Aalto University
School of Electrical Engineering
Department of Communications and Networking

Shankar Lal

# Privacy Preserving Log File Processing in Mobile Network Environment

Master's Thesis
Espoo, April 30, 2015

| | |
|---|---|
| Supervisors: | Professor Tarik Taleb<br>Aalto University |
| Advisor: | Dr. Ian Oliver D.Sc. (Tech.)<br>Nokia Networks Oy |

Aalto University
School of Electrical Engineering

Aalto University
School of Electrical Engineering
Department of Communications and Networking

ABSTRACT OF
MASTER'S THESIS

|  |  |  |  |
|---|---|---|---|
| **Author:** | Shankar Lal |  |  |
| **Title:**<br>Privacy Preserving Log File Processing in Mobile Network Environment |  |  |  |
| **Date:** | April 30, 2015 | **Pages:** | 64 |
| **Major:** | Networking Technology | **Code:** | S.38 |
| **Supervisors:** | Professor Tarik Taleb<br>Aalto University |  |  |
| **Advisor:** | Dr. Ian Oliver D.Sc. (Tech.)<br>Nokia Networks Oy |  |  |

Network operators collect huge amount of user data flowing on their networks. The purpose of one specific type of data collection is to understand the network usage pattern and network monitoring for anomaly detection etc.

Network operators share this data confidentially with equipment manufactures and vendors to make statistical analysis over the data to find the unusual behavior e.g. Malware traffic etc. Since this data contains user identifiable information, there is a need to anonymize such data sets to protect user privacy and to comply with privacy laws. Thus, protecting user privacy is top priority for the network operator and they are also legally obliged to do so.

The analysis presented in this thesis work is based on data sets, obtained from network traces (LTE, IP). For some usage, these data sets are required to be anonymized. There are various classes of algorithms to achieve this e.g. Encryption, Hashing, Field suppression, $\kappa$-anonymity, $\ell$-diversity, t-closeness, Differential Privacy etc. Since none of these algorithms are individually perfect for anonymizing data completely, these should be used in conjunction with each other to get the better level of anonymization without compromising the usability and semantic integrity of the data.

Differential privacy and $\ell$-diversity algorithms are implemented in this thesis for anonymization of network traces along with process pipeline for them. Comparison of statistical properties of both original and perturbed data is also presented to check the efficiency of anonymization techniques.

**Keywords:** Data Privacy, Differential Privacy, $\ell$-diversity, $\kappa$-anonymity

**Language:** English

# Acknowledgements

My deepest gratitude goes to my instructor **Dr. Ian Oliver** at Nokia Networks providing unending support and sparing his valuable time each and every day to make this thesis possible.

Great thanks to my manager **Mr. Gabriel Waller** at Nokia Networks for giving me the opportunity to work in his team and also supporting me throughout this work.

Special thanks to my supervisor professor **Mr. Tarik Taleb** at Aalto University for taking time to review this thesis and also providing his valuable feedback.

Most importantly, I would like to express my heart-felt gratitude to all my colleagues and family members, specially to my daughter **Jheel** to whom this thesis is dedicated. She has been a constant source of love, concern, support and strength all these years.

Espoo, April 30, 2015

Shankar Lal

**To my charming daughter *Jheel***

# Contents

# List of Tables

# List of Figures

# Abbreviations and Acronyms

| | |
|---|---|
| PII | Personally Identifying Information |
| PET | Privacy Enhancing Techniques |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| ICMP | Internet Control Message Protocol |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |
| FTP | File Transfer Protocol |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| HIV | Human Immunodeficiency Virus |
| PLMN | Public Land Mobile Network |
| SGSN | Serving GRPS Support Node |
| GGSN | Gateway GPRS Support Node |
| RNC | Radio Network Controller |
| GTP | GPRS Tunneling Protocol |
| UCP | Universal Computer Protocol |
| SMPP | Short Message Peer-to-Peer |
| IMEI | International Mobile Station Equipment Identity |
| IMSI | International Mobile Subscriber Identity |
| MSBs | Most Significant Bits |
| LSBs | Least Significant Bits |

# Chapter 1

# Introduction

Huge amount of user data is being collected and processed for various purposes such as measuring network performance, network anomaly detection etc. This data is gathered from users and hence contains some personal data or personal identifying information (PII)[6]. This data is often utilized for various business purposes without taking care of user privacy which opens the window of opportunity for adversaries to correlate and obtain in depth knowledge about the user profiles.

Mobile operators are obliged by law to protect their customer's privacy. The data collected by mobile operators contains sensitive information about their customers. There is a requirement for the mobile operators to share this data with other partners and equipment manufactures for detecting abnormal network behavior, or anomaly in the network traffic. Sharing the original data of the customers expose them to inference attacks or possible re-identification of the users, thus there arrive a need to anonymize this kind of network data.

Anonymizing network data is not an easy task considering its semantics and properties of its fields i.e. continuous and discrete fields and it is also difficult to balance between the level of anonymization and usefulness of the output data. One needs to parameterize the privacy preserving algorithms to tune the trade-off between data utility and data privacy.

## 1.1 Objectives

The main objectives of this thesis work are as follows:

- To analyze the network trace data from IP and mobile network environment and make statistical analysis to observe its properties.

- To identify the sensitive and non-sensitive fields in the trace data and to observe if combination of some of these fields can provide information about users or traffic behavior.
- To demonstrate privacy preserving algorithms such as Differential privacy and $\ell$-diversity, to anonymize different fields of network trace data and compare their statistical properties.

## 1.2 Structure

This thesis is divided into following chapters.

Chapter 2 sets the basis for understanding the main objectives of this work. It provides fundamental concepts about privacy along with its definition, need for privacy in user data and some basic understanding about the meaning of the terms Personal Data and PII (Personally Identifiable Information). Chapter 3 provides understating about the state of the art privacy preserving algorithms i.e. Differential privacy, $\kappa$-anonymity and $\ell$-diversity with detailed examples. The chapter also lists the parameters used in these algorithms by which level of anonymization can be controlled. Chapter 4 presents a sample of network trace and discuss about the information content in it. The chapter also gives detailed statistical analysis over each field of the data using various visualization methods and provides results in the form of various statistics to present better overview of the traffic pattern and nature of data.

Chapter 5 describes some novel techniques of noise addition which can be employed on continuous fields such as numerical and temporal. Various noise addition techniques are used to perturb these fields and corresponding results are compared using statistical parameters e.g. mean and standard deviation and five number summary. Chapter 6 describes an anonymization technique for discrete fields of the data sets e.g. protocol field, which is anonymized using $\ell$-diversity technique. Chapter 7 presents the conclusion and future outlook for extension of this work.

# Chapter 2

# Background

This chapter presents the basis for understanding the data privacy and its need in user's data. In many cases, Privacy results in information loss hence the trade-off between the privacy and data utility is also need to be discussed. The chapter also covers the discussion regarding personal and sensitive information present in user's data in both general and network domain.

## 2.1 Definition of privacy

Privacy is recognized as fundamental human right which is protected in various regions in the world. There has been lot of discussion amongst lawyers, philosophers and other social scientists in particular Daniel J. Solove, Helen Nissenbaum [7] about privacy issues. With the advent of information and computer science technologies, where data is being collected at larger scale, raises concerns about the privacy of an individual [8].

Every human needs privacy in his life as his fundamental right which includes choice of living life in his or her own determined way. Based on this fundamental definition of privacy, data privacy can be defined as primary right to the data or information regarding some person. This data or information might be created either by that person or by someone else by just observing or by processing over that information.

The main idea behind privacy is that any individual should be independent to take any decisions regarding his life without any interference. The same idea follows in data privacy that any individual should have control over the use of his information. These two ideas relate and support each other within the privacy domain since as we humans need privacy in our lives, our data/information also demands privacy [9].

## 2.2 Need for privacy

Most of the government agencies and other organizations, such as population registrar or tax offices, collect personal data which is mandatory for example, census data for counting on population and tax data of the citizens etc. The census data, voter registration data or hospital data can be published for research and statistical analysis purposes, in public interest. Some data is however collected for the altruistic purpose to diagnosis some problems i.e. Malware infected network trace from the internet to find unusual behaviors and share the results with the public or to raise the awareness of the problem [10].

Due to the growing use of internet based applications, which have digitized most of the businesses, have also introduces drawbacks such as looking at the usage patterns of a social media profile of a person can help to predict his behavior, interests and sexual orientation, or guessing if a woman is pregnant based on what she is purchasing online is reality nowadays.

There are, however, some business needs to collect customers data. The data collection allows companies to know their customers in better way and treat them accordingly, this is done by extracting useful insight from customer's data which is required for behavioral and targeted marketing. This helps companies to identify what kind of product to offer to customers in future, which in turns provides a great value of personalized products to the customers. [11].

## 2.3 Sharing network traces

Stakkato's attack [12] is known for worldwide cyber-attack which was carried out for almost two years and effected many popular government websites such as US military, NASA and many universities. Network traces collected by authorities contained the fingerprints of such attack but they also contained the sensitive information about that organization and sharing the raw network trace could lead to the unintended disclosure of the information. A network trace can contains all network traffic which includes websites visited by the users, location of their emails and any other credentials which are not secured by encryption. Thus sharing raw network trace is big problem.

Data sharing can provide the benefit for detecting fingerprints of popular security attacks such as DDoS attack or Malware detection. The information flow from different sources can enable wider scope of analysis of the data but as a matter of fact sharing of network traces between the research organizations is very limited since network traces contain personally identifiable information of the people inside the organization whose privacy might be on the stake [13].

This is the same case for mobile network operators who collect their customer data to analyze and troubleshoot network problems or detecting anomalies. There are however, times when this data need to be shared with other partners or third parties. Since this data can reveal personal identifying information like IP address or location data, it will be unwise to share raw data with third parties, who might process it to identify some individuals. Hence there arises a need to anonymize such form of data, this is where role of Privacy Enhancing Technologies (PET) comes into play [14].

## 2.4 Cases of popular privacy breaches

### 2.4.1 Netflix user privacy breach

Most popular privacy breaches resulted when Netflix released their user's data of movie rating of their 500,000 subscribers for the purpose of improving their movie recommendation engine based on individual preferences. Although, they anonymized the data set by replacing the real name of the subscribers with random numbers but later it became possible for researchers to reconstruct the identities of some individuals by mapping them to publically available IMDB database [15]. Movie rating can prove to be sensitive information about some person as it can reveal some person's view about politics, religion or homosexuality.

### 2.4.2 AOL anonymous data

Similar case happened with AOL, when they released user search queries to internet community for research purpose but later they turned it down because many of users were identified by their search queries. As the number search queries collected for specific user increased, it even became easier to trace back to that user. A 60 years old widow woman living in Georgia state in USA having three dogs was identified by her search queries about '60 years old single man' and 'dogs that urinates on everything' [16].

### 2.4.3 Identification of medical records of former governor of Massachusetts

One type of inference attack (a kind of attack for illegitimately gaining knowledge about a target by analyzing the data) was demonstrated by Latanya Sweeney [17], where she exposed the medical condition of the governor of Massachusetts by mapping two kinds of data. one kind of data was obtained from Group Insurance Commission (GIC) who were responsible for health insurance of government staff. Other kind of data was obtained from Massachusetts voter registration department. Latanya Sweeney demonstrated that the tuple zip code, birth date and sex were shared attributes in both the data sets which caused the re-identification of governor of Massachusetts .

## 2.5 Trade-off between data utility and data privacy

Publishing data provides useful information for the researchers but exposes the risk on people's privacy where attacker can launch an inference attack by mapping the data set with other publically released data set to re-identify some individuals.

Many privacy mechanisms transform data to some other form i.e. hashing, encryption etc. in order to ensure privacy but as a result information loss is high in these methods. Modifying the representation of the data by changing it to some other form might yield better privacy but results in great loss of data utility. Data utility is defined as the measure of the usefulness of the data to its consumers. The general idea about privacy and utility trade-off in network domain is the more data is removed or obfuscated from a network trace file, the better privacy can be assured to the users while of data will less useful to the researchers who make analysis over it [18].

The privacy and utility are inversely proportional to each other and it's difficult to calculate the balance point between them. A balance point between data privacy and information loss depends on used cases and applications for which anonymized data will be used [19].

## 2.6 Personal Data and PII (Personally Identifying Information)

The terms Personal Data and Personally Identifiable Information are synonymous and can be interchanged in many contexts. The term, Personal data comes from EU directives and legislation [20] and term personally identified information (PII) comes from US privacy laws [21]. The definition of both terms coincide each other and roughly points to the same meaning. In this thesis, term Personal Data is used for convenience. European Union published new General publication Data Protection Regulation (GDPR) which defined personal data as:

*'Any type of information which can be utilized with other set of information or on its own to discover or recognize some individuals, locations or from which any other useful piece of personal information can be extracted'* [22].

And

*'Personal data is any information relating to an individual, whether it relates to his or her private, professional or public life. It can be anything from a name, a photo, an email address, bank details, posts on social networking websites, medical information, or a computer's IP address.'*

Examples of personal data in EU and US privacy laws are as below:

- Full name
- Home address
- Social Security Number
- Date of birth
- Telephone number

There are two types of attributes, which often appear in the data sets namely sensitive attributes and non-sensitive attributes. The definition is given below.

Sensitive attributes are kind of attributes which can directly identify an individual such as health condition, email addresses etc. So the association between an individual and his sensitive attributes should not be disclosed i.e. it would not be permissible to hospitals to disclose which patient has cancer disease. All other attributes apart from sensitive attributes are considered as non-sensitive attributes as they are shared by many people for example, gender, age, zip code and others.

Combination of some non-sensitive attributes in the data set is known as Quasi-identifiers [23]. A Quasi identifier can be used to re-identify individuals from the data set by mapping it to the external data. An example of Quasi identifier is the tuple of Age, gender and zip code in voter registration list. In network trace, quasi identifiers can be a tuple of time-stamp, protocol and packet length.

## 2.7 IP address as personal data

The discussion on IP address being personal data is very broad. In European privacy laws IP address is personal data (UK is exception here) while in US laws it isn't. Here are few arguments that why and why not an IP address can be considered as personal data [24].

**IP address as personal data**

- IP address can refer to a person, an entity or a place via geo-location coordinates [25].
- If the IP address is assigned by an ISP (Internet Service Provider) under subscriber account then it can be used to identify the subscriber and hence it is personal data.
- Even though IP address changes most often but ISPs keep the record of internet activities of the customers and logs the data that who is assigned what IP and when, can enable the tracking of some individuals.
- Most Internet websites create profile of their customer's activities if they have been assigned a static IP address. The profile contains customer's browsing behavior and other online activities on that website. Some websites collect demographic information about their users by IP address to show the number of visitors based on countries.

**IP address as non-personal data**

- An IP address in isolation is not personal data as it targets a machine and not an individual, however combination of some data fields with IP address can be considered as personal data.
- An IP address is mostly not permanent and keeps changing. IP address assigned to one computer may have been assigned to some other computer one week ago.
- Several computers using NAT (Network Address Translation)or DHCP (Dynamic Host Configuration Protocol) service to access internet via only one public IP, makes it hard to trace back the user.

## 2.8 Summary

This Chapter gives some background knowledge to understand the data privacy. The question of why privacy is needed in user data is also discussed in detail. In addition to that, this chapter lists some of the most popular privacy breaches occurred in user data in past i.e. Netflix and AOL attacks. This chapter also explains most commonly used terms in privacy field i.e. PII and personal data. In the end, a brief explanation is presented about whether an IP address being personal data or not with some arguments on both sides.

# Chapter 3

# Privacy preserving algorithms

This chapter introduces some data anonymization techniques namely differential privacy, $\kappa$-anonymity and $\ell$-diversity. At first, data privacy process is discussed along with some fundamental concepts of data anonymization and noise addition. Later in this chapter, fundamental concepts behind these techniques are discussed in details with the examples of network data.

## 3.1 Data privacy process

Data privacy process is described by figure 3.1. The process of data anonymization starts from removing direct identifiers (hiding/removal of sensitive attributes) and then perturbing the data fields by using privacy preserving algorithms. Noise is usually added to the data field for perturbation purpose. In the end, the sensitive data fields are completely suppressed to get the final anonymized data Set [26].

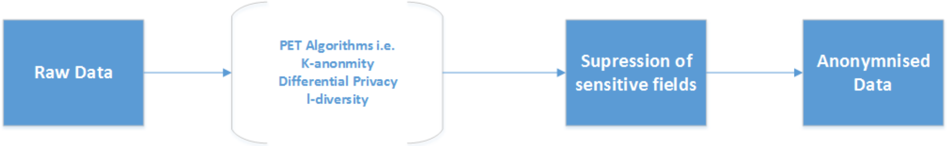


Figure 3.1: A typical flow chart of privacy preserved network trace.

As mentioned in previous chapter that inferences can be made to some individuals by mapping the anonymized data with another set of released data therefore noise is added to the original data fields after suppressing personal data to increase further confidentiality of the data set.

## 3.2 Foundational concepts

Data anonymization is performed to protect the privacy of the people whose information is present in that data set so that it can be shared with third parties, other organizations or even with data agencies for making statistical analysis and post-anonymization observations on it. To avoid any unintended disclosure, the data should be anonymized in such a way that information of the people in the data sets remains completely anonymous. Traditionally, data anonymization is performed by hashing the data fields. The hashing functions get the set of data and change it to the fixed length representation. Thus hashing only changes the representation of the data but not the information content in it. Due to significant advent in computing power which opens possibilities to recover original values from hashes and also through pre-computed hash tables also known as rainbow tables, it is possible to recover original data from the hashed data. Considering these possibilities, it would be difficult to rely on such methods [19] [23].

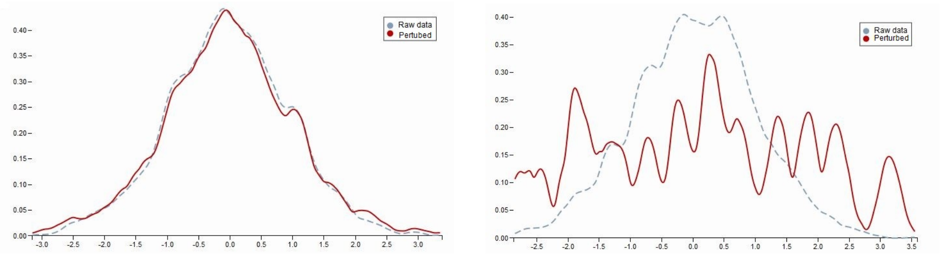
### 3.2.1 Noise addition

The purpose of the noise addition is to preserve the overall distribution of the data, while still anonymizing the individual records to prevent the exposure of original data values. Adding noise to original data values provides defense against inference attacks which involves exact matching of the data for some particular attributes, obtained from publically released data [27].

Noise addition can be additive or multiplicative. In Additive noise, a random number is added with each element of the data, while in multiplicative noise, random number is multiplied with every element of data set. Typically, random value used for noise addition is chosen from normal distribution with zero mean and small standard deviation [26].

Figure 3.2 (a) depicts the controlled noise addition in the original values such that the perturbed data distribution almost follows the original pattern, while figure 3.2 (b) shows the overmuch noise addition in original data such that the pattern of the original distribution cannot be identified from the perturbed data. It is obvious to note that controlled noise addition preserves the statistical properties of the original data while overmuch noise addition completely ruins the original data and makes it completely useless for analysis.

Many privacy preserving algorithms have been introduced in recent years. In this work, differential privacy, $\kappa$-anonymity and $\ell$-diversity algorithms are discussed with their salient features and used cases.



(a) Examples of controlled noise addition (b) Examples of uncontrolled noise addition

Figure 3.2: Examples of controlled and uncontrolled noise addition [1]

## 3.3 Differential privacy

Differential privacy is a technique to obfuscate the data fields using Laplace noise addition and was introduced by Cynthia Dwork in 2006. Differential privacy mechanism assures that output of any analysis is not greatly affected if a single record in a data set is added or removed, so the mechanism provides guarantee that an input of any person either included or excluded in the data set would not change the final distribution of the result significantly, even if the information of an individual is perturbed, the output is nearly unchanged. Thus differential privacy guarantees the information privacy of an individual in a data set [28].

### 3.3.1 Mathematical definition

Consider two neighboring data sets D1, D2. The neighboring data sets are the data sets which differ only from one entry, one row or one record. They produce output S when mechanism K which satisfies $\epsilon$-differential privacy, is applied. The mechanism K, which might be a randomized function to add jitter to some data field, fulfills the condition about the information disclosure related to any individual. Information disclosure is the leakage of information about any individual violating his/her privacy. Differential privacy algorithm states that probability of data set D1 producing output S is nearly the same as the probability of data set D2 producing same output [10] [28].

Dwork's definition of differential privacy is following:

A randomized function $\mathcal{K}$ satisfies *$\epsilon$-differential privacy* if for all pairs of adjacent databases $D'$ and $D''$, and all $S \subseteq Range(\mathcal{K})$,

$$Pr[\mathcal{K}(D') \in S] \leq e^{\epsilon} \times Pr[\mathcal{K}(D'') \in S] \tag{3.1}$$

Here, $\epsilon$ is known as privacy parameter. The value of $\epsilon$ corresponds to the strength of the privacy. This value is not fixed and depends on the used case [10]. Differential privacy uses Laplace noise to perturb data fields. Laplace noise addition depends on the sensitivity and privacy parameter $\epsilon$. These concepts are discussed in next sections.

### 3.3.2 Sensitivity

Sensitivity is the maximum amount the outputs that can be perturbed by adding or removing some records from the data sets. This measures how much output can be altered if a response is either included or excluded from the result.

Sensitivity of the function is connected to differential privacy since in order to make a query differentially private, the amount of noise to be added in the query is proportional to its sensitivity. To get an idea about the value of sensitivity, if a data set is queered that 'how many rows have certain property will yield sensitivity value of 1' [28].

### 3.3.3 Laplace Noise

The Laplace mechanism is used for adding noise to the data sets which satisfies the Laplace statistical distribution. Below is the mathematical formula for adding random noise using Laplace mechanism

$$b = \Delta f/\epsilon$$

Where b is known as Laplace noise, $\Delta$f is the sensitivity value and $\epsilon$ is privacy parameter. When the query on function f with data set X is made, the mechanism K responds with

$$f(X) + \Delta f/\epsilon$$

By meaning, random noise with the magnitude Lap $(\Delta f/\epsilon)$ will be added to all elements of data set $X$. Here, if the value of $\epsilon$ is decreased, the amount of noise will increase and vice versa, when the value of sensitivity is fixed, for example if the sensitivity value is 1 and if $\epsilon$ is taken as 0.1, the magnitude of noise added will be 10. Similarly if $\epsilon$ is 0.01, the noise magnitude will increase to 100 [29].

Differential privacy is state of art technique for preserving privacy but it also suffers from some weaknesses for example with the increase of small value of $\epsilon$, statistical properties of the data like mean, standard deviation etc, can change dramatically so it can be challenging to maintain good utility of the data. Figure 3.3 shows the probability density function of Laplace distribution with mean 0 and different values of scale parameter b.

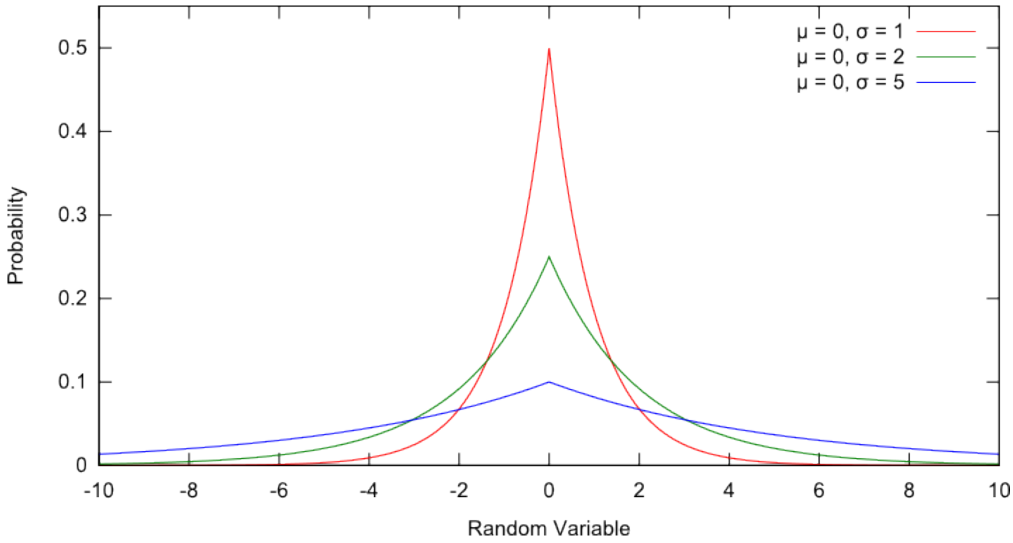


Figure 3.3: Example of Laplace Distribution [2]

## 3.4 $\kappa$-anonymity

$\kappa$-anonymity is the data anonymizing technique, introduced by Latanya Sweeney in 2002, by which personally identifiable information is removed from the data set or changed into some state for useful analysis and processing. $\kappa$-anonymity technique had been mostly used in anonymizing medical data sets [17].

The main idea behind $\kappa$-anonymity is to form a group of k records or rows in such a way that they possess similar features, for example, group of rows that look identical and indistinguishable. In the same way, in $\kappa$-anonymized data set each row is indistinguishable from at least $\kappa$-1 other rows within the data set for some set of identifying attributes.

Once can think of $\kappa$-anonymity as in example that if someone tries to find an individual in the released data but the only information available about that individual is his/her birth date and gender. There will be k other persons in the released data with the same birth data and gender thus making linkage ambiguous [17]. The three well known methods used to achieve these $\kappa$-anonymity attributes, are generalization, suppression and perturbation.

In generalization, the original value of the data field is replaced by a general categorical value that only contains small amount of information for example replacing last octet of IP address with * to hide user identifying information. In suppression technique, the data fields are omitted and replaced by a null value, this is done with some sensitive fields. In contrary, perturbation technique doesn't omit the data field but obfuscate them with new value by adding noise to it, for example, perturbing the original values of times-tamp field in network trace by adding random noise. The figure 3.4 portrays above concepts.

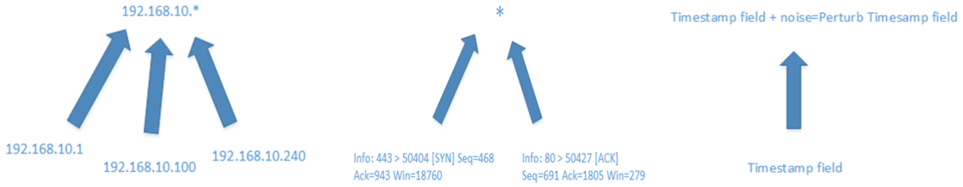
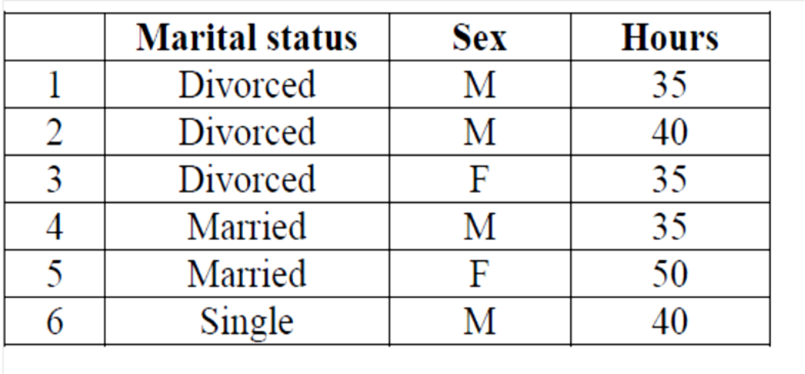

192.168.10.*
192.168.10.1
192.168.10.100
192.168.10.240
*
Info: 443 > 50404 [SYN] Seq=468 Ack=943 Win=18760
Info: 80 > 50427 [ACK] Seq=691 Ack=1805 Win=279
Timestamp field + noise=Perturb Timesamp field
Timestamp field

Figure 3.4: Example of generalization, suppression and perturbation.

### 3.4.1 Example of $\kappa$-anonymity

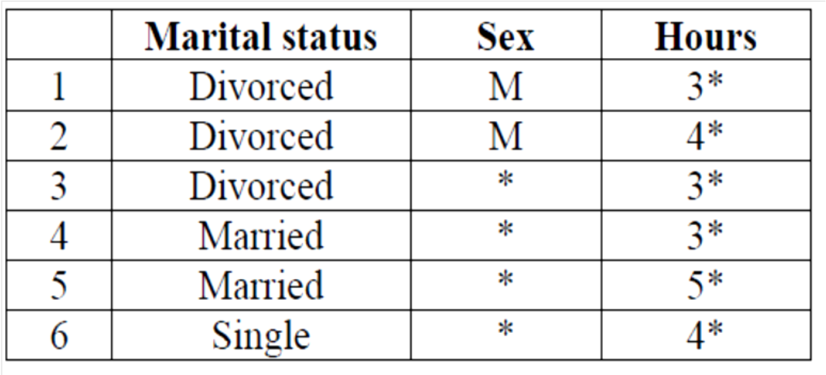
Consider a data table containing information about some individuals about their age, sex, marital status and working hours shown in figure 3.5. In this data, even though direct identifiers like name, email address, social security number etc. are removed, but combination of given fields can lead to the linkage attack with the help of other data set. Hence, there is need to $\kappa$-anonymize this data set using the generalization and suppression techniques such that any inference to any individual cannot be made.



| | Marital status | Sex | Hours |
|---|---|---|---|
| 1 | Divorced | M | 35 |
| 2 | Divorced | M | 40 |
| 3 | Divorced | F | 35 |
| 4 | Married | M | 35 |
| 5 | Married | F | 50 |
| 6 | Single | M | 40 |

Figure 3.5: A data table containing information of the employees.

2-anonymous version of the original data set is shown in figure 3.6, where k=2. It is easy to notice that some fields are suppressed completely while others are suppressed to certain level so that information loss can be controlled thus making data set useful for analysis [3].



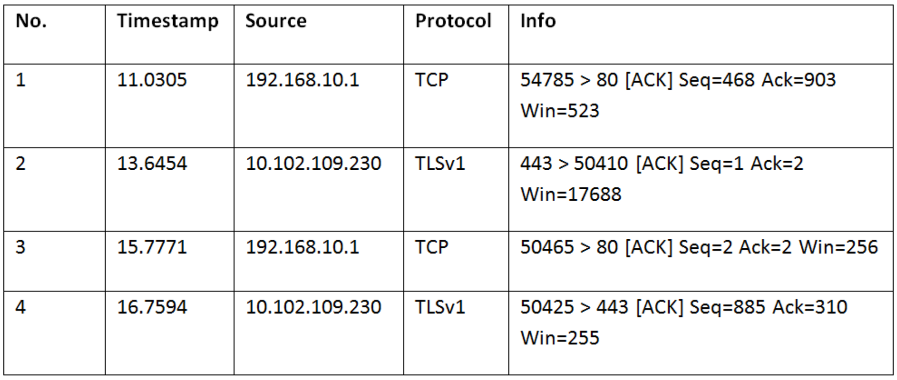
| | Marital status | Sex | Hours |
|---|---|---|---|
| 1 | Divorced | M | 3* |
| 2 | Divorced | M | 4* |
| 3 | Divorced | * | 3* |
| 4 | Married | * | 3* |
| 5 | Married | * | 5* |
| 6 | Single | * | 4* |

Figure 3.6: 2-anonymous data table [3].

There are some unique records like number 6 in the above data set, which are hard to remove even though they are anonymized. In large data set, these records are considered as outliers which might lead to re-identification of the records. If these types of records are small in number in the data set, they can be fully suppressed, which will have only negligible effect on the anonymized data set [23].

### 3.4.2 Example of $\kappa$-anonymity in network data

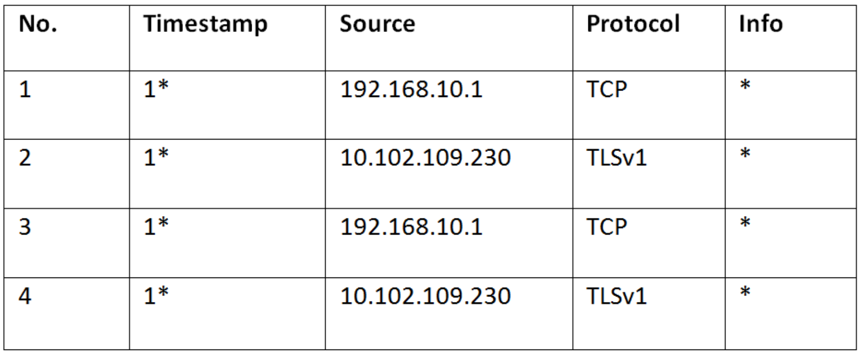
Consider a sample of network data containing some fields like time stamp, source IP, protocol and info field as shown in figure 3.7. The combination of certain fields in this data set can provide the additional information that can lead to identification of some individuals.



| No. | Timestamp | Source | Protocol | Info |
|---|---|---|---|---|
| 1 | 11.0305 | 192.168.10.1 | TCP | 54785 > 80 [ACK] Seq=468 Ack=903 Win=523 |
| 2 | 13.6454 | 10.102.109.230 | TLSv1 | 443 > 50410 [ACK] Seq=1 Ack=2 Win=17688 |
| 3 | 15.7771 | 192.168.10.1 | TCP | 50465 > 80 [ACK] Seq=2 Ack=2 Win=256 |
| 4 | 16.7594 | 10.102.109.230 | TLSv1 | 50425 > 443 [ACK] Seq=885 Ack=310 Win=255 |

Figure 3.7: A sample of network data.

The 2-anonymous version of the original data set is shown in figure 3.8. In order to get 2-anonymous data set, an Info field is completely suppressed, while time-stamp field is partially suppressed to get some data utility. 2-anonymous data set implies that there are at-least two similar and indistinguishable records present in the data set.



| No. | Timestamp | Source | Protocol | Info |
|---|---|---|---|---|
| 1 | 1* | 192.168.10.1 | TCP | * |
| 2 | 1* | 10.102.109.230 | TLSv1 | * |
| 3 | 1* | 192.168.10.1 | TCP | * |
| 4 | 1* | 10.102.109.230 | TLSv1 | * |

Figure 3.8: 2-anonymous network data table

### 3.4.3 Attacks on $\kappa$-anonymity

$\kappa$-anonymity technique is vulnerable to inference attacks if the attacker has some prior knowledge about the data. For example, in network data set, if the attacker knows the IP address of somebody, he can know that what kind of traffic is being generated by that user, this is called background knowledge attack. If IP addresses are anonymized, then it can difficult to find such traffic patterns but again consider a case when there are only few types of protocol types present in the data set, then it can be easy for attacker to correlate a user from anonymized IP addresses against the protocol field. Due to these attacks, a new technique called $\ell$-diversity was introduced and preferred over $\kappa$-anonymity [30].

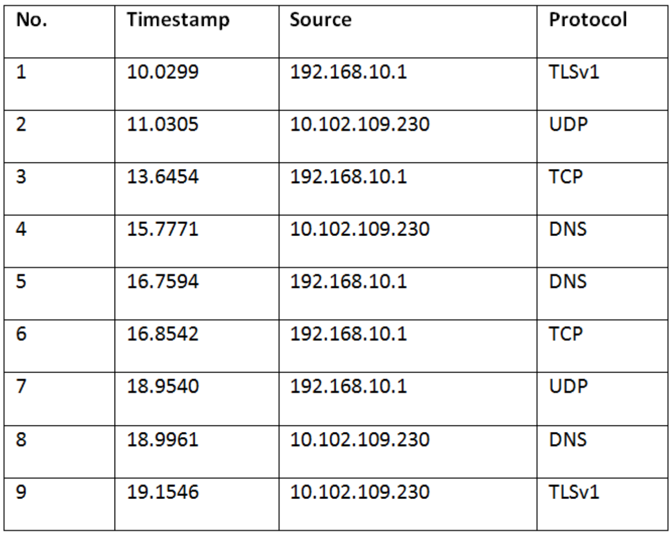
## 3.5 $\ell$-Diversity

It is shown earlier that $\kappa$-anonymity suffers from some attacks so there is need for a robust notion of privacy in data sets. $\ell$-Diversity technique was introduced to address this problem. The main idea behind this technique is data sets should be divided into blocks so that each block should contain diverse values of sensitive attributes of the data set. Machanavajjhala et al. defines $\ell$-diversity principle as below:

"A q$*$-block is $\ell$-diverse if contains at least $\ell$ "well-represented" values for the sensitive attribute S. A table is $\ell$-diverse if every q$*$-block is $\ell$-diverse." [31]

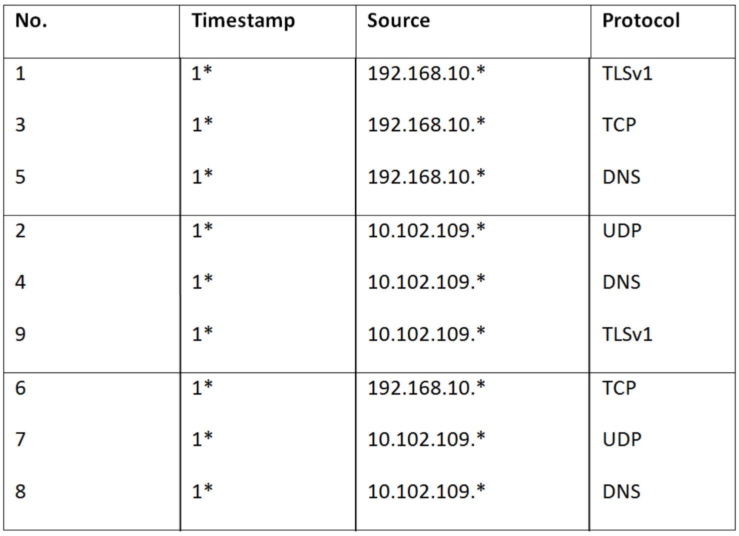
### 3.5.1 Example of $\ell$-diversity in network data

$\ell$-Diversity ensures that every block of data in the data set should have "well-represented" values or in other words, at lease $\ell$ diverse values for sensitive attributes within each equivalence class should be present. Figure 3.9 shows a sample of some fields from network data set. 3-diverse version of this data sample is shown in figure 3.10.



| No. | Timestamp | Source | Protocol |
|---|---|---|---|
| 1 | 10.0299 | 192.168.10.1 | TLSv1 |
| 2 | 11.0305 | 10.102.109.230 | UDP |
| 3 | 13.6454 | 192.168.10.1 | TCP |
| 4 | 15.7771 | 10.102.109.230 | DNS |
| 5 | 16.7594 | 192.168.10.1 | DNS |
| 6 | 16.8542 | 192.168.10.1 | TCP |
| 7 | 18.9540 | 192.168.10.1 | UDP |
| 8 | 18.9961 | 10.102.109.230 | DNS |
| 9 | 19.1546 | 10.102.109.230 | TLSv1 |

Figure 3.9: A data table of network traffic



| No. | Timestamp | Source | Protocol |
|---|---|---|---|
| 1 | 1* | 192.168.10.* | TLSv1 |
| 3 | 1* | 192.168.10.* | TCP |
| 5 | 1* | 192.168.10.* | DNS |
| 2 | 1* | 10.102.109.* | UDP |
| 4 | 1* | 10.102.109.* | DNS |
| 9 | 1* | 10.102.109.* | TLSv1 |
| 6 | 1* | 192.168.10.* | TCP |
| 7 | 1* | 10.102.109.* | UDP |
| 8 | 1* | 10.102.109.* | DNS |

Figure 3.10: 3-diverse data

### 3.5.2 Limitation of $\ell$-diversity

Although $\ell$-diversity overcomes the limitations of $\kappa$-anonymity but it doesn't provides the protection from identity disclosure and attribute disclosure in some cases. $\ell$-diversity suffers from two major attacks known as skewness attack and similarity attack.

In skewness attack [32], the distribution of sensitive attributes is skewed, meaning that for a 2-diverse table if there are 99 percent negative values and 1 percent positive values of sensitive attributes, which could happen for a test result i.e. HIV having only positive and negative values. In this case, $\ell$-diversity seems to be useless since there are 99 percent chances of guessing the correct disease of the victim. Similarly, for a 2-diverse equivalence class, any individual can be identified by the probability of 50 percent if there are only two values of sensitive attribute .

In similarity attack [32], the values of sensitive attributes are semantically same since $\ell$-diversity doesn't consider the semantic of the values of sensitive attributes, for example in a block of $\ell$-diverse data set, the sensitive attribute refer to same equivalence class i.e. in medical data set, diverse table containing disease values as stomach cancer, gastritis and ulcer still provides potential information to the attacker. So its a problem with $\ell$-diversity that it doesn't understand the meaning of sensitive attributes.

## 3.6 Summary

In this chapter, some basic concepts regarding data anonymization and noise addition are discussed with flow chart of data privacy process. Three techniques of data anonymization i.e. differential privacy, $\kappa$-anonymity and $\ell$-diversity are presented along with examples of network data. Some attacks on $\kappa$-anonymity techniques and limitation of $\ell$-diversity technique are also discussed.

# Chapter 4

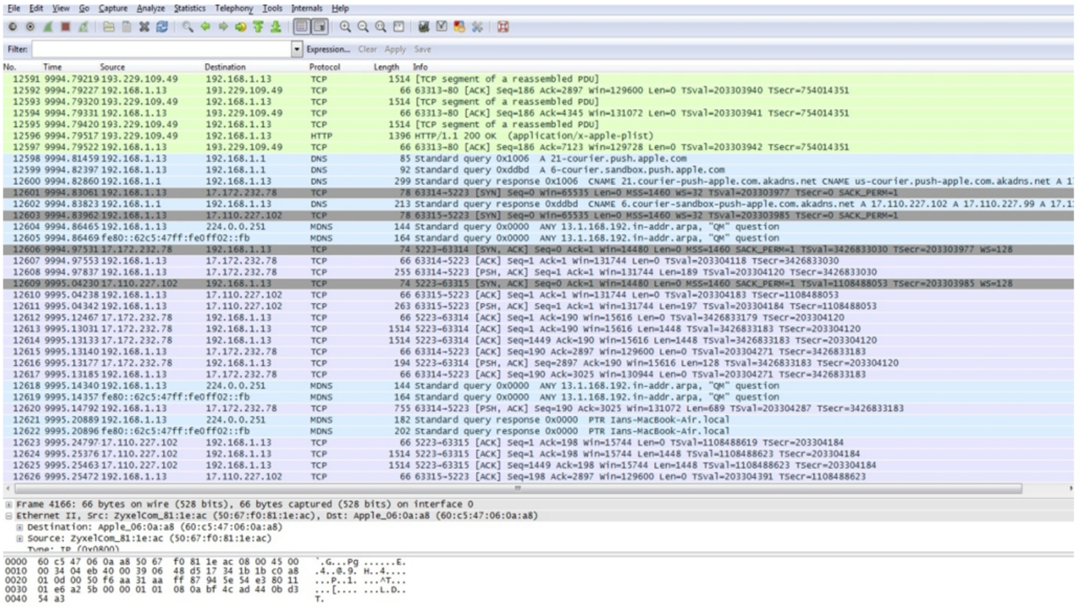
# Network trace files

This chapter presents the overview of network trace files that how they are collected and processed, what kind of information they contain. This chapter also provides statistical analysis over network trace file from different perspectives to understand the type, nature and properties of data set.

## 4.1 Introduction

A network trace capture is like a wire-tapping over a machine's or device's interface data to intercept what is flowing over the wire. Network traces are used for analyzing and troubleshooting network related issues and also for educational and research purposes. In order to understand and get useful meaning of the network trace, one should know how various protocols functions for examples, how they send and receive data.

Wireshark [33] (previously known as Ethereal) is the most commonly used tool to capture the network traces and to intercept all the packets flowing through the machine or network. Figure 4.1 depicts one example of network trace captured from Wireshark. Wireshark captures, opens and analyzes each packet flowing through the network interface and produces full insight of the packet for example, which protocol is used, what are the source and destination IP addresses with packet length, port number and time of packet leaving the interface. The purpose of capturing network traces varies from analyzing network issues, detecting malicious data, monitoring bandwidth utilization and debugging new communication protocols etc.


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time       Source          Destination     Protocol Length Info
12591 9994.79219 193.229.109.49  192.168.1.13    TCP      1514 [TCP segment of a reassembled PDU]
12592 9994.79227 192.168.1.13    193.229.109.49  TCP        66 63313→80 [ACK] Seq=186 Ack=2897 Win=129600 Len=0 TSval=203303940 TSecr=754014351
12593 9994.79320 193.229.109.49  192.168.1.13    TCP      1514 [TCP segment of a reassembled PDU]
12594 9994.79331 192.168.1.13    193.229.109.49  TCP        66 63313→80 [ACK] Seq=186 Ack=4345 Win=131072 Len=0 TSval=203303941 TSecr=754014351
12595 9994.79420 193.229.109.49  192.168.1.13    TCP      1514 [TCP segment of a reassembled PDU]
12596 9994.79517 193.229.109.49  192.168.1.13    HTTP     1396 HTTP/1.1 200 OK  (application/x-apple-plist)
12597 9994.79522 192.168.1.13    193.229.109.49  TCP        66 63313→80 [ACK] Seq=186 Ack=7123 Win=129728 Len=0 TSval=203303942 TSecr=754014351
12598 9994.81459 192.168.1.13    192.168.1.1     DNS        85 Standard query 0x1006  A 21-courier.push.apple.com
12599 9994.82397 192.168.1.13    192.168.1.1     DNS        92 Standard query 0xddbd  A 6-courier.sandbox.push.apple.com
12600 9994.82860 192.168.1.1     192.168.1.13    DNS       299 Standard query response 0x1006  CNAME 21.courier-push-apple.com.akadns.net CNAME us-courier.push-apple.com.akadns.net A 1
12601 9994.83061 192.168.1.13    17.172.232.78   TCP        78 63314→5223 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=203303977 TSecr=0 SACK_PERM=1
12602 9994.83823 192.168.1.1     192.168.1.13    DNS       213 Standard query response 0xddbd  CNAME 6.courier-sandbox-push-apple.com.akadns.net A 17.110.227.102 A 17.110.227.99 A 17.1
12603 9994.83962 192.168.1.13    17.110.227.102  TCP        78 63315→5223 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=203303985 TSecr=0 SACK_PERM=1
12604 9994.86465 192.168.1.13    224.0.0.251     MDNS      144 Standard query 0x0000  ANY 13.1.168.192.in-addr.arpa, "QM" question
12605 9994.86469 fe80::62c5:47ff:fe0ff02::fb     MDNS      164 Standard query 0x0000  ANY 13.1.168.192.in-addr.arpa, "QM" question
12606 9994.97531 17.172.232.78   192.168.1.13    TCP        74 5223→63314 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=3426833030 TSecr=203303977 WS=128
12607 9994.97553 192.168.1.13    17.172.232.78   TCP        66 63314→5223 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=203304118 TSecr=3426833030
12608 9994.97837 192.168.1.13    17.172.232.78   TCP       255 63314→5223 [PSH, ACK] Seq=1 Ack=1 Win=131744 Len=189 TSval=203304120 TSecr=3426833030
12609 9995.04230 17.110.227.102  192.168.1.13    TCP        74 5223→63315 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1108488053 TSecr=203303985 WS=128
12610 9995.04238 192.168.1.13    17.110.227.102  TCP        66 63315→5223 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=203304183 TSecr=1108488053
12611 9995.04342 192.168.1.13    17.110.227.102  TCP       263 63315→5223 [PSH, ACK] Seq=1 Ack=1 Win=131744 Len=197 TSval=203304184 TSecr=1108488053
12612 9995.12467 17.172.232.78   192.168.1.13    TCP        66 5223→63314 [ACK] Seq=1 Ack=190 Win=15616 Len=0 TSval=3426833179 TSecr=203304120
12613 9995.13031 17.172.232.78   192.168.1.13    TCP      1514 5223→63314 [ACK] Seq=1 Ack=190 Win=15616 Len=1448 TSval=3426833183 TSecr=203304120
12614 9995.13133 17.172.232.78   192.168.1.13    TCP      1514 5223→63314 [ACK] Seq=1449 Ack=190 Win=15616 Len=1448 TSval=3426833183 TSecr=203304120
12615 9995.13140 192.168.1.13    17.172.232.78   TCP        66 63314→5223 [ACK] Seq=190 Ack=2897 Win=129600 Len=0 TSval=203304271 TSecr=3426833183
12616 9995.13177 17.172.232.78   192.168.1.13    TCP       194 5223→63314 [PSH, ACK] Seq=2897 Ack=190 Win=15616 Len=128 TSval=3426833183 TSecr=203304120
12617 9995.13185 192.168.1.13    17.172.232.78   TCP        66 63314→5223 [ACK] Seq=190 Ack=3025 Win=130944 Len=0 TSval=203304271 TSecr=3426833183
12618 9995.14340 192.168.1.13    224.0.0.251     MDNS      144 Standard query 0x0000  ANY 13.1.168.192.in-addr.arpa, "QM" question
12619 9995.14357 fe80::62c5:47ff:fe0ff02::fb     MDNS      164 Standard query 0x0000  ANY 13.1.168.192.in-addr.arpa, "QM" question
12620 9995.14792 192.168.1.13    17.172.232.78   TCP       755 63314→5223 [PSH, ACK] Seq=190 Ack=3025 Win=131072 Len=689 TSval=203304287 TSecr=3426833183
12621 9995.20889 192.168.1.13    224.0.0.251     MDNS      182 Standard query response 0x0000  PTR Ians-MacBook-Air.local
12622 9995.20896 fe80::62c5:47ff:fe0ff02::fb     MDNS      202 Standard query response 0x0000  PTR Ians-MacBook-Air.local
12623 9995.24797 17.110.227.102  192.168.1.13    TCP        66 5223→63315 [ACK] Seq=1 Ack=198 Win=15744 Len=0 TSval=1108488619 TSecr=203304184
12624 9995.25376 17.110.227.102  192.168.1.13    TCP      1514 5223→63315 [ACK] Seq=1 Ack=198 Win=15744 Len=1448 TSval=1108488623 TSecr=203304184
12625 9995.25463 17.110.227.102  192.168.1.13    TCP      1514 5223→63315 [ACK] Seq=1449 Ack=198 Win=15744 Len=1448 TSval=1108488623 TSecr=203304184
12626 9995.25472 192.168.1.13    17.110.227.102  TCP        66 63315→5223 [ACK] Seq=198 Ack=2897 Win=129600 Len=0 TSval=203304391 TSecr=1108488623
```

```
⊞ Frame 4166: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
⊟ Ethernet II, Src: ZyxelCom_81:1e:ac (50:67:f0:81:1e:ac), Dst: Apple_06:0a:a8 (60:c5:47:06:0a:a8)
  ⊞ Destination: Apple_06:0a:a8 (60:c5:47:06:0a:a8)
  ⊞ Source: ZyxelCom_81:1e:ac (50:67:f0:81:1e:ac)
    Type: IP (0x0800)
```

```
0000  60 c5 47 06 0a a8 50 67  f0 81 1e ac 08 00 45 00   `.G...Pg ......E.
0010  00 34 04 eb 40 00 39 06  48 d5 17 34 1b 1b c0 a8   .4..@.9. H..4....
0020  01 0d 00 50 f6 aa 31 aa  ff 87 94 5e 54 e3 80 11   ...P..1. ...^T...
0030  01 e6 a2 5b 00 00 01 01  08 0a bf 4c ad 44 0b d3   ...[.... ...L.D..
0040  54 a3                                              T.
```

Figure 4.1: Example of a network trace in wireshark.

## 4.2 Capturing network traces

Network traces can be captured from different interfaces, e.g. from some computer's or router's interface or from mobile network's Gn interface. Figure 4.2 depicts the Public Land Mobile Network (PLMN) Architecture. All GPRS support nodes have a interface called Gn interface. This interface is located between SSGN and other SGNSs and GGSNs. The protocol used by these nodes is called GTP (GPRS Tunneling Protocol) [4] [34].

The choice of interface to capture the trace depends upon the application and purpose of the interception. As shown in Figure 4.1, the network trace contains fields like time, source and destination IP of the packet, protocol with packet length and some additional information in info field. Info field contains different kind of information depending on type of interface used to capture the network trace. If the network trace is captured from a traditional network, then info field may contains port numbers, ARP requests and TCP SYN and ACK messages. In case of capture from Gn interface, it might contain some parameter like IMSI number and IMEI number etc. For malware capture, info field might contain URL of the malicious code.

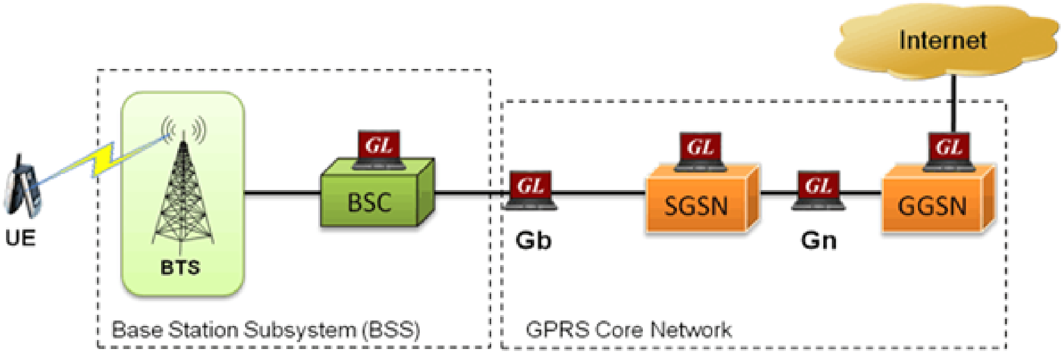
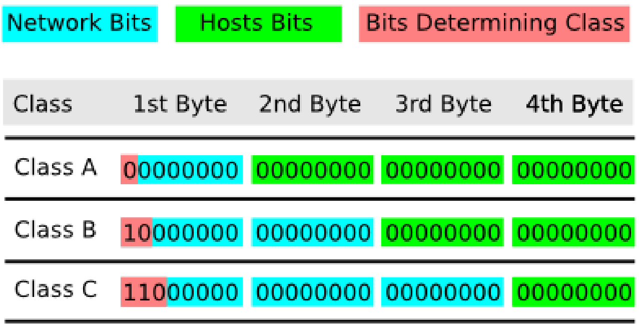


Figure 4.2: Public Land Mobile Network Architecture [4]

## 4.3 Information content

IP addresses are kind of fields which contain much more information than it apparently seems. The numbers in different octets in IP address represent some class, for example, numbers between 0 to 126 in first octet of the IP address represent class A addresses, 128 to 191 represent class B addresses and so on. Figure 4.3 shows the types of information obtained from an IP address. One example of re-identification from IP address class is of North Korea which only uses one block of IPv4 address that is, 175.45.176.0 - 175.45.179.255 [35]. Thus it's easy to know that originator of data is from North Korea if source IP is from above mentioned block of IP addresses.


Network Bits Hosts Bits Bits Determining Class

| Class | 1st Byte | 2nd Byte | 3rd Byte | 4th Byte |
|---|---|---|---|---|
| Class A | 00000000 | 00000000 | 00000000 | 00000000 |
| Class B | 10000000 | 00000000 | 00000000 | 00000000 |
| Class C | 11000000 | 00000000 | 00000000 | 00000000 |

Figure 4.3: Information contained in an IP address [5].

The protocol, length and info fields contain some sensitive information but it's difficult to make some inferences by looking at each field individually. Although single fields might not be enough for re-identification but combination of certain fields like IP address, location data etc. can lead to re-identification.

There are some functional dependencies among the fields in the network trace, for example, in protocol and length fields and also between protocol and port number i.e. HTTP and FTP have port number 80 and 21 respectively. Transport layer protocols like TCP and UDP, carry data payload have most packets of larger length e.g. more than 1500 bytes. The other management protocols for example ICMP, DNS etc. have packet lengths mostly fewer than 200 bytes. So due to this field relation, it might be easy to guess about the protocol type by looking at packet length. Figure 4.4 depicts the protocol and packet length dependency chart.

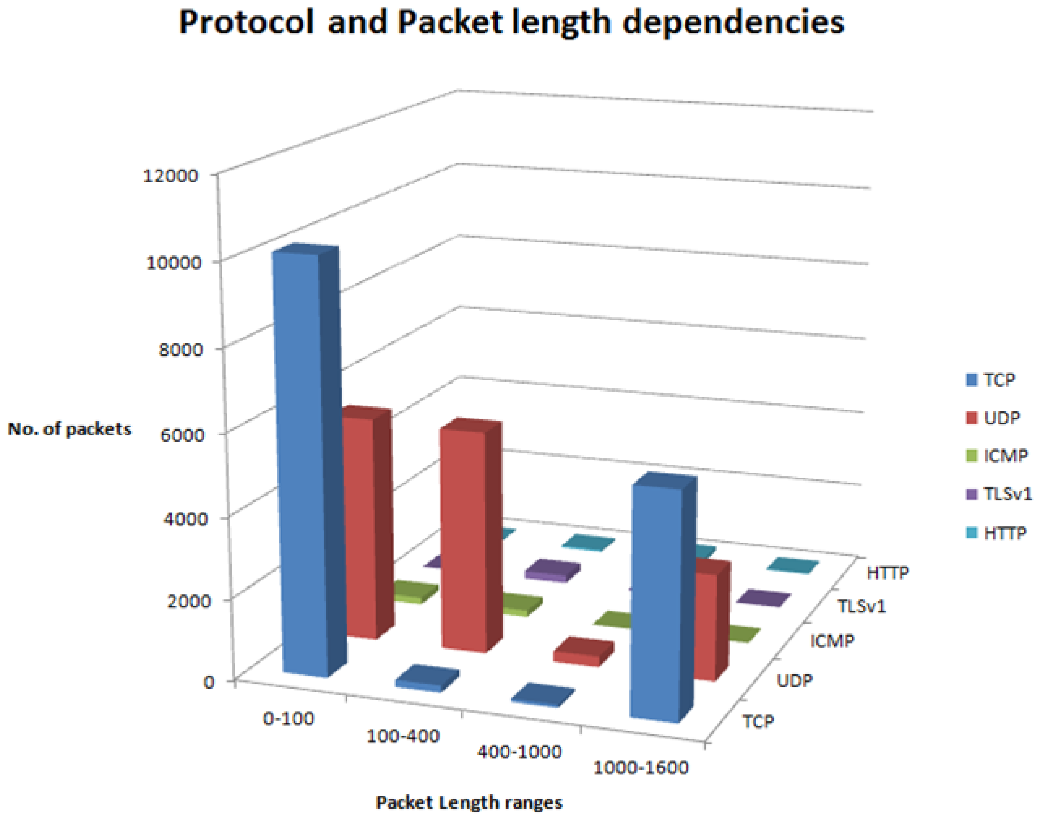


Figure 4.4: Protocol and packet length dependency chart.

## 4.4 Statistical analysis on our network trace

Before doing statistical analysis on network data set, one needs to understand the statistical properties of the data. Statistical tools can be utilized to organize, display and summarize the data. Goal of statistical analysis is to gain understanding from data.

**Mean:**It is the average of values after their total sum is taken.

**Standard deviation:** It is the measure of how data points are deviated from the mean.

**Distribution:** Distribution of a variable defines the values, the variable can take and how often the variable takes these values. Usually distribution is plotted by histograms or by stem plots.

**Spread:** Simplest useful numerical description of data consists of both a measure of center and a measure of spread. One way to describe spread is to give several percentiles. The p:th percentile is the value such that p% of the measurements fall at or below it i.e. Median is the 50th, first quartile (Q1) is the 25th and third quartile (Q3) is the 75th percentile. [36]

**Five number summary:** The five number summary comprises of minimum value, 1st quartile Q1, median, 3rd quartile Q3 and maximum value. It provides good summary of the distribution in a nutshell. The five number summary is generally depicted using box plot in which central box spans the quartiles Q1 and Q3 and line marks the median, lines extend from the box to mark the maximum and minimum. It is especially suitable for comparison of distributions.

Network trace file in this thesis work was converted from .pcap to .csv format for ease of processing. Python modules were used for processing it for example, CSV module which was used for reading csv files and then Numpy and Pandas modules which were used for cleaning, transforming, merging and reshaping the data set, followed by Pylab and Matplotlib modules which were used for visualizing and plotting the results. This is depicted in figure 4.5.

To start statistical analysis on network trace, packet length distribution is plotted by a histogram using Matplotlib as shown in figure 4.6. It can

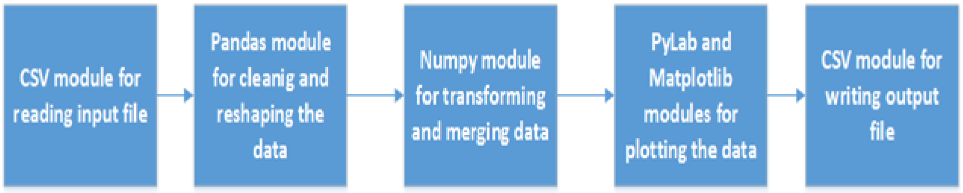

CSV module for reading input file
Pandas module for cleanig and reshaping the data
Numpy module for transforming and merging data
PyLab and Matplotlib modules for plotting the data
CSV module for writing output file

Figure 4.5: Python modules.

be seen that there are peaks at both left and right hand side. Most of the packet length falls into two regions, one below 200 bytes and other greater than 1400 bytes. The peaks at low and high packet length values in the histogram show that either there are huge numbers of small packet lengths or high packet length values. The packet length falling under 200 bytes mostly refers to management protocol like ARP, DNS, ICMP etc and also TCP SYN and ACK messages while packet length greater than 1400 bytes are mostly used for transferring data payload using mostly TCP and UDP protocols. The five number summary of the packet length distribution presented below shows that mean is the center of distribution the Q1 and Q3 quartiles shows the spread of middle half of the data and minimum and maximum values tells about the full spread of the data.

**Five number summary:** (Minimum, Q1, Median, Q3, Maximum) = 44, 62, 107, 1414, 1514

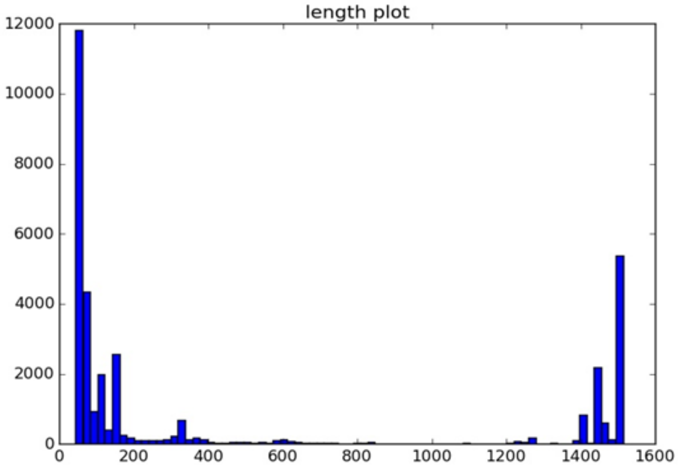


Figure 4.6: Histogram of packet length distribution(bins 100).

Next, histogram of time-stamps of the packets is plotted to check the time periods when traffic was high. This is depicted in figure 4.7. It can be observed that there was not much traffic generated for first 100 seconds but between 200 and 400 seconds traffic volume was high. Thus using time-stamp histogram plots one can analyze about different traffic patterns and detect anomalies in it.

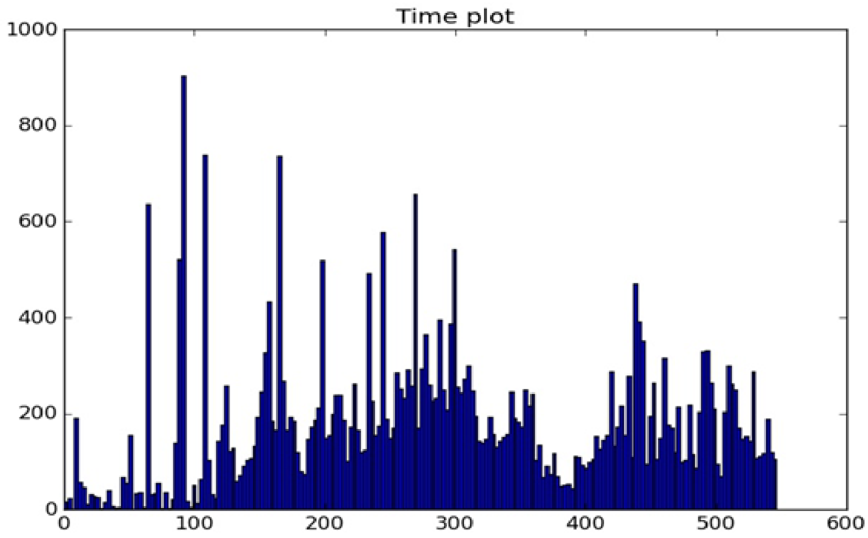


Figure 4.7: Histogram of packet Time-stamp distribution (bins 100).

**Five number summary:** (Minimum, Q1, Median, Q3, Maximum) = 0, 178.96, 282.78, 421.20, 546.15

Figure 4.8 is plotted to further analyze the data set that what are the most commonly used protocols and their number of occurrence, this will help to understand the nature of the traffic. With no surprise, they appear to be frequently used transport layer protocols i.e. TCP and UDP. There are also high number of mobile network protocol i.e. GTP, UCP and SMPP. The other management protocols like DNS, ICMP and also security protocols i.e. TLS, SSL are slightly used. Thus by plotting most frequent protocol chart, one can infer about the type of traffic being generated by the users.

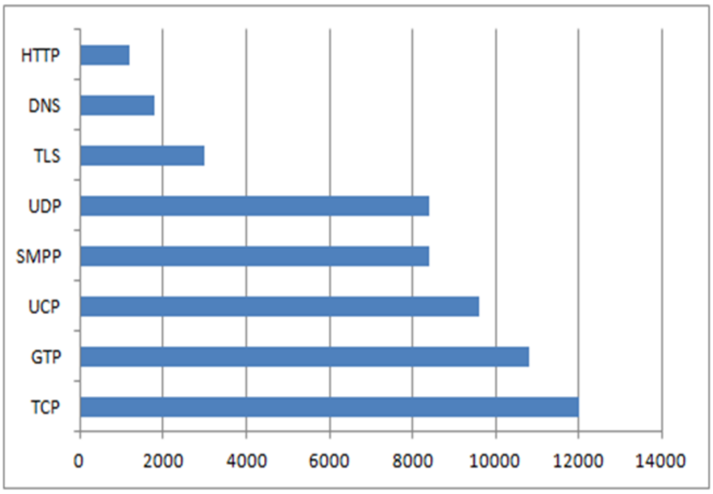


Figure 4.8: Bar chart of most used protocols.

It is also good to know about the value of packet lengths to check the association between the protocol and packet length since some protocol might be using similar packet length every time. Figure 4.9 shows bar chart of most used packet length. It can be seen that packet lengths '54', '1514' and '62' are greatly used. Values such as '54', '62' and '66' seems to be used by DNS, ICMP protocols and also by TCP for sending SYN and ACK message while values like '1514' and '1414' are used by transport layer protocol i.e. TCP and UDP for transferring the data payload.

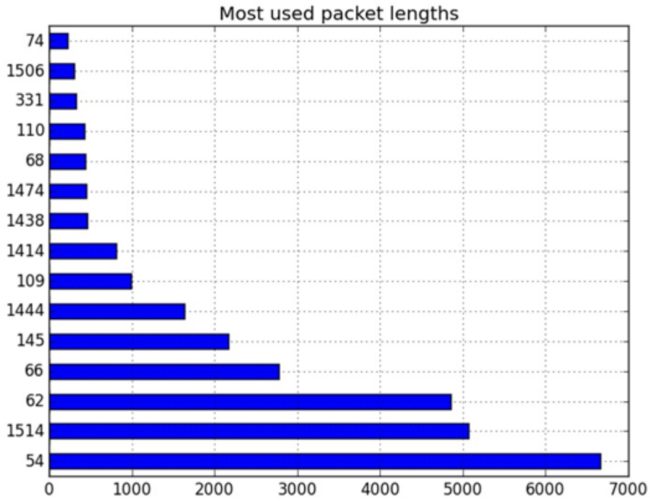


Figure 4.9: Bar chart of most used packet lengths.

IP addresses represent much more information than just being numbers. For various purposes, IP addresses are often classified by their class group i.e. class A, class B etc. IP addresses can be converted into their respective classes to check the class distribution to observe which class addresses are mostly used. Figure 4.10 shows the number of IP addresses of different classes. It is clearly seen that Class A and Class C addresses are most often used.

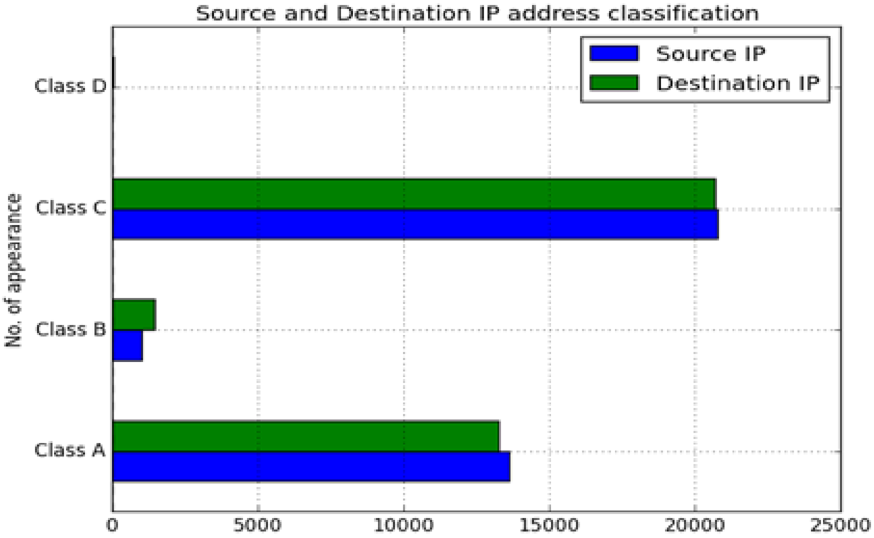


Figure 4.10: Bar chart of IP classes of source and destination.

Lastly, a subplot for different classes of IP addresses can be plotted for packet length over time-stamps to observe their behavior. Figure 4.11 shows the subplot of four IP Class addresses. From the figure, it can be inferred Class A addresses generated high traffic with higher packet length size while class C addresses have high traffic only at certain intervals. Class B addresses generated only small amount of traffic. There is not even a single Class D address used.

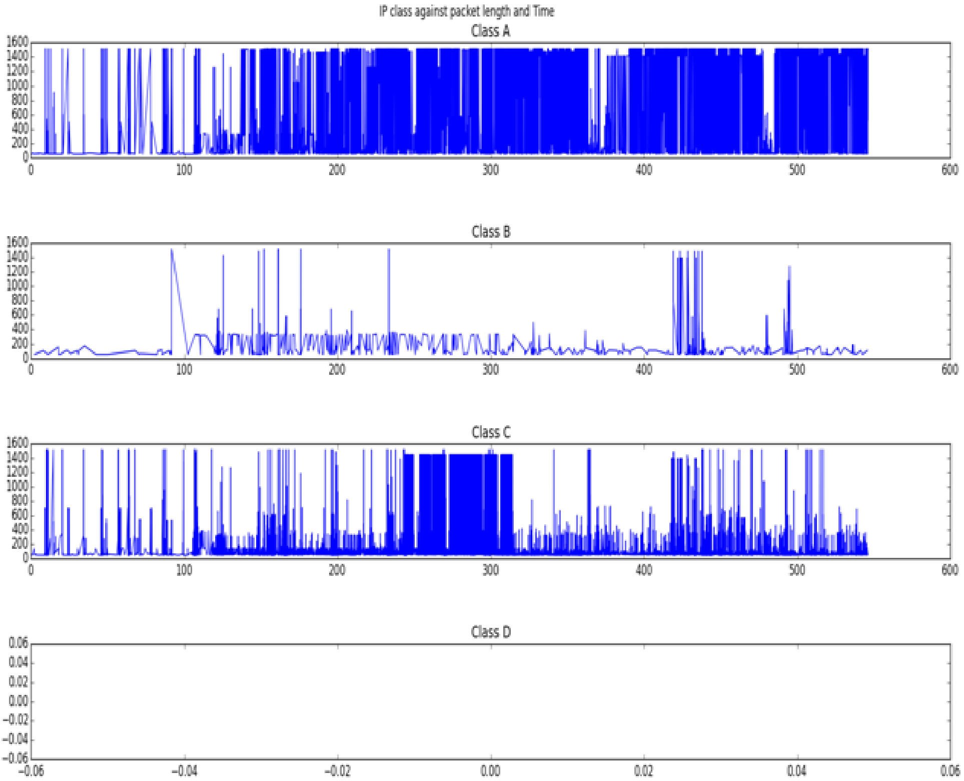


Figure 4.11: Subplot of IP class addresses.

## 4.5 Summary

This chapter mainly describes the properties of a network trace by doing statistical analysis over it. In the beginning, chapter describes network traces and information content in them. Sensitive fields in network traces were identified. The chapter also presents the python modules used in processing the network traces. The statistical analysis section contains detailed analysis over the network trace to understand its properties.

# Chapter 5

# Implementation of differential privacy technique

Noise addition is the technique of obfuscating the data for anonymization purpose while preserving its statistical properties. In network traces, time stamp and packet length fields are considered to be sensitive, hence random noise is added in both fields to anonymize them. In this chapter, three novel techniques of noise addition are discussed with the results. Comparison of the statistical properties of both original and perturbed data is also presented.

Noise addition techniques used in this work are:

1. Random noise addition through differential Privacy
2. Zero mean random noise addition
3. Random noise addition by summing LSBs

These techniques are discussed in detail with the experiment results below.

## 5.1 Noise addition through differential privacy

From chapter 3, it is known that differential privacy uses Laplace noise which is calculated by the following formula.

$$b = \Delta f/\epsilon$$

Where $\Delta$f is the sensitivity of the function and $\epsilon$ is the privacy parameter. In this experiment, sensitivity value is 1 since single numerical value of desired field is processed at one time. Therefore in this case, the query ”what is the value of specific field” yields sensitivity of 1 and then random Laplace noise is added by selecting appropriate value of $\epsilon$.

At first, $\epsilon$ value of 0.01 is selected and after applying Laplace noise to packet length and time stamp field, a histogram of both original and perturbed data is plotted as shown in figure 5.1 and 5.2. It can be clearly seen in these figures that noise is added in such a manner that it has modified the individual records of the data significantly and as a result the perturbed data doesn’t follow the distribution of the original data and it is also shifted towards high values resulting in different shape and pattern of the distribution than the original data. The $\epsilon$=0.01 seems to be inappropriate value for this kind of network trace as noise can destroys useful features of the data fields and makes statistical analysis useless.

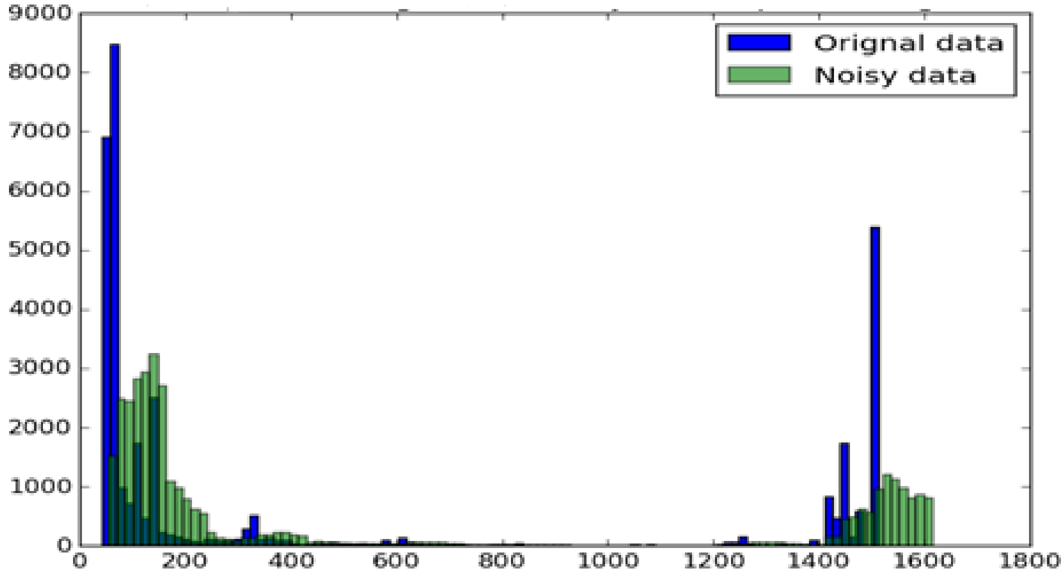


Figure 5.1: Noise addition in Packet length with $\epsilon$=0.01

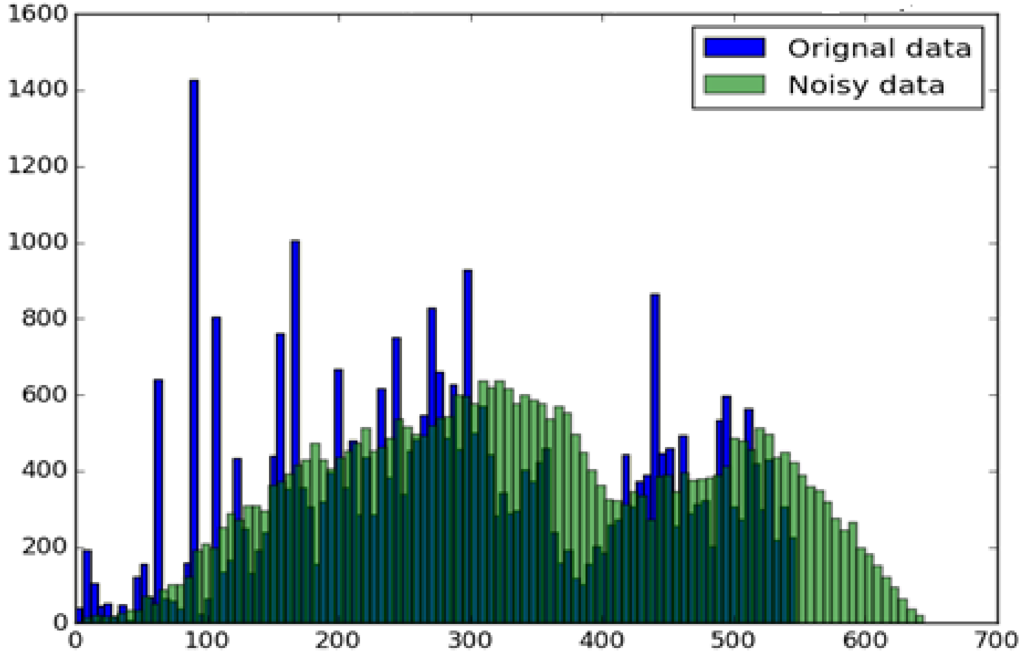


Figure 5.2: Noise addition in time stamp with $\epsilon$=0.01

The selection of appropriate value of privacy parameter $\epsilon$ can be challenging. One has to try different values and check which value of $\epsilon$ fits best for particular kind of data. The value $\epsilon$=0.01 produced high amount of noise earlier. Next value of $\epsilon$ is selected as 0.1 and after applying Laplace noise to each values of packet length and time stamp fields, histograms are plotted to observe the effect of the changing the $\epsilon$ value.

In figure 5.3 and 5.4, original and perturbed data sets are nearly replica of each other and perturbed data set completely follows the distribution of the original data set There is also uniqueness in the patterns of raw and perturbed data as the two data sets increase and decrease together in same direction. Thus this random noise addition with $\epsilon$=0.1, preserve the statistical properties of the original data.

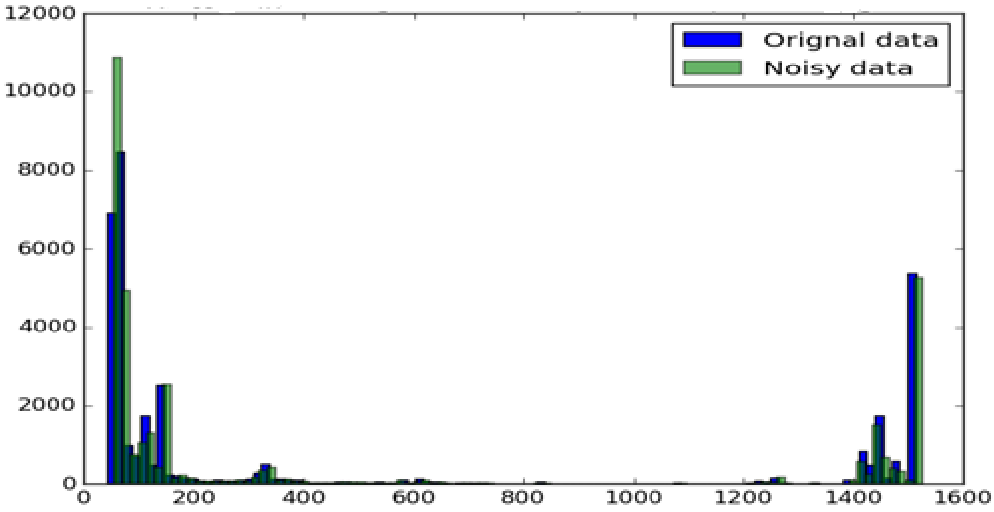


Figure 5.3: Noise addition in packet length with $\epsilon$=0.1

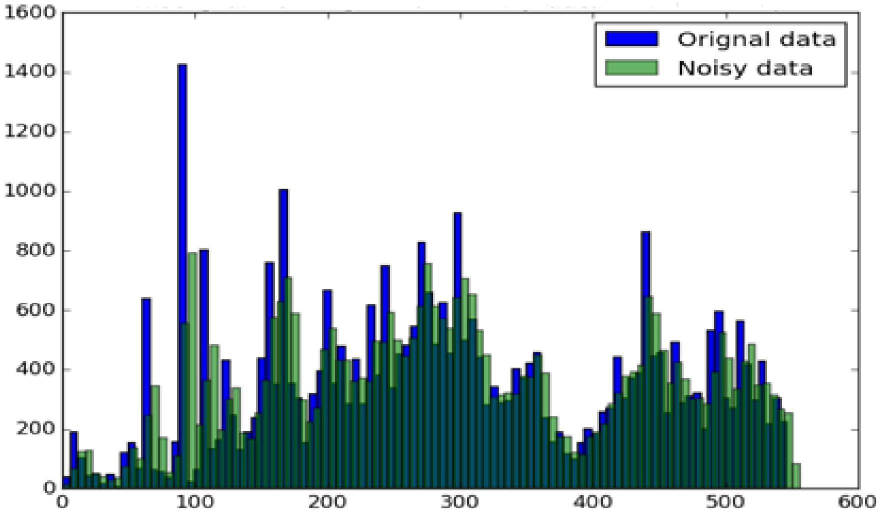


Figure 5.4: Noise addition in time-stamp with $\epsilon$=0.1

### 5.1.1 Comparison between original and perturbed data

Box plots of original and perturbed data set is plotted in figure 5.5 and 5.6 with five number summary. As it can be seen that median or midpoint value (red line) is around 100 bytes which means that most packet lengths are concentrated at low values while the values about the median (upper quartile) are equally distributed and the time stamp values are well dispersed along the whole data set as the median value is in the middle of the data set. The perturbed data pattern set on box plots for both packet length and time stamp fields seems like exact replica of original data which implies that although original data fields are perturbed but their overall statistical properties are preserved.

**Five number summary of original packet length field:**
(Minimum, Q1, Median, Q3, Maximum) = 44, 62, 109, 1414, 1514

**Five number summary of perturbed packet length field:**
(Minimum, Q1, Median, Q3, Maximum) = 54, 66, 113, 1418, 1524

**Five number summary of original time-stamp field:**
(Minimum, Q1, Median, Q3, Maximum) = 0, 178, 282, 421, 546

**Five number summary of perturbed time-stamp field:**
(Minimum, Q1, Median, Q3, Maximum) = 1, 185, 287, 426, 555

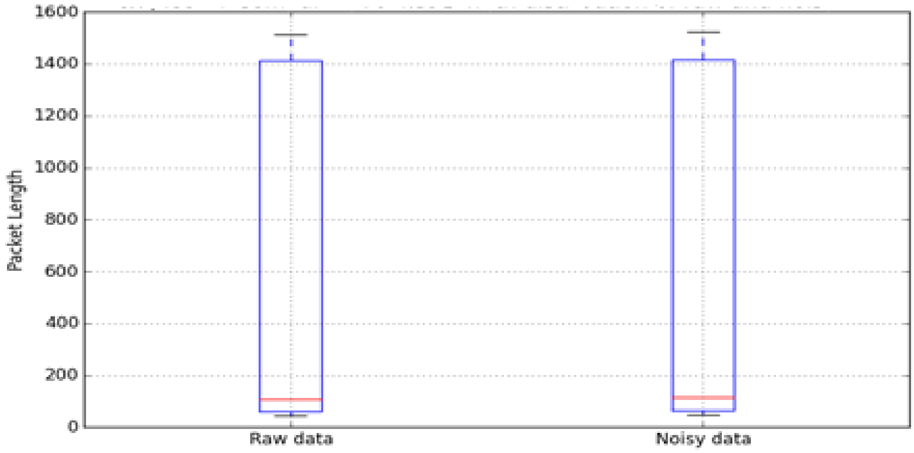


Figure 5.5: Box plots of Packet length for original and perturbed data

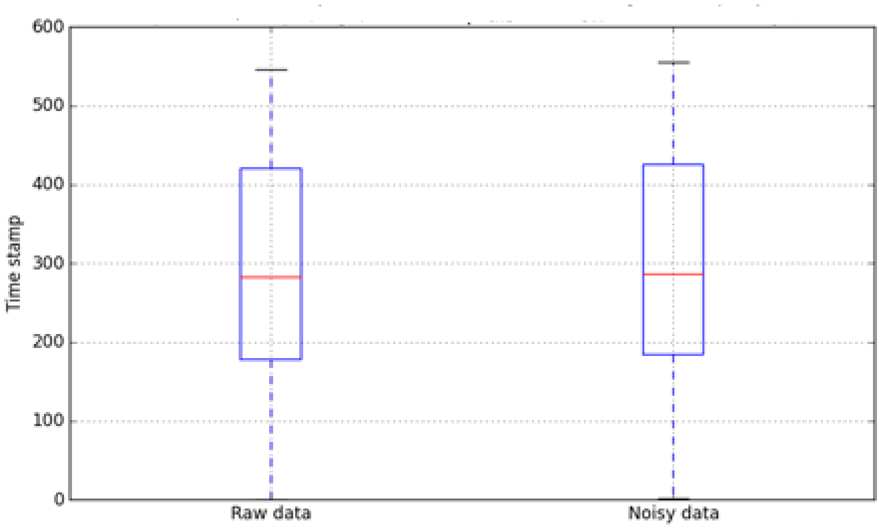


Figure 5.6: Box plots of time-stamp for original and perturbed data

In addition to that some statistical properties of both data sets are shown in table 1. From the statistical parameters, it can be noticed that mean and standard deviation values of original and perturbed data are quite close to each other as the noise addition doesn't greatly ruins these properties therefore it can be deduced that the privacy parameter value $\epsilon$=0.1 is most suitable for this network trace.

| | **Mean** | **Standard deviation** |
|---|---|---|
| **Original data** | 495.54 | 615.13 |
| **Perturbed data** | 500.53 | 615.16 |

Table 5.1: Statistical properties of original and perturbed data

## 5.2 Zero Mean Random Noise addition

The main idea in zero mean noise addition technique is to generate noise vectors whose mean is zero. This technique is used by K. Mivule [26]. In this technique, data is divided into three parts based on the packet length value. Then, three noise vectors are generated with different amount of noise level. First noise vector contains random noise values between 0 and 15, second noise vector contains values between 0 and 30, and third noise vector has values between 0 and 100. The noise is added in such a way that smaller packet length gets smaller amount of noise and so on. First noise vector was added to the packet lengths whose value is less than 150 bytes. Similarly, second noise vector was added to the packet lengths between 150 and 1000 bytes and third noise vector was added to the packet length values over 1000 bytes. This preserves the overall distribution of the packet length shown in histogram in figure 5.7 and for time stamp field shown in figure 5.8. For adding noise in time-stamp, the amount of random noise is reduced to 0 to 10 for first noise vector, 0 to 15 for second and 0 to 30 for third noise vector according to the values in time stamp field.

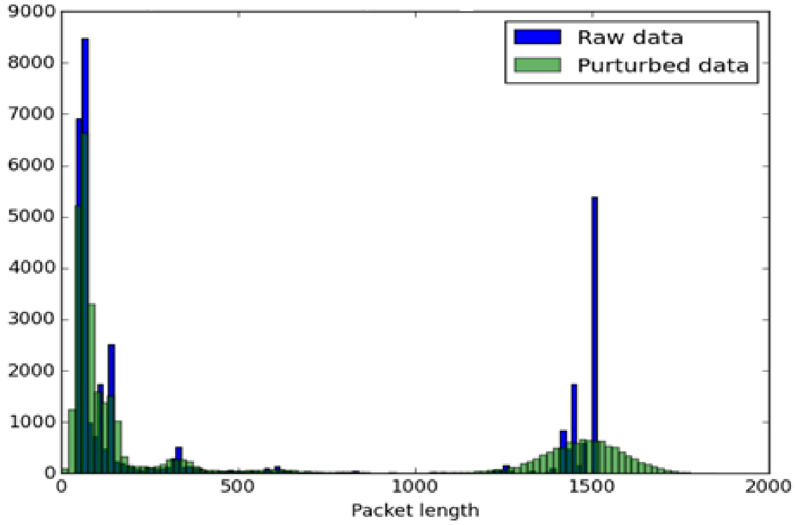


Figure 5.7: Histogram of packet length of original and perturbed data

Histograms in figure 5.7 and 5.8 show that perturbed distribution work well for one half of the data values while for other half the noise distribution has lower peaks as compare to original data. In histogram for time stamp field shown in figure 5.8, the two distributions are not perfect mirror of each other since they contains different peaks at different time intervals.

### 5.2.1 Comparison between raw and perturbed data

Five number summary for packet length and time stamp is given below. It can be seen that corresponding values differ from each other from small margin. This technique also provides almost similar statistical values as the

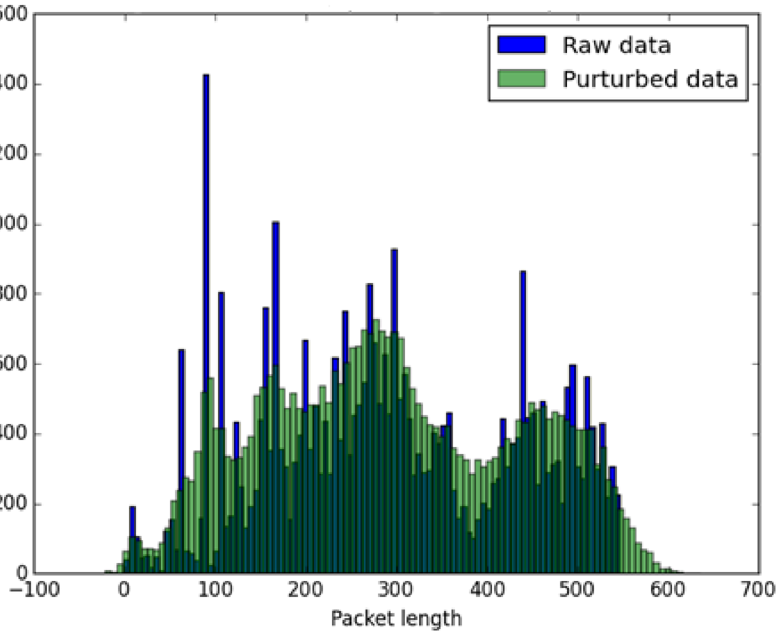


Figure 5.8: Histogram of time stamp of original and perturbed data

original data.

**Five number summary of original packet length field:**
(Minimum, Q1, Median, Q3, Maximum) = 44, 62, 109, 1414, 1514

**Five number summary of perturbed packet length field:**
(Minimum, Q1, Median, Q3, Maximum) = 12, 61, 99, 1376, 1694

**Five number summary of original time-stamp field:**
(Minimum, Q1, Median, Q3, Maximum) = 0, 178, 282, 421, 546

**Five number summary of perturbed time-stamp field:**
(Minimum, Q1, Median, Q3, Maximum) = 0, 180, 282, 421, 577

## 5.3 Noise addition by summing LSBs

Noise addition by summing LSBs is new proposed technique by this thesis work. It is very simple and works very well for numerical fields containing at most 4 digits. This technique does not require generation of random noise vector but instead it utilizes original values of the data to generate the noise values. At first, most significant bit (MSB) of the original values is kept as it is and remaining digits of the original value are summed together and appended with MSB to get the noise values. perturbed value is obtained by adding the original value and noise value. For example, for packet length of 1414, MSB 1 is kept intact and all the LSBs i.e. 414 are summed together which yields 9. Adding this to original value will give perturbed value as 1423. Every number in the field is different so their sum generates the random number and only same numbers can generate similar random numbers. Figure 5.9 depicts the histogram of both original and perturbed data.

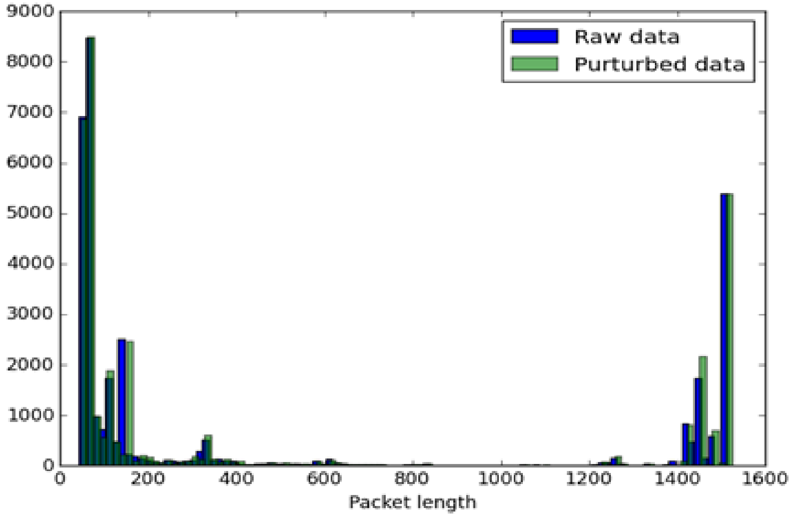


Figure 5.9: Histogram of packet length of original and perturbed data

### 5.3.1 Comparison between raw and perturbed data

From the figure 5.9, it is shown that this technique works well for packet length field as the perturbed distribution has the similar patterns as the original distribution however it is bit shifted towards high values. Table 5.2 shows the statistical properties of the original and perturbed data with five number summary.

| | **Mean** | **Standard deviation** |
|---|---|---|
| **Original data** | 495.54 | 615.13 |
| **Perturbed data** | 502.39 | 617.85 |

Table 5.2: Statistical properties of original and perturbed data

**Five number summary of original packet length field:**
(Minimum, Q1, Median, Q3, Maximum) = 44, 62, 109, 1414, 1514

**Five number summary of perturbed packet length field:**
(Minimum, Q1, Median, Q3, Maximum) = 48, 64,115, 1423, 1524

## 5.4 Discussion

Three novel techniques of random noise addition in continuous data fields are discussed in this chapter. In differential privacy technique, the value of privacy parameter $\epsilon$ is important as it determines the level of privacy to be guaranteed. It can be observed that value of $\epsilon$=0.1 is most suitable. By comparing the statistical properties for this technique, the mean of the standard deviation in both original and perturbed data are very close to each other which means that data points in both data sets are equally deviated from the mean as both data sets rise and fall together along the time axis. Differential privacy technique with $\epsilon$=0.1 proved to be the best noise addition technique as it completely preserves the statistical values like mean and standard deviation of the data set.

Zero mean noise addition technique worked well for the first half of the data set but for other half it produced different patterns in the histogram. Also by looking at five number summary of both raw and perturbed data, it seems that this technique is not as effective as differential privacy technique.

Noise addition by summing LSBs technique is new proposed technique. It is found that this technique is very useful for numerical values of up to size 4 digits and give exaggerate results when this limit is exceeded, therefore this technique has been only implemented on packet length field and not on time stamp field which contains fairly large number of digits.

## 5.5 Summary

In this chapter, some noise addition techniques were implemented for the purpose of anonymization of network trace. Random noise was added to packet length and time stamp fields using these techniques. After perturbing these two fields, results of each technique was compared with original data fields by five number summary and box plot which is used as statistical method to compare the distributions.

# Chapter 6

# Implementation of $\ell$-diversity technique

After obfuscating the data sets using k-anonymity technique, there seem to be the number of identical records present in the anonymized data and it has been found that inference can be made to individuals even when data is anonymized, this is where $\ell$-diversity technique made its basis.

The main idea behind this technique is to create number of blocks of data set. The size of the block depends on the total size of data set. Large data sets usually require the blocks of larger length. The distribution of discrete fields values i.e. protocol field, needs to be consistent in each block. Figure 6.1 shows the percentage of protocol distribution present in the network trace trace. The benefit of $\ell$-diversity technique is to introduce the diversity of protocols fields in each block. If this is not done, the certain portion of data which contains same protocol fields would be vulnerable to inference attack.

## 6.1 Equivalence class creation

In order to obfuscate the protocol value, it can be changed to its equivalence class name. The benefit of doing so is to avoid any inference attack which might occur if original protocol values are exposed.

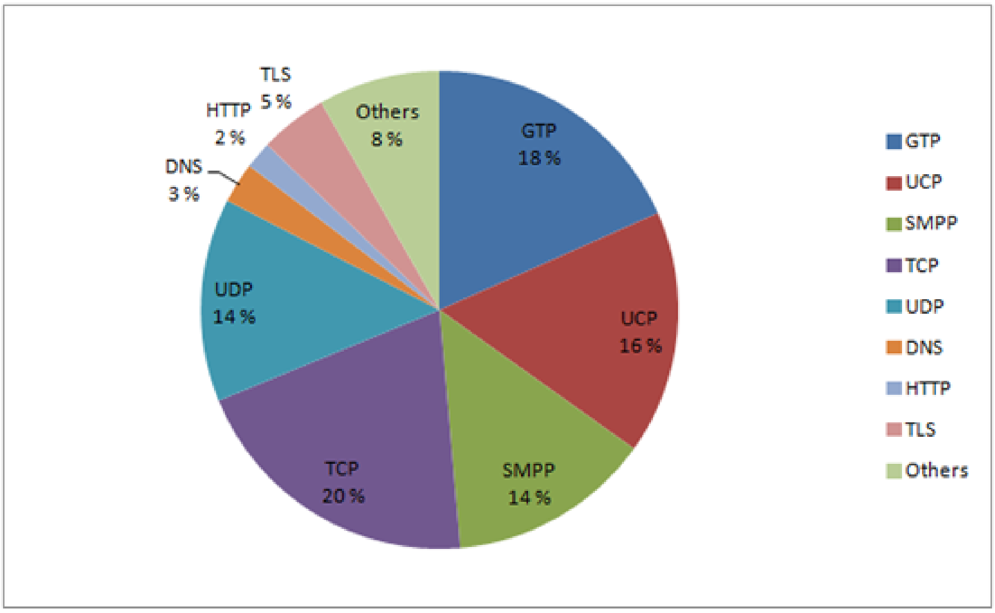


Figure 6.1: Percentage of most used protocols

Protocol field in network trace can be grouped in a family equivalence class, where family name represents the characteristics of its members. To create the protocol equivalence classes, type of protocols present in the data set are examined first and then protocols of similar functionality are grouped and are put in their respective equivalence class for example TCP, UDP and other transport protocols, are placed in equivalence class named as transport protocol. Similarly protocols like ICMP, DNS, ARP etc, are replaced with equivalence class named as management protocols. This process is repeated for all protocols. Table 6.1 depicts the protocol equivalence class family and its members. Although, replacing the protocol field value with its equivalence class name ruins some amount of data but it still gives some amount of information that what types of protocol were anonymized, this is actually the trade-off between utility and privacy.

Number of occurrence of each protocols equivalence class can be easily calculated now. Distribution of each protocols equivalence class is plotted using again pie chart in Figure 6.2.

| **Equivalence class name** | * | * | * | * |
|---|---|---|---|---|
| Transport Protocols | TCP | UDP | * | * |
| Management Protocols | DNS | ARP | DHCP | ICMP |
| Security Protocols | TLS | SSL | TLSv1 | TLSv2 |
| Mobile Networks Protocols | GTP | UCP | SSMP | GTPv2 |
| Other Protocols | * | * | * | * |

Table 6.1: Equivalence class of protocols

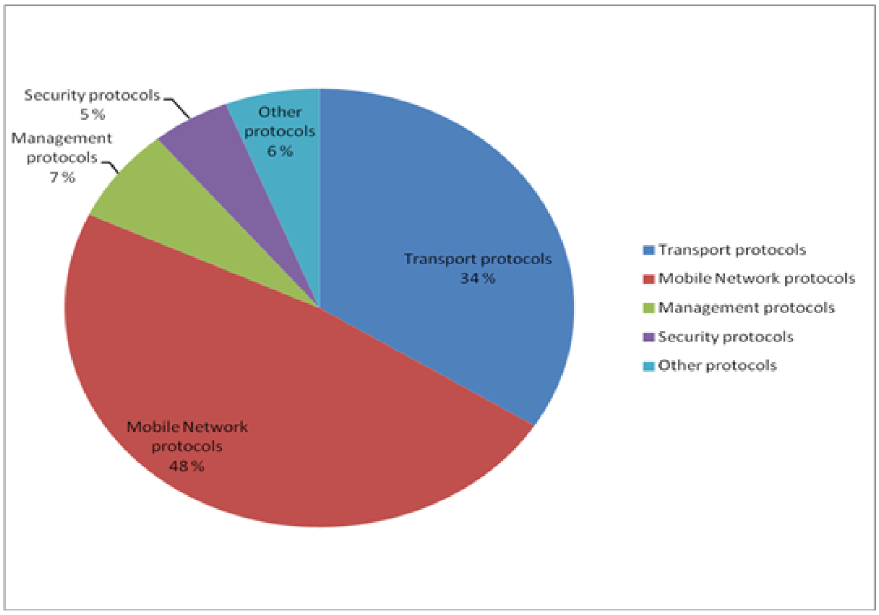


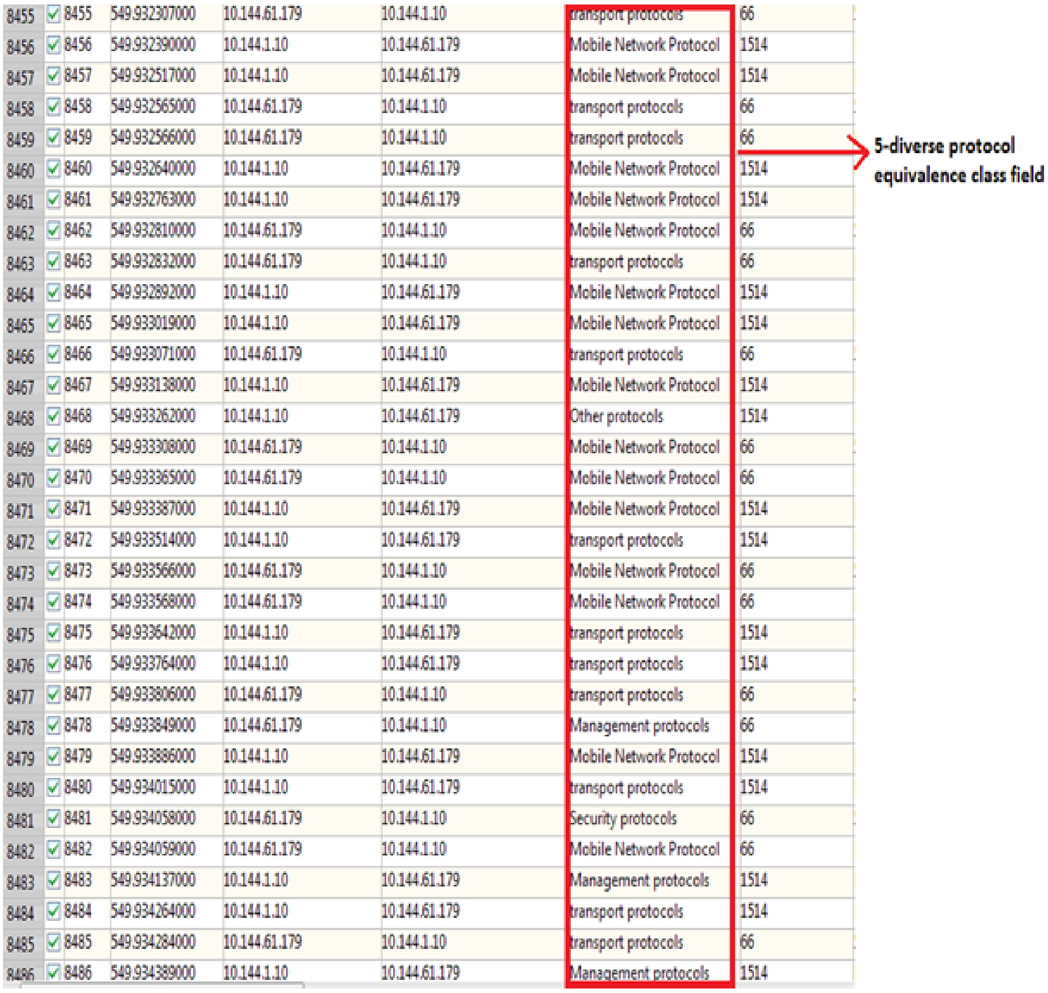
Figure 6.2: Distribution of protocol equivalence class

In network trace, it was determined that how many number of blocks of which length should be created. This is done by selecting the length of each block as 100 and then dividing this length to the length of whole data set to get the total number of blocks. It should also be assured that percentage of each protocol type remains same in each block of data.

## 6.2 Results

The number of equivalence class determines the $\ell$ parameter in $\ell$-diversity technique i.e. three equivalence classes will give 3-diverse anonymized data set and so on. In the network trace, five protocol equivalence classes were created and therefore anonymized network trace is 5-diverse.

The question of how many number of equivalence class should be sufficient, depends upon level of anonymization needed. The relation between $\ell$ parameter and privacy is direct, meaning that the higher the $\ell$ parameter the better the privacy so number of equivalence classes determines the level of ensured protection. Figure 6.3 shows the sample portion of 5-diverse anonymized data where is $\ell$=5 indicating the number of diverse values in each block of the data field.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8455 | ☑ | 8455 | 549.932307000 | 10.144.61.179 | 10.144.1.10 | transport protocols | 66 |
| 8456 | ☑ | 8456 | 549.932390000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8457 | ☑ | 8457 | 549.932517000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8458 | ☑ | 8458 | 549.932565000 | 10.144.61.179 | 10.144.1.10 | transport protocols | 66 |
| 8459 | ☑ | 8459 | 549.932566000 | 10.144.61.179 | 10.144.1.10 | transport protocols | 66 |
| 8460 | ☑ | 8460 | 549.932640000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8461 | ☑ | 8461 | 549.932763000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8462 | ☑ | 8462 | 549.932810000 | 10.144.61.179 | 10.144.1.10 | Mobile Network Protocol | 66 |
| 8463 | ☑ | 8463 | 549.932832000 | 10.144.61.179 | 10.144.1.10 | transport protocols | 66 |
| 8464 | ☑ | 8464 | 549.932892000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8465 | ☑ | 8465 | 549.933019000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8466 | ☑ | 8466 | 549.933071000 | 10.144.61.179 | 10.144.1.10 | transport protocols | 66 |
| 8467 | ☑ | 8467 | 549.933138000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8468 | ☑ | 8468 | 549.933262000 | 10.144.1.10 | 10.144.61.179 | Other protocols | 1514 |
| 8469 | ☑ | 8469 | 549.933308000 | 10.144.61.179 | 10.144.1.10 | Mobile Network Protocol | 66 |
| 8470 | ☑ | 8470 | 549.933365000 | 10.144.61.179 | 10.144.1.10 | Mobile Network Protocol | 66 |
| 8471 | ☑ | 8471 | 549.933387000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8472 | ☑ | 8472 | 549.933514000 | 10.144.1.10 | 10.144.61.179 | transport protocols | 1514 |
| 8473 | ☑ | 8473 | 549.933566000 | 10.144.61.179 | 10.144.1.10 | Mobile Network Protocol | 66 |
| 8474 | ☑ | 8474 | 549.933568000 | 10.144.61.179 | 10.144.1.10 | Mobile Network Protocol | 66 |
| 8475 | ☑ | 8475 | 549.933642000 | 10.144.1.10 | 10.144.61.179 | transport protocols | 1514 |
| 8476 | ☑ | 8476 | 549.933764000 | 10.144.1.10 | 10.144.61.179 | transport protocols | 1514 |
| 8477 | ☑ | 8477 | 549.933806000 | 10.144.61.179 | 10.144.1.10 | transport protocols | 66 |
| 8478 | ☑ | 8478 | 549.933849000 | 10.144.61.179 | 10.144.1.10 | Management protocols | 66 |
| 8479 | ☑ | 8479 | 549.933886000 | 10.144.1.10 | 10.144.61.179 | Mobile Network Protocol | 1514 |
| 8480 | ☑ | 8480 | 549.934015000 | 10.144.1.10 | 10.144.61.179 | transport protocols | 1514 |
| 8481 | ☑ | 8481 | 549.934058000 | 10.144.61.179 | 10.144.1.10 | Security protocols | 66 |
| 8482 | ☑ | 8482 | 549.934059000 | 10.144.61.179 | 10.144.1.10 | Mobile Network Protocol | 66 |
| 8483 | ☑ | 8483 | 549.934137000 | 10.144.1.10 | 10.144.61.179 | Management protocols | 1514 |
| 8484 | ☑ | 8484 | 549.934264000 | 10.144.1.10 | 10.144.61.179 | transport protocols | 1514 |
| 8485 | ☑ | 8485 | 549.934284000 | 10.144.61.179 | 10.144.1.10 | transport protocols | 66 |
| 8486 | ☑ | 8486 | 549.934389000 | 10.144.1.10 | 10.144.61.179 | Management protocols | 1514 |

5-diverse protocol equivalence class field

Figure 6.3: 5-diverse anonymized data sets

## 6.3 Summary

In this chapter, anonymization technique called $\ell$-diversity is presented to obfuscate discrete fields in the network trace file. Implementation of this technique is carried out by creating protocol equivalence class and diversifying its distribution. In the result section, analysis is presented along with the sample of anonymized data set.

# Chapter 7

# Conclusion

This chapter presents the summary of the results and possible implication of these results.

## 7.1 Summary of results

The aim of this master's thesis work is to anonymize network trace file to protect user privacy. This thesis also explains some statistical analysis methods that can be used to compare statistical propertied of two network traces.

During statistical analysis of the network trace, it was found that protocol and packet length have some functional dependencies over each other for example certain protocols which carry only management information like DNS, ARP, DHCP etc have packet length of size less than 200 bytes while some protocol which carry data payload i.e. TCP, UDP etc are heavier in size e.g. 1500 bytes. Therefore, one has to keep these dependencies in mind when anonymizing the network trace.

Differential privacy technique is used to anonymize continuous fields of the network data. The privacy parameter $\epsilon$ was found to be very important in deciding the level of noise to be added to the data fields. Different values of $\epsilon$ were used to observe the effect of noise addition to the data fields and it was observed that as value of $\epsilon$ approached to zero, the data field is modified significantly since noise magnitude approached to infinity.

In order to anonymize protocol field, $\ell$-diversity technique proves to be very useful. Each block of the anonymized data contains well represented or diversified values of the protocol fields. This is done changing the representation of the values by creating equivalence class and replacing protocol values by their corresponding equivalence class. This although ruins fair amount of information but on the other hand provides better privacy.

The level anonymization needed in data set can be determined from the type of analysis carried out on the data set. If the data set is to be published publically for research use, the sensitive fields can be heavily anonymized so that nobody can de-anonymize it. For some useful analysis for example detecting anomalies, the level of anonymization is set such that it shouldn't ruin the data significantly while still preserving the user privacy.

There exist plenty of ways to anonymize IP addresses, traditionally it is done by hashing or changing them to some real number but these methods don't provide any means to carry out statistical analysis over anonymized IP addresses. Two different ways were tried in this thesis work to anonymize IP addresses. First method suppresses the last octet of the IP address while keeping other octets as intact. This technique ensures that user who generated the data packet cannot be traced back while on the hand provides information about the network topology which might be useful in certain analysis. In the second technique, the IP addresses were replaced with their corresponding class type for example 192.168.1.10 IP is replaced by Class C. Although this technique ruins the information about the network ID and subnet mask but still provides some information about which class of IP address was used.

# Chapter 8

# Future work

This section gives some of the topics that were not covered in this work and can be extended as future work. Figure 8.1 depicts road map of our work and also some future work in Blue Square.

**Framework for calculating value of privacy parameter $\epsilon$:** A framework can be created that calculate the best value of privacy parameter $\epsilon$ for any given network trace. The framework can be capable of getting data set as input, process it, applying $\epsilon$-differential privacy technique and producing the anonymized trace.

**Re-identification testing:** The anonymized data set can be processed again to check if it can be de-anonymized to certain level of confidence. This can tell about the efficiency of the anonymization algorithms.

**Anomaly detection:** This work can be extended in a way that it becomes possible to detect anomalies i.e. Malware detection. This can be accomplished by achieving the right level of anonymization i.e. suitable parameters of privacy algorithms.

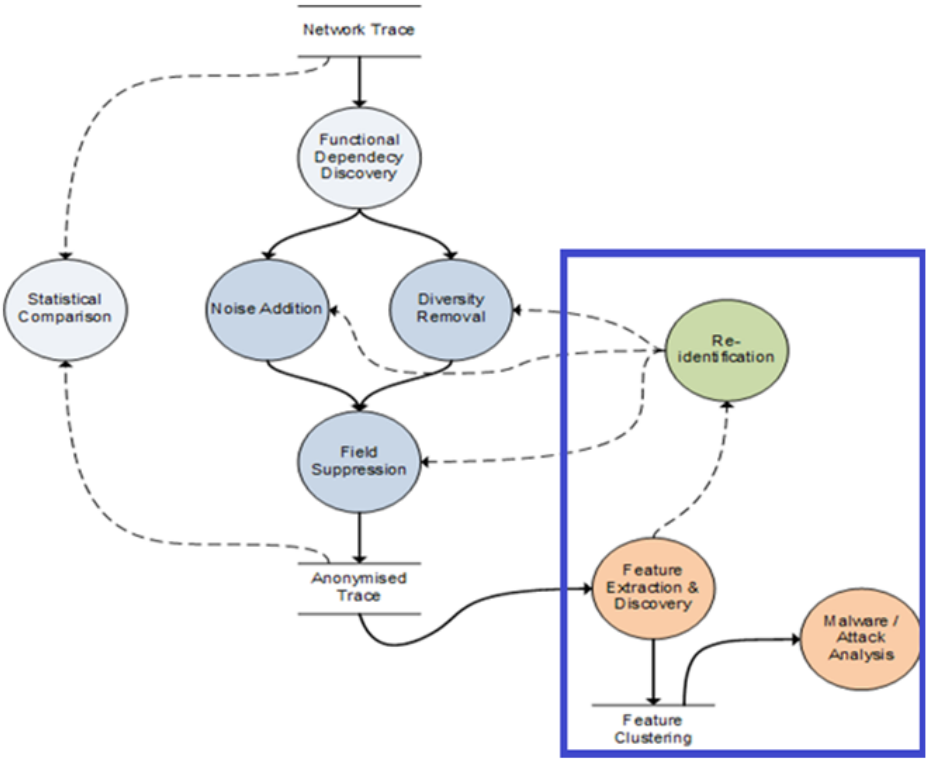


Figure 8.1: An overview of this thesis work

# Bibliography

[1] Anthony Tockar. Differential Privacy: The Basics. http://content.research.neustar.biz/blog/differential-privacy/DensityWidget.html, 2014. [Online; accessed 19-May-2015].

[2] Laplace Distriution. http://beta.boost.org/doc/libs/1_42_0/libs/math/doc/sf_and_dist/html/math_toolkit/dist/dist_ref/dists/laplace_dist.html, 2014. [Online; accessed 19-May-2015].

[3] Senthil R Madhan Subramaniam. An analysis on preservation of privacy in data mining. 2010.

[4] GPRS Architecture Interfaces and Protocols Training Document GPRS System Course at Nokia. http://www.roggeweck.net/uploads/media/Student_-_GPRS_Architecture.pdf, 2013. [Online; accessed 19-May-2015].

[5] Cisco Is Easy. http://ciscoiseasy.blogspot.fi/2010/11/lesson-28-ipv4-address-dissected-part-2.html, 2010. [Online; accessed 19-May-2015].

[6] Balachander Krishnamurthy and Craig E Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 7–12. ACM, 2009.

[7] Helen Nissenbaum. *Privacy in context: Technology, policy, and the integrity of social life.* Stanford University Press, 2009.

[8] Michelle Finneran Dennedy, Jonathan Fox, and Thomas Finneran. *The Privacy Engineer's Manifesto: Getting from Policy to Code to QA to Value.* Apress, 2014.

[9] Simone Fischer-Hübner. *IT-security and privacy: design and use of privacy-enhancing security mechanisms.* Springer-Verlag, 2001.

[10] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

[11] Barbara J Boe, Julia M Hamrick, and Marjorie L Aarant. System and method for profiling customers for targeted marketing, May 22 2001. US Patent 6,236,975.

[12] Leif Nixon. The stakkato intrusions: What happened and what have we learned?. In *CCGRID*, page 27, 2006.

[13] Yingbo Song. *A Behavior-based Approach Towards Statistics-Preserving Network Trace Anonymization.* PhD thesis, Columbia University, 2012.

[14] Nicholas Hopper Mikhail Atallah. *Privacy Enhancing Technologies.* Springer-Verlag Berlin Heidelberg, 2010.

[15] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[16] Michael Barbaro, Tom Zeller, and Saul Hansell. A face is exposed for aol searcher no. 4417749. *New York Times*, 9(2008):8For, 2006.

[17] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[18] Eytan Adar. User 4xxxxx9: Anonymizing query logs. In *Proc of Query Log Analysis Workshop, International Conference on World Wide Web*, 2007.

[19] S Vijayarani and A Tamilarasi. Bit transformation perturbative masking technique for protecting sensitive information in privacy preserving data mining. *International Journal of Database Management Systems (IJDMS)*, 2(4), 2010.

[20] European data protection law, ”Handbook on European data protection law. `http://fra.europa.eu/en/publication/2014/handbook-european-data-protection-law.`, 2008. [Online; accessed 19-May-2015].

[21] Homeland security, ”Handbook for Safeguarding Sensitive Personally Identifiable Information. `https://www.dhs.gov/sites/default/files/publications/privacy/Guidance/`

`handbookforsafeguardingsensitivePII_march_2012_webversion.pdf.`, 2008. [Online; accessed 19-May-2015].

[22] General publication Data Protection Regulation (GDPR). `http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+TA+P7-TA-2014-0212+0+DOC+XML+V0//EN.`, 2008. [Online; accessed 19-May-2015].

[23] Dr Ian Oliver. Privacy engineering: A dataflow and ontological approach. 2014.

[24] IP addresses and the Data protection act). `http://www.out-law.com/page-8060`, 2008. [Online; accessed 19-May-2015].

[25] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. Ip geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review*, 41(2):53–56, 2011.

[26] Kato Mivule. Utilizing noise addition for data privacy, an overview. *arXiv preprint arXiv:1309.3958*, 2013.

[27] Matthias Templ, Bernhard Meindl, and Alexander Kowarik. Introduction to statistical disclosure control (sdc). *Project: Relative to the testing of SDC algorithms and provision of practical SDC, data analysis OG*, 2013.

[28] Cynthia Dwork. Differential privacy: A survey of results. In *Theory and applications of models of computation*, pages 1–19. Springer, 2008.

[29] Rathindra Sarathy and Krishnamurty Muralidhar. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Transactions on Data Privacy*, 4(1):1–17, 2011.

[30] Charu C Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, pages 901–909. VLDB Endowment, 2005.

[31] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[32] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007.*

*ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[33] Angela Orebaugh, Gilbert Ramirez, and Jay Beale. *Wireshark & Ethereal network protocol analyzer toolkit.* Syngress, 2006.

[34] MAPS GPRS Gb Interface Emulator. `http://www.gl.com/gprs-gb-over-ip-emulator-maps.html`, 2012. [Online; accessed 19-May-2015].

[35] APNIC database. `http://www.apnic.net/apnic-bin/whois.pl`, 2014. [Online; accessed 19-May-2015].

[36] Thomas Glover and Kevin Mitchell. *An introduction to biostatistics.* Waveland Press, 2008.