis and malaria in each quasi identifier block and the entire dataset has similar distribution.

Table 5 *t*-closeness

Sl. no.	*Age*	*Gender*	*Job*	*Diagnosis*
1	≤ 35	*	Any job	Influenza
2	≤ 35	*	Any job	Influenza
3	≤ 35	*	Any job	Hepatitis
4	≤ 35	*	Any job	Hepatitis
5	≤ 35	*	Any job	Malaria
6	≤ 35	*	Any job	Malaria
7	> 35	*	Any job	Influenza
8	> 35	*	Any job	Influenza
9	> 35	*	Any job	Hepatitis
10	> 35	*	Any job	Hepatitis
11	> 35	*	Any job	Malaria
12	> 35	*	Any job	Malaria

One of the main issues with *t*-closeness is that different levels of sensitivity should be specified for different sensitive attributes. The approach of *t*-closeness protects against attribute disclosure, but does not protect against identity disclosure. Another challenge with this approach is when more number of sensitive attributes is to be considered. Privacy can be questioned if more number of sensitive attributes are published in a table.

Another important challenge is with the data quality after performing anonymisation approaches like generalisation and suppression. Data quality can be improved if both the approaches are used correctly. A more useful measure other than EMD will bring good results. EMD will not be efficient for preventing attribute linkage on numeric sensitive values.

5.4 Differential privacy

The main issue with all the privacy models discussed so far is that privacy breach occurs because of publishing the exact data which is available in the dataset. So, if an adversary is having strong back ground knowledge, he can deduce the status of sensitive attribute of an individual in a published dataset. This can be made possible with the help of other published information as well.

The idea of differential privacy is to publish the results of a query by adding some noise to the data which is already available. So, an attacker will not be able to conclude anything with 100% confidence. The main conviction is that the conclusions obtained about an individual are due to the data from the entire dataset and not due to a particular tuple in the published table.

The notion of this approach is that the risk of individual privacy should not be increased by having a record in the statistical database. This model ensures that removal or addition of a particular record in the published database does not affect the overall analysis of the data in the table. The model is quite good in overcoming linkage attacks.

The model can be explained as:

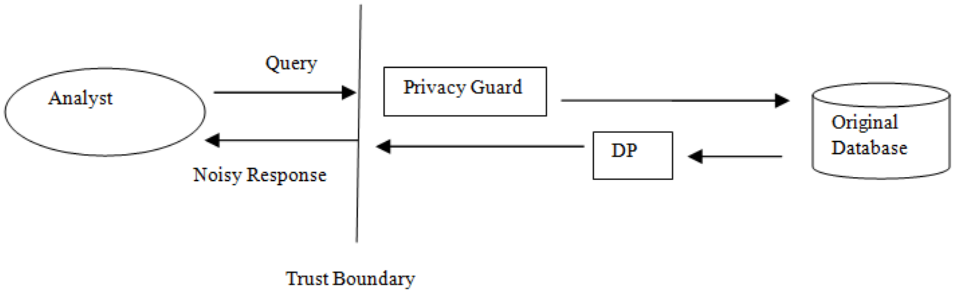
Definition 4: A randomised function K gives ε-differential privacy if for all datasets D and D′ differing on at most one row, and all $S \subseteq \text{Range}(K)$, $\Pr[K(D) \in S] \le \exp(\varepsilon) \times \Pr[K(D') \in S]$ (Dwork, 2011).

The choice of the value for 'ε' depends upon the context of the data and the query. This model can protect the individual against adversary having back ground knowledge also. Even if the individual is not giving the correct information for publishing, it will not affect anonymisation algorithms and operations. This model ensures security for the individual to publish their records, by having the guarantee that nothing can be discovered from the database about their details. This can work with both interactive and non-interactive queries.

The base architecture for differential privacy mechanism as suggested by Friedman and Schuster (2010), and Microsoft Corporation, 'Differential Privacy for Everyone' (2012) is given in Figure 3.

As discussed in the report of Microsoft Corporation, 'Differential Privacy for Everyone' (2012), differential privacy plays a pivotal role in preserving the privacy of individuals by inserting a layer between the original dataset and the researcher or analyst. The analyst sends a query to the privacy guard. The privacy guard in turn gets the data from the original dataset. If the result of the query is resulting in privacy breach, some noise is added by the differential privacy mechanism and the noisy response will be given to the user. Differential privacy aims in distorting the data to a small amount so that it will neither affect the analysis of the researcher nor questions the privacy of the individual.

Figure 3 Architecture for differential privacy



Various mechanisms for adding the noise in numerical and categorical data have been proposed by various authors. Sarathy and Muralidhar (2011) evaluated the privacy and utility performance of Laplacian noise addition for numeric data in order to achieve differential privacy. Dwork (2008) discussed about a privacy mechanism which adds an amount of noise that grows with the complexity of the query sequence. McSherry and Talwar (2007) proposed a general mechanism with guarantees on the quality of output even for functions that are not robust to additive noise. Li et al. (2010) proposed a matrix mechanism for answering a workload of predicate counting queries, which preserves differential privacy but increases accuracy.

There are many limitations for differential privacy. One of the major limitations of differential privacy is that it does not give guarantee against attribute and record linkage. This model is mainly used in scenarios where very less number of queries are used, that too with low sensitivity. This is good in the case of restricted class of queries.

6 Big data privacy models

All the privacy preserving mechanisms we have discussed so far is efficient for dealing with traditional data like relational databases. In this section, a discussion on the adoption of these privacy models to work with big datasets such as live streaming data and unstructured data is given. This section will give a good insight into big data techniques using the MapReduce model also.

The privacy models k-anonymity and l-diversity are discussed with the help of social network graphs. The idea of t-closeness is explained for data streams. Differential privacy is explained using MapReduce implementation. As we all know that the main sources of big data includes social network data and streaming data, this section will give a good insight into the anonymisation practices that can be adopted for big data.

6.1 Anonymising social network data modelled as graphs

Let us first discuss about social network graphs briefly. We generally think about social networking sites such as Facebook when we hear about social networks. But, there are a lot of other social network graphs such as telephone networks, e-mail networks, etc. (Leskovec et al., 2011). Even though a graph model is used for representing social networks, all graphs cannot be modelled as social network graphs. This is because; community clusters should be made from the social network graphs. This explains the idea of locality in graphs.

The nodes in a social network graph represent ‘people’ usually and the edges are created between two nodes if there is any link between them. When we consider about friend networks such as Facebook, nodes are the ‘individuals’ and an edge is used for representing friendship between two individuals. Various models for social network analysis are presented by Carrington et al. (2005). Social network data analysis help in many ways including disease outbreak prediction and evaluation of network faults. Even then, it is difficult to access network data because of the sensitive content it possess. Anonymisation of network graphs becomes important in such a scenario. We will discuss about the techniques for anonymising social network data in this section.

One of the main attacks in the case of social network graphs is neighbourhood attack. Even if the details about the individual is preserved using traditional techniques like k-anonymity and l-diversity, privacy can be questioned by linking the background knowledge about the neighbour of the individual. Suppose that a social network graph is published just by removing the names of nodes in the graph. Even then, the details of the individual can be found out if the person is having two friends in common, just by getting the details about his neighbours.

The major challenges in the case of privacy preservation of social network data are pointed out below (Zhou and Pei, 2011):

1 It is very difficult to model the back ground knowledge.

2 It is quite challenging to find the information loss in social network data.

3 Finding out the best technique for anonymising social network data is yet another interesting challenge.

A social network is modelled as a simple graph with vertices, edges, labels and a labelling function. In order to correctly model the problem, the information that can lead to a privacy breach should be identified first. Then only, the best model for attack can be devised. Various methods have been proposed for efficient anonymisation of graphs. All these methods use the concept of privacy models for traditional data.

Zhou and Pei (2011) proposed a method of two-level vertex anonymisation, where the first level of anonymisation is applied for the labels by giving a more general value instead of a specific value and the second level of anonymisation includes adding an edge, but not adding or removing a vertex.

Definition 5: Let G be a social network and G′ an anonymisation of G. If G is *k*-anonymous, then with the neighbourhood background knowledge, any vertex in G cannot be re-identified in G′ with confidence larger than 1/*k*.

There are two major steps to achieve *k*-anonymity in social network graphs: First step is to identify the neighbourhoods of all vertices in the graph. In order to identify the neighbours of a vertex, a coding technique is proposed. The DFS approach can be used to encode all the edges and vertices in the graph G. The second step focuses on anonymisation. If some set of vertices have the same degree, they should be organised as a group and anonymisation techniques should be applied. There are mainly two properties for anonymisation in social networks. One property is the 'vertex degree in power law distribution' (Adamic et al., 2001) and the other one is 'the small-world phenomenon' (Watts and Strogatz, 1998).

Social network greedy anonymisation (SaNGreeA) approach was proposed by Campan and Truta (2009) to effectively anonymise social network data based on clustering approach. Nodes are described by quasi identifiers and the edges are undirected and unlabelled. First step focuses on partitioning the nodes of a particular category into various clusters and the second step deals with homogenising the nodes in a particular cluster by means of generalisation approach. Two information loss measures are addressed in this approach: generalisation information loss and structural information loss measure.

Hay et al. (2008) proposed an idea of anonymising elements of a network and sampling from the model, for analysing the information. First step relies on relabeling the nodes of a graph using the naive anonymisation technique and the second step deals with grouping the nodes into partitions. A privacy parameter '*k*' is specified as input, which decides on the size of super nodes to be '*k*'. This model helps in resisting structural re-identification.

The method of link anonymisation was proposed by Fard and Wang (2015) using the neighbourhood randomisation scheme. Link anonymisation refers to the process of anonymising the sensitive edges between two nodes. This sanitisation approach relies on a structure aware randomisation scheme where either the source or destination of a link is hidden so that it would be hard to find out the true existence of a link. The idea is to use the randomisation technique where a destination is retained with a probability '*p*' and replaced with a wrong destination with a probability '1-*p*'.

Another method of identity anonymisation was proposed by Clarkson et al. (2010), which is based on the *k*-anonymity approach. This includes creating a new degree sequence which satisfies *k*-anonymity and creating a super graph out of the original graph. The main drawback of *k*-anonymity model in social network graph is due to the lack of diversity in sensitive attributes. Linkage attack is the major attack in this case.

Definition 6: Let G be a social network and G′ be an anonymisation of G. G′ is said to be *l*-diverse if in every equivalence group of vertices, at most 1/*l* of the vertices are associated with the most frequent sensitive label.

The algorithm for *l*-diversity for social networks based on vertex anonymisation is given below (Zhou and Pei, 2011):

Step 1 Mark all the vertices as un-anonymised.

Step 2 Choose a seed vertex.

Step 3 For all the vertices in the vertex list, find the cost between the vertices and the seed vertex.

Step 4 If the size of the un-anonymised vertices in the vertex list is greater than or equal to 2*l*-1, the candidate set will contain *l*-1 vertices with the least cost to the seed vertex such that these *l*-1 vertices carry the unique sensitive values which is different from that of the seed vertex. Otherwise, the set will contain the rest of vertices which are not anonymised.

Step 5 If there are enough *l*-diverse values, then for the vertices in the candidate set, perform anonymisation for the neighbours of seed vertex and the vertices. Update the vertex list.

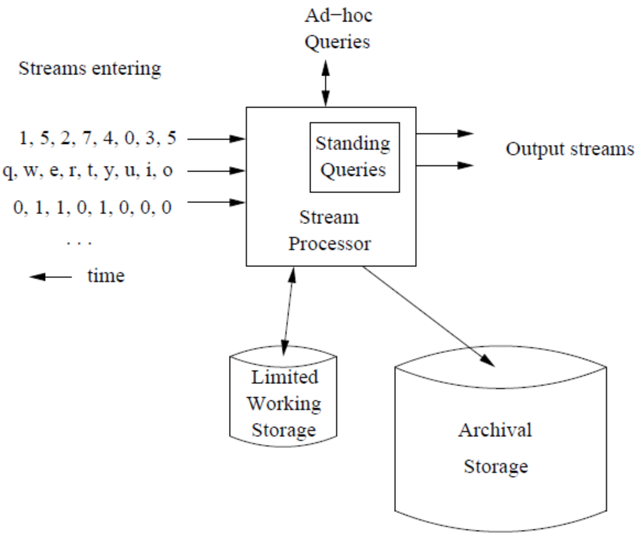
Step 6 Iterate the process until all the neighbourhoods of the selected seed vertex is anonymised and a valid candidate set is formed.

The drawback is that the idea just focuses on only one type of attack, the neighbourhood attack. There are a variety of attacks in the case of social network graphs. SybilGuard is a protocol for limiting the corruptive influences of Sybil attacks (Yu et al., 2006). Error tolerance of complex networks is discussed in Albert et al. (2000). Bilge et al. (2009) investigated about various attacks against popular social networking sites, in order to gain access to a large volume of personal user information. Another important drawback to consider is the closeness of sensitive attributes in the labels of nodes in the graph.

6.2 *Anonymisation approaches for streaming data*

Usually, if we want to process some data, it will be available beforehand. But, in the case of streaming data, the data will be arriving as streams. If we are not able to analyse the data or store it as soon as it is arriving, the data will be lost forever. Figure 4 shows the data stream model (Leskovec et al., 2011).

Figure 4 The data stream model



There are mainly two types of queries: ad hoc queries and standing queries. Ad hoc queries will be fired only when it is needed. But, the standing queries are the queries which are stored permanently, and executed. Streaming queries can be fired for streaming data (Chandrasekaran and Franklin, 2002). There are various algorithms for high quality clustering (O'Callaghan et al., 2002) and sentiment discovery (Bifet and Frank, 2010) for streaming data. This section discusses about the anonymisation techniques for streaming data.

The idea of *t*-closeness considers the closeness of values of sensitive attributes into account. This is done using the concept of bucketisation and redistribution. The need of *t*-closeness for big data is similar to the need of this approach in the case of relational data. The approach takes into account the streaming data and explains how it can be made available to the public by anonymising the data.

Sensitive Attribute Bucketization and REdistribution (SABRE) is a framework by Cao et al. (2011a) for achieving privacy using *t*-closeness approach. SABREw is an extension for SABRE framework to work with streaming data. For each input data stream, a window is set. As one of the window moves, the next window will be processed. The tuples which are not yet published from the old window will be given as output. These are called as the expiring tuples. Anonymisation is performed for each window data by comparing it with some sample. For all the data in input stream, the anonymised data is sent as the output stream. The sensitive attribute distribution in the equivalence class does not differ from that of all tuples by more than a threshold t. The window buffers the first set of tuples from the input stream. When the new set arrives, the first set of tuples will get expired and will be sent as the output stream.

Various filtering mechanisms can be used to filter the data streams. Deep packet inspection using parallel bloom filters is discussed by Dharmapurikar et al. (2003). A new architecture for data stream management, Aurora is proposed by Abadi et al. (2003) and an approximate Kalman filter for ocean data assimilation is proposed by Fukumori and Malanotte-Rizzoli (1995). One of the drawbacks of this model is that auto detection of '*t*' in *t*-closeness for data streams is not addressed deeply in any research. Another drawback is that the data is not republished as it gets updated.

CASTLE, the technique proposed by Cao et al. (2011b) uses adaptive clustering method for continuously anonymising data streams. The anonymised data also will be kept up-to-date by satisfying the maximum delay that can be allowed between the incoming and outgoing data streams. FAANST (Zakerzadeh and Osborn, 2011) is an anonymising algorithm which is basically dedicated for working with numerical data. This is based on cluster-based *k*-anonymity approach for anonymisation.

The method of stream *k*-anonymity was proposed by Li et al. (2008) for facilitating continuous anonymisation of data streams. A specialisation tree is built based on domain generalisation techniques. There are two nodes: candidate nodes and work nodes. Candidate nodes does not satisfy the *k*-anonymity requirement for the current set of tuples, when compared to all the streaming data seen so far, where as work nodes satisfy the requirements. When a new tuple arrives, it searches for the best generalisation approach and check to see whether it is a work node or candidate node. If it is a work node, it will be anonymised immediately. Otherwise, it will be kept until it satisfies the *k*-anonymity constraint. A randomised algorithm framework is proposed by Zhou et al. (2009) for satisfying anonymity in data streams.

Large iterative multitier ensemble (LIME) classifiers have been proposed by Abawajy et al. (2014) for ensuring better security for big data. This four-tier ensemble classifier is used for detecting malware in big data and can be combined with various privacy models for building a secure framework for batch and streaming data.

6.3 Differential privacy for big data

The main issue which is addressed in the case of differential privacy is to give a noisy response to the analyst who fires the query. This is because; knowingly or unknowingly the privacy of an individual should not be questioned. This occurs only when we find that the result of the query can lead to a privacy breach. For this, different probability distribution functions should be devised based on the type of query that can be asked and also based on the type of data which is present in the dataset. In the case of big data processing, this poses a great challenge as we have to find out the noise based on the entire dataset. This issue is addressed in the following section.

Airavat (Roy et al., 2010) is a MapReduce system which preserves the privacy of MapReduce computations with un-trusted code. This system includes an un-trusted/trusted mapper and a trusted reducer. The main idea is to enable privacy preserving computations on a large set of data items, which belongs to different data owners. Differential privacy has been employed here to give perturbed results to the program. There is no need to audit the mappers. But, the Reducer will prevent the leakage of information by adding some noise. In this system, reducers are responsible for enforcing privacy. This system integrates MAC with differential privacy for better results.

Some notations for differential privacy used by Airavat are given as follows: sensitivity of a function refers to the maximum deviation in the answer when a single item is either added or removed from the original dataset. Finding out the sensitivity of a function is not an easy task. So, the analyst should specify the function's sensitivity while giving the code. The amount of noise which is to be added also depends on this sensitivity parameter. Privacy budget is the term used for calculating the number of queries that can be asked on a particular dataset. If the privacy budget is used completely, results will not be automatically declassified. Along with the sensitivity of the function, the analyst is asked to mention the range of values he expects for the output. This enforcement of range values gives more priority to privacy when compared to accuracy of the results.

There are mainly three entities in Airavat: The data provider, the analyst and the framework. The analyst writes the MapReduce code for analysis. The data provider will be setting the parameters needed for analysis. The data owner will be specifying the privacy parameters '∈' and 'δ', and the privacy budget 'PB'. The parameter '∈' trade-offs between privacy and utility. The parameter 'δ' stands for small absolute differences in probabilities. The analyst/computation provider specifies the map function, the minimum and maximum value expected and the number of output keys. The pseudo code for MapReduce implementation is given below. In this implementation, Laplacian noise has been added to give a noisy response for the query.

6.3.1 Map phase

1 Initially, check for the privacy budget. If PB – ∈ × N is less than 0, specify that the limit is exceeded. Otherwise, set the budget as PB = PB – ∈ × N.

2 For each record r, find the set of key-value pairs for the query.

3 If any value does not fall in the range, give the average of minimum and maximum value specified by the analyst, as the value of that key.

4 Emit all the key-value pairs.

6.3.2 Reduce phase

1 Set the count as the number of output keys.

2 In the reduce function, if the value of – – count turns out to be less than or equal to 0, skip the step. Otherwise, find the sum of all values.

3 Add the Laplacian noise to the original value and print the output for all the values from 1 to N.

Table 6 Privacy models for big data

Sl. no.	*Anonymisation approach*	*Privacy model/algorithm*	*Issues addressed*	*References*
1	Vertex anonymisation	*k*-anonymity	Neighbourhood attack	Zhou and Pei (2011)
2	Identity anonymisation	Greedy-swap algorithm	Identity attack	Clarkson et al. (2010)
3	Graph generalisation	Naive anonymisation	Structural re-identification	Hay et al. (2008)
4	Link anonymisation	Neighbourhood randomisation	Link privacy	Fard and Wang (2015)
5	SaNGreeA algorithm	Greedy clustering approach	Data anonymisation	Campan and Truta (2009)
6	Vertex anonymisation	*l*-diversity	Anonymising sensitive attributes	Zhou and Pei (2011)
7	Data stream anonymisation	SABREw *t*-closeness	Diversity of attributes in the entire set	Cao et al. (2011a)
8	Clustering approach	CASTLE	Delay between incoming and outgoing streams	Cao et al. (2011b)
9	Cluster-based *k*-anonymity approach	FAANST	Numerical data anonymisation	Zakerzadeh and Osborn (2011)
10	Domain generalisation approach	Generalisation	Data stream anonymisation	Li et al. (2008)
11	Four-tier classifier based on random forest	LIME classifier	Detection of malware	Abawajy et al. (2014)
12	Untrusted mapper and trusted reducer	Airavat framework differential privacy	Privacy budget	Roy et al. (2010)
13	Hierarchical multi population approach	Evolutionary optimisation technique	Search space	Bhattacharya et al. (2015)
14	Two phase top down specialisation	*k*-anonymity	Scalability of large datasets	Zhang et al. (2014)

Differential privacy can also be applied to social network graphs. The mechanism is discussed by Hay et al. (2009). Evolutionary optimisation techniques can be applied in differential privacy model to explore the search space efficiently (Bhattacharya et al., 2014). Spatial information is obtained for the entire population and communities are formed based on this data. This multi level hierarchical approach aims in optimising the search space efficiently. A top-down specialisation technique is proposed by Zhang et al. (2014) for large-scale data anonymisation, where the datasets are partitioned and anonymised in the first phase and the results are merged for satisfying *k*-anonymity in the second phase.

One of the limits with this approach is un-trusted coding of Mapper. We cannot limit every computation based on an un-trusted code. The major critique about differential privacy is discussed in 'Fool's gold: an illustrated critique of differential privacy' (Bambauer et al., 2013). The work explains that if differential privacy is the only privacy model used for protecting the privacy of data before publishing, the resultant dataset will provide a lot of wrong results to the analyst. In order to avoid this, it would be efficient if this technique is used along with other techniques before publishing the data. Various big data landscapes have been discussed in 'Comprehensive analysis of big data variety landscape' (Abawajy, 2015), which explains about the various dimensions of big data.

Table 6 summarises the various privacy models adopted for streaming data and social network graphs.

7 Conclusions

Big data privacy is a critical component in today's digital world where people, devices and sensors are connected and data is generated, accessed and shared widely with each other. This paper gives a detailed survey on the various privacy models used for anonymising relational databases and big datasets. Various attacks on each privacy model have been studied. This paper gives a good insight on extending the privacy models set for traditional data to work with big data. Each privacy model for big data has been studied using different sets of input like social network graphs and streaming data.

References

'Executive summary, data growth, business opportunities, and the IT imperatives' (2014) [online] http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm (accessed 2015).

'Google moves on from MapReduce, launches cloud dataflow' (2014) [online] http://www.i-programmer.info/news/141-cloudcomputing/7471-google-moves-on-from-mapreduce-launches-cloud-dataflow.html (accessed 2015).

'Tech giants may be huge, but nothing matches big data' (2014) [online] http://www.theguardian.com/technology/2013/aug/23/tech-giants-data (accessed 2014).

'Why 'big data' is a big deal' (2014) [online] http://harvardmagazine.com/2014/03/why-big-data-is-a-big-deal (accessed 2015).

Abadi, D.J. et al. (2003) 'Aurora: a new model and architecture for data stream management', *The VLDB Journal – The International Journal on Very Large Data Bases*, Vol. 12, No. 2, pp.120–139.

Abawajy, J. (2015) 'Comprehensive analysis of big data variety landscape', *International Journal of Parallel, Emergent and Distributed Systems*, Vol. 30, No. 1, pp.5–14.

Abawajy, J.H., Kelarev, A. and Chowdhury, M. (2014) 'Large iterative multitier ensemble classifiers for security of big data', *IEEE Transactions on Emerging Topics in Computing*, Vol. 2, No. 3, pp.352–363.

Adamic, L.A., Lukose, R.M., Puniyani, A.R. and Huberman, B.A. (2001) 'Search in power-law networks', *Physical Review E*, Vol. 64, No. 4, p.046135.

Albert, R., Jeong, H. and Barabási, A.L. (2000) 'Error and attack tolerance of complex networks', *Nature*, Vol. 406, No. 6794, pp.378–382.

Andoni, A., Indyk, P. and Krauthgamer, R. (2008) 'Earth mover distance over high-dimensional spaces', in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.343–352, Society for Industrial and Applied Mathematics.

Bambauer, J., Muralidhar, K. and Sarathy, R. (2013) 'Fool's gold: an illustrated critique of differential privacy', *Vand. J. Ent. & Tech. L.*, Vol. 16, No. 4, p.701.

Bhattacharya, M., Islam, R. and Abawajy, J. (2014) 'Evolutionary optimization: a big data perspective', *Journal of Network and Computer Applications*, DOI: 10.1016/j.jnca.2014.07.032, in press.

Bifet, A. and Frank, E. (2010) 'Sentiment knowledge discovery in twitter streaming data', in *Discovery Science*, pp.1–15, Springer, Berlin, Heidelberg.

Big Data Preliminary Report (2014) ISO/IEC JTC 1, Information Technology.

Big Data Working Group, Cloud Security Alliance (2013) *Expanded Top Ten Big Data Security and Privacy Challenges*.

Bilge, L., Strufe, T., Balzarotti, D. and Kirda, E. (2009) 'All your contacts are belong to us: automated identity theft attacks on social networks', in *Proceedings of the 18th International Conference on World Wide Web*, ACM, pp.551–560.

Campan, A. and Truta, T.M. (2009) 'Data and structural k-anonymity in social networks', in *Privacy, Security, and Trust in KDD*, pp.33–54, Springer, Berlin, Heidelberg.

Cao, J., Carminati, B., Ferrari, E. and Tan, K.L. (2011b) 'Castle: continuously anonymizing data streams', *IEEE Transactions on Dependable and Secure Computing*, Vol. 8, No. 3, pp.337–352.

Cao, J., Karras, P., Kalnis, P. and Tan, K.L. (2011a) 'SABRE: a sensitive attribute bucketization and redistribution framework for t-closeness', *The VLDB Journal*, Vol. 20, No. 1, pp.59–81.

Cardoso, J-F. (1997) 'Infomax and maximum likelihood for blind source separation', *IEEE Signal Processing Lett.*, Vol. 4, No. 4, pp.109–111.

Carrington, P.J., Scott, J. and Wasserman, S. (Eds.) (2005) *Models and Methods in Social Network Analysis*, Vol. 28, Cambridge University Press, USA.

Chandrasekaran, S. and Franklin, M.J. (2002) 'Streaming queries over streaming data', in *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB Endowment, pp.203–214.

Chen, M., Mao, S. and Liu, Y. (2014) 'Big data: a survey', *Mobile Networks and Applications*, Vol. 19, No. 2, pp.171–209.

Clarkson, K.L., Liu, K. and Terzi, E. (2010) 'Toward identity anonymization in social networks', in *Link Mining: Models, Algorithms, and Applications*, pp.359–385, Springer, New York.

Dharmapurikar, S., Krishnamurthy, P., Sproull, T. and Lockwood, J. (2003) 'Deep packet inspection using parallel bloom filters', in *11th Symposium on High Performance Interconnects, 2003. Proceedings*, IEEE, pp.44–51.

Dumbill, E. (2012) *Big Data Now: Current Perspectives*, O'Reilly Radar Team, O'Reilly Media, USA.

Dwork, C. (2008) 'Differential privacy: a survey of results', in *Theory and Applications of Models of Computation*, pp.1–19, Springer, Berlin, Heidelberg.

Dwork, C. (2011) 'Differential privacy', in *Encyclopedia of Cryptography and Security*, pp.338–340, Springer, USA.

Emani, C.K., Cullot, N. and Nicolle, C. (2015) 'Understandable big data: a survey', *Computer Science Review*, August, Vol. 17, pp.70–81.

Fard, A.M. and Wang, K. (2015) 'Neighborhood randomization for link privacy in social network analysis', *World Wide Web*, Vol. 18, No. 1, pp.9–32.

Friedman, A. and Schuster, A. (2010) 'Data mining with differential privacy', in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.493–502.

Fukumori, I. and Malanotte-Rizzoli, P. (1995) 'An approximate Kalman filter for ocean data assimilation: an example with an idealized Gulf Stream model', *Journal of Geophysical Research: Oceans (1978–2012)*, Vol. 100, No. C4, pp.6777–6793.

Fung, B., Wang, K., Chen, R. and Yu, P.S. (2010) 'Privacy-preserving data publishing: a survey of recent developments', *ACM Computing Surveys (CSUR)*, Vol. 42, No. 4, p.14.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A. and Khan, S.U. (2015) 'The rise of 'big data' on cloud computing: review and open research issues', *Information Systems*, January, Vol. 47, pp.98–115.

Hay, M., Li, C., Miklau, G. and Jensen, D. (2009) 'Accurate estimation of the degree distribution of private networks', in *Ninth IEEE International Conference on Data Mining, 2009, ICDM'09*, IEEE, pp.169–178.

Hay, M., Miklau, G., Jensen, D., Towsley, D. and Weis, P. (2008) 'Resisting structural re-identification in anonymized social networks', *Proceedings of the VLDB Endowment*, Vol. 1, No. 1, pp.102–114.

Ho, S.W. and Yeung, R.W. (2010) 'The interplay between entropy and variational distance', *IEEE Transactions on Information Theory*, Vol. 56, No. 12, pp.5906–5929.

Katal, A., Wazid, M. and Goudar, R.H. (2013) 'Big data: issues, challenges, tools and good practices', in *2013 Sixth International Conference on Contemporary Computing (IC3)*, IEEE, pp.404–409.

Leskovec, J., Rajaraman, A. and Ullman, J. (2011) *Mining of Massive Datasets*, Cambridge University Press, UK.

Li, C. et al. (2010) 'Optimizing linear counting queries under differential privacy', *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ACM.

Li, J., Ooi, B.C. and Wang, W. (2008) 'Anonymizing streaming data for privacy protection', in *ICDE 2008. IEEE 24th International Conference on Data Engineering*, IEEE, pp.1367–1369.

Li, N., Li, T. and Venkatasubramanian, S. (2007) 't-closeness: privacy beyond k-anonymity and l-diversity', in *ICDE 2007. IEEE 23rd International Conference on Data Engineering, 2007*, IEEE, pp.106–115.

Lodha, S. and Thomas, D. (2008) 'Probabilistic anonymity', in *Privacy, Security, and Trust in KDD*, pp.56–79, Springer, Berlin, Heidelberg.

Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M. (2007) 'l-diversity: privacy beyond k-anonymity', *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 1, No. 1, p.3.

McSherry, F. and Talwar, K. (2007) 'Mechanism design via differential privacy', in *48th Annual IEEE Symposium on Foundations of Computer Science, 2007, FOCS'07*, IEEE, pp.94–103.

Microsoft Corporation, 'Differential Privacy for Everyone' (2012) [online] http://www.microsoft.com/en-us/download/details.aspx?id=35409 (accessed 2015).

O'Callaghan, L., Meyerson, A., Motwani, R., Mishra, N. and Guha, S. (2002) 'Streaming-data algorithms for high-quality clustering', in *ICDE*, IEEE, p.0685.

Roy, I., Setty, S.T., Kilzer, A., Shmatikov, V. and Witchel, E. (2010) 'Airavat: security and privacy for MapReduce', in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10)*, Vol. 10, pp.297–312.

Sagiroglu, S. and Sinanc, D. (2013) 'Big data: a review', in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, IEEE, pp.42–47.

Sarathy, R. and Muralidhar, K. (2011) 'Evaluating Laplace noise addition to satisfy differential privacy for numeric data', *Transactions on Data Privacy*, Vol. 4, No. 1, pp.1–17.

Sweeney, L. (2002) 'k-anonymity: a model for protecting privacy', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp.557–570.

Tene, O. and Polonetsky, J. (2013) 'Big data for all: privacy and user control in the age of analytics', *11 Nw. J. Tech. & Intell. Prop. 239*, Vol. 11, No. 5, pp.240–273.

Watts, D.J. and Strogatz, S.H. (1998) 'Collective dynamics of 'small-world' networks', *Nature*, Vol. 393, No. 6684, pp.440–442.

Wu, X., Zhu, X., Wu, G.Q. and Ding, W. (2014) 'Data mining with big data', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 1, pp.97–107.

Yu, H., Kaminsky, M., Gibbons, P.B. and Flaxman, A. (2006) 'SybilGuard: defending against Sybil attacks via social networks', *ACM SIGCOMM Computer Communication Review*, Vol. 36, No. 4, pp.267–278.

Zakerzadeh, H. and Osborn, S.L. (2011) 'Faanst: fast anonymizing algorithm for numerical streaming data', in *Data Privacy Management and Autonomous Spontaneous Security*, pp.36–50, Springer, Berlin, Heidelberg.

Zhang, X., Yang, L.T., Liu, C. and Chen, J. (2014) 'A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 2, pp.363–373.

Zhou, B. and Pei, J. (2011) 'The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks', *Knowledge and Information Systems*, Vol. 28, No. 1, pp.47–77.

Zhou, B., Han, Y., Pei, J., Jiang, B., Tao, Y. and Jia, Y. (2009) 'Continuous privacy preserving publishing of data streams', in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, ACM, pp.648–659.