# Enabling Anonymous Crime Reporting on Mobile Phones in the Developing World




Mark John Burke

Department of Computer Science

University of Cape Town

A thesis submitted for the degree of

*Master in Science*

I would like to dedicate this thesis to my loving parents ...

# Acknowledgements

I would like to thank the Mandela Rhodes Foundation for their generous financial assistance as well as the experiences they have allowed me to share in. The financial assistance of the National Research Foundation (NRF) towards this research is also hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

I would not have been able to complete this thesis without the kind and patient help of my supervisor, Dr. Anne Kayem. It has been a great benefit to have a comforting and reassuring voice, as well as guidance in times when I was uncertain.

I am grateful to all those that have supported me throughout this journey, from a warm meal at the right time to a bed or a couch towards the end. You know who you are. Especially my two elder brothers for their love and support - Nicholas and Christopher.

Lastly, thank you to the Lord for the life and opportunities I have been given. *And we know that in all things God works for the good of those who love Him, who have been called according to His purpose.*

# Plagiarism Declaration

I know the meaning of Plagiarism and declare that all of the work in the document, save for that which is properly acknowledged, is my own.

# Abstract

Various campaigns indicate that a need exists for a convenient and anonymous crime reporting framework for the context of the developing world. In this thesis a contribution is made by means of a framework that facilitates reporting crime anonymously based on a secure platform. The framework comprises of two main components namely, a reporting module that is facilitated by unstructured supplementary service data (USSD) on a mobile phone and an anonymization module that is supported by a k-anonymity algorithm. The advantage of using USSD is that it is available to all mobile phones (including the more basic/nonsmart phones that are used by a large percentage of the poorer population in developing countries); and reports made via USSD cannot be traced to the participant. Anonymization has the advantage of guaranteeing user privacy in the management of the reported data. In order to decide on an appropriate anonymization technique for the crime reporting system, we implemented and compared existing popular k-anonymity based algorithms as well as suggesting a crime data anonyimzation algorithm tailored for specific sets of data. The proposed crimemod algorithm is found to provide satisfactory performance and security results. Our results indicate that anonymization algorithms that use hierarchy based generalization techniques, are the best suited to crime reporting scenarios.

# Contents

# List of Figures

# Chapter 1

# Introduction



This thesis presents a crime reporting framework that is based on a mobile platform. In particular we are interested in enabling anonymous reporting in an effective and secure manner in order to protect the privacy of the reports. By "effectively" we imply that the system must allow a reporter to make the report successfully and with a procedure that is simple to use but that guarantees data integrity. By "securely" we imply that the system must guarantee confidentiality and anonymity, without impacting negatively on performance.

The collection, storage and publishing of information for purposes such as business, health, census, government and/or social work typically evokes the notions of privacy and security. This information needs to be protected from unauthorised access but at the same time the information must be easy to access if a user meets the identification criteria of the authentication mechanism. Achieving the right balance between usability and security or privacy can be a challenging problem. Security experts have argued that there is an unavoidable trade-off between preserving security and ensuring usability, therefore system designers must strive to strike a balance between both requirements [3]. The fact that usability and privacy is difficult to quantify and measure, further complicates the building of a system that is both usable and privacy preserving. Where tools exist to measure these two notions, they are often specifically tailored to make assumptions about the nature of a dataset as well as security requirements of protecting the data. For instance, most solutions assume that a certain number of participants are honest

and/or that all data collected is uniformly distributed, of the same length, size and type. Obviously in the anonymous mobile crime reporting scenario data can be submitted using different platforms and could possibly cover only a small segment of the total population distribution. Models built on theoretical assumptions of participant honesty and uniform data distribution could result in misinterpretation of reports. Users and participants of on-line data collection and mining mechanisms are aware of these complications and consequently have legitimate security concerns when it comes to submitting personal and/or sensitive information to computing systems [9].

## 1.1 Motivation

The hypothesis justifying this study is that a need exists for a system where participants can report crimes anonymously using a simple mobile environment.

There is a dual challenge for anonymous crime reporting in the developing world: The first is the fact that participants share the same human apprehensions regarding online data collection found in the developed world - a fear of identity disclosure that can lead to negative consequences. The second is that even if participants can be convinced to contribute to online data collection, for their own good and that of their immediate community, the challenge of platform availability still limits prospects for data collection.

The first challenge relates to privacy concerns and is addressed through the assurance of anonymity in data. The second challenge needs to be addressed in an innovative way that makes a crime reporting platform available to as many citizens as possible. An example of platform availability is perhaps best understood by taking context into consideration: Close to 6 million South Africans have internet access on their phones, but virtually all South Africans have access to a mobile phone [1]. An approach where internet access is required to make reports is thus a limited solution, whereas a service that makes use of all mobile phones seems more suitable. Research on developing world economics has indicated that a big

[1] http://www.worldwideworx.com/?p=294

proportion of the population use mobile devices, and mobile phones in particular, for day-to-day transactions. This is in contrast to other electronic devices such as desktop/laptop computers [19] and is especially true in areas where technological infrastructure is very basic or almost non-existent [12]. For this reason, we expect that users will prefer wireless devices to participate in online crime reporting activities. When using such devices it is important to ensure that the same amount (or more) of privacy is guaranteed as would typically be the case with a stationary desktop computer through a fixed/wired connection. It is however also important to ensure understandable and manageable user interfaces on devices that facilitate crime reports. This is of particular importance on mobile devices, where the screen resolution and size is typically smaller than that of a conventional computer hosting a web browser.

### 1.1.1 Potential outcomes & Impact

The success of this project could lead to a number of advantageous developments:

1. A new crime reporting system will be made available, whereby users can safely and conveniently report crimes they might not otherwise have reported.

2. Research work on the amount of anonymization necessary to ensure safety and privacy. There are various applications for an algorithm which ensures anonymity, while it preserves valuable information.

3. If this project is successful and implementable on a small scale, it could be expanded to a larger audience, possibly leading to less crime and corruption over the long term.

## 1.2 Problem Statement

There exists a reasonable fear of identity disclosure when reporting sensitive information such as a witnessed or suffered crime. In this thesis we aim to build a framework to enable anonymous crime reporting using mobile devices. The collected data will then be recorded in a dataset and each field will be sanitized to

ensure the anonymity and privacy of each participant. We assume that the devices for reporting and networks are safe and secure in the sense that anonymization is only necessary before publication of results (rather than during collection)[1].

The thesis will therefore focus on the following research questions:

1. Can a mobile service be made available to collect data and if so what is the best approach to implementing a mobile crime reporting service?

2. Can anonymity solve the privacy protection problem for participants and how can effective anonymity be ensured?

3. What is the best way to handle anonymizing crime report data for use in applications such as crime statistics databases and crime area avoidance?

## 1.3 Objectives and Contribution

We aimed to develop a platform for reporting transgressions on a mobile device, which would include:

1. A method of sending information from a mobile device

2. A program for capturing such information

3. A database for recording the information captured

In this thesis we present a framework that has achieved this goal. The platform is composed of two main modules and two complimentary modules. The main modules are a data collection module and a data anonymization module. The data collection module handles collecting the reported data from the mobile devices and structuring the data to facilitate information retrieval. The anonymizaton module on the other hand, ensures that any data that is released to the public is suitably anonymized - sanitization of tuples ensures the anonymity and privacy of each participant before the publication of the data.

The distinct modules for data collection and anonymization are presented as a

[1]So, essentially the network/service provider is considered to be trustworthy.

dual contribution in a framework specifically tailored for crime data. Additional modules in the framework exists to facilitate these modules (These modules being the database and authentication mechanism described in Sections 3.3.2 & 3.3.3 respectively). To our knowledge no such a framework exist for the anonymous reporting of crime data.

### 1.3.1 Data Collection

A suitable service for collecting data is Unstructured Supplementary Service Data or USSD. This is a very simple text exchange service that allows for menus to be displayed on a mobile phone and enabling users to interact with these menus. USSD enables a distinctly interactive user experience: As soon as a user provides a reponse to a particular prompt, they are directed to a related prompt based on the answer to the previous prompt. In contrast with other services like short message service (SMS), these menus can enable the developer to limit/control user responses so they are useable and manageable and not unintelligible.

Apart from being able to direct responses, another big consideration is reachability. It would be intuitive to develop a crime reporting application that is downloadable to a smart phone in the developed world, but this will most likely have a low utilization rate in a developing country. As an example consider South Africa where close to 6 million (out of roughly 50 million)[1] South African's have internet access on their phones, but virtually all standard phones have access to USSD [2]. Considering the fact that all mobile phones have access to USSD and the use of simple mobile phones is rapidly increasing in the developing world, USSD has excellent mobile reach. Furthermore it requires no special applications to be downloaded and no internet access or set-up is needed on a user's phone.

Moreover, and perhaps most importantly, a particular benefit for this framework is that no session/message is logged on a phone. While an installed application, SMS text or dialled number is visible on a phone, dialling a USSD number is not logged on a phone's history. This essentially means that no individual can pick

[1] http://finweek.com/2013/01/22/mobile-phone-usage-in-sa/
[2] http://www.worldwideworx.com/?p=294

up a phone after a crime report and see a trace of such a report. All traces are removed as soon as they are sent. This is an important security consideration, given that an integral part of protecting participants is to ensure their anonymity by removing evidence of reports made. To our knowledge, no USSD service for crime reporting exists.

### 1.3.2 Data Anonymization

After data collection from a mobile platform, data is stored in a MySQL database table which is a flexible, large and fast platform. As a second module the fields in the table should be scrutinized for generalization candidature in order to ensure there is no disclosure of sensitive information. To this extent, research was conducted into what the most appropriate privacy requirements are for crime data, evaluating k-anonymity, l-diversity and their respective extensions. Additionally, research into attribute generalization that does not result in the loss of valuable and useful information was conducted. Minimal generalization (and as a result maximized utility) is guaranteed by means of information loss and classification accuracy comparisons of different anonymization techniques. This thesis thus addresses the security question of finding a good anonymization algorithm to enable crime reporting in a secure and efficient way. The other modules of our framework will exist in order to facilitate the achievement of this goal.



## 1.4 Outline

The rest of this thesis is structured as follows: Chapter 2 discusses work related to defining privacy, the problem of anonymous crime reporting to ensure privacy, as well as the current use of mobile devices in data gathering. The crime reporting framework and its various components are outlined in Chapter 3. A proposed crime data anonymization algorithm is presented in Chapter 4. In Chapter 5 we discuss results of a prototype implementation of our framework, and show that k-anonymity algorithms that are based on hierarchy generalization techniques are the best suited to the crime reporting scenario. Finally, we provide concluding remarks in Chapter 6.

# Chapter 2

# Related Work

This Chapter comprises of three main topics: Section 2.1 deals with the concept of privacy and issues related to user privacy concerns from the perspective of data gathering. Section 2.2 outlines approaches that have been suggested as possible approaches to anonymous data collection. Finally, privacy preservation in data publishing is addressed in Section 2.3, 2.4 and 2.5 - these sections describe techniques to preserve privacy as well as particular privacy requirements used with these techniques.



## 2.1 Information Privacy

> *"Individual information has become a valuable commodity that is now being collected, catalogued, and traded in ways never before envisaged"*. - [40]

A simple Google search is capable of finding personal information such as an individuals education, social background, address, contact details and employment condition. It is argued that the release of personal or private information, often without consent, is forced on internet users by institutions of all natures (commercial-, social-, and public). The release of this information contributes to data availability that can have several grave consequences, including the ability of governments, both good and bad, having the ability to pry open the personal lives of their citizens. For example in China the trend has always been for government

to focus more on public security than the privacy of citizens. It is unfortunate that although privacy should be, it is not recognized as a universally established individual right. Amidst all of this, people still see the release of data as a necessary part of modern life. However, there is a feeling amongst the public that they have lost control of their data [11, 17, 28].

Consequently, keeping information private has become the most important concern for several online privacy scholars. It is clear that there is a definite change of individual rights and concerns when it comes to privacy in the 21st century.

Privacy, though often used, is just as often misunderstood. It leads to confusion and further complicates solving the privacy preservation problem. This section aims to provide clarity on the meaning of privacy in the context of data collection, as well as explaining related concepts, especially the role of users and their privacy concerns. An overview of the status quo in privacy preservation is also presented.

### 2.1.1 Basic concepts related to privacy

Privacy has traditionally been defined as giving an individual the right to choose how much and when they want to release any information that relates to their person [40].

It is a lack of control that makes users concerned about data sharing. A User can help secure the online environment if he/she is in charge and has the ability of avoiding the danger associated with losing control over his/her personal information. Privacy preservation is the process of exercising control over how much, in what manner and at what time information about individuals is released [28].

#### 2.1.1.1 Online self disclosure

As an extension to privacy, online privacy in particular is understood to include the steps taken to protect individual identities of internet users [41]. These users are defined as individuals who make use of the internet services to either express their opinions or to gather and distribute information.

Online self disclosure is both the information that a user reveals as well as the

ease with which this can be done in order to ensure the user can be identified as a real person. Self disclosure can be seen through two prisms, namely depth and breath. Disclosure depth is defined as the intent of disclosing accurate and honest information. Disclosure breath is considering how often and how much information is disclosed. More self disclosure leads to more certain interactions. If a user provides more information, the transaction is seen to be more reliable and it is for this reason that organizations usually demand the disclosure of certain pieces of information from a user.

The negative side of obtaining or disclosing information online can range from victimization and harassment to intellectual property theft. It is not all negative though: valuable information can be given to- and collected from authorities and organizations [35].

#### 2.1.1.2 Self Disclosure in the Mobile Environment

Smartphones and handheld devices that disclose a users physical location via geographical positioning systems have become very common. At the same time it is becoming easier to access web browsers and other online applications seamlessly while on the move, regardless of location. A lot of research has been done on protecting individuals reporting their locations and patterns of moving[4, 14, 19]. Individuals reporting such information usually do it to gain some benefit from doing so. As the market grows however, there are increasing threats of privacy loss, especially since users are not always aware of the particular privacy policies applicable to their situation [4, 14].

Not enough is being done to ensure the privacy and informed consent of users. There are shortcomings across platforms with regards to privacy policy. Devices with geo location capabilities include laptop, desktop and mobile software, GPS cameras and a host of applications on the web. Smartphones in particular present the opportunity for user interface- and service design level approaches to privacy. However, there are several different runtime environments and a large degree of fragmentation exists in the market, with differences in specification, deployment, installation and implementation. As an example consider the differences in how

applications install: some install in real time by downloading the necessary content from the web, while others are pre-installed and are then permanently available. Web resources are typically more secure through mechanisms such as policies for access, certificates and secured connections. However, even the web resources differ between different web browsers regarding consent and permission.

Considering the fragmentation and lack of standards, careful thought has to be given to how to provide a standard interface for mobile self disclosure of data collection participants. Ideally an application for contributing information should not require installation and should not put- and/or make a participant feel at risk because of a lack of clarity in privacy policy.

#### 2.1.1.3 Privacy Participants

Several variables factor into the amount and frequency of disclosures that individuals make. Studies have shown that elements such as age, social status, wealth, educational background, career and gender play a role in how much information individuals are willing to give out [17]. For example, it has been shown that the sharing- and privacy preserving actions of elders above the age of 55 is influenced by the desire to build social capital and expand relationships. Friends' action thus dictate, to some extent, the actions of individuals sharing information [11].

Furthermore, there exists distrust in the enforcement of law and the application of privacy protecting technologies. Considering how citizens understand data processing, it is clear that the public understanding and knowledge of the concepts of privacy, data protection and the legal framework that governs these concepts is lacking. This is especially true when privacy and protection is considered in terms of security and surveillance provided by authorities - a feeling of anxiety and uncertainty is present in society. [17]

The privacy protecting environment is seen as too complex and lacks clarity. This is understandable as it tends to be both an unknown and invisible environment. The very right to- and justification of privacy is a hard concept to define and changes as both the law and society progresses. Additionally, there is a lack of understanding of privacy policies by the public. What is clear however, is that the

public allocates a high priority to privacy and data protection [17, 40, 41]. People comprehend that mechanisms operate to protect privacy, but they are unsure how and why these mechanisms operate.

Citizens seem to believe in the general good intent of governments, but doubt their ability to control data as their good intentions would dictate to them. This could be due to media focus on data leaks. Convincing individuals of the trustworthiness of data gathering entities such as a medical- or crime databases, could thus be of paramount importance. Different state sectors enjoy different levels of public trust. Medical services for example are highly trusted, while local service providing authorities are not.

Apart from trust in authority, the public is also unsure of who is responsible safeguarding information. People have the perception that the crux of privacy protection lies with the data distributing entity and not the data processing itself. Reluctance to share information comes from the fear that information could be shared with unauthorised third parties or used without permission and knowledge of the user that has provided the information citeMohamed2012, Taddei2013. Once again this makes a strong case for convincing the public of reliability.

There is a lack of awareness of privacy as a fundamental right and the complexity and uncertainty associated with the privacy protecting environment leads to individuals taking unwanted measures, in terms of good data flow, to protect themselves. These measures could include providing false information or completely refusing to provide any information.

Considering the status quo, it becomes evident that solving the privacy problem lies at the intersection of legislation, technological ingenuity and citizens taking responsibility for themselves [41].



### 2.1.2 Information disclosure and privacy concerns

As outlined above, the relationship between individual privacy concerns-, trust- and control over personal information correlates with the amount of self disclosure users are willing to make. [35].

Furthermore, there exists a relationship among the amount of trust-, the level of concern-, and the cultural background that consumers have and the privacy statements that organizations make [40]. In order to gather data or provide services to consumers or participants, companies will usually promise better privacy to those who display more concern or less trust. Individuals protects from risks because they perceive an outcome to be severe, themselves to be vulnerable, or a particular response to be effective in addressing the risk. A failure to protect is usually because of response costs, the expected rewards associated with taking a risk, or because users believe in their self efficacy in dealing with a risk later on. However, users tend to teach themselves to exercise control over their privacy mechanisms in order to protect their information [17].

#### 2.1.2.1 Privacy Control

Control is the extent to which users can manage the amount of anonymity they preserve as well as the ability to alter disclosed information. Although it is a complex relationship, it is evident that the more users believe they are in control of information, the more trust they have. To better understand the complex relationship, consider that trust leads to a reduction in the perception of privacy risk. A lower perception of privacy risk leads to more disclosures. On the other hand, distrust leads to individuals avoiding negative consequences. This is done by selecting a path with the least risk, which is usually to share less information [11]. It's worth pointing out that when one has full control, trust is not required. There is thus a further relationship: This is between the privacy concerns of users, providing anonymity and the amount of self disclosure that users are willing to make.[35]

#### 2.1.2.2 The anonymity and privacy relationship

Considering both the negative and the positive aspects of information disclosure, it does happen that users wish to remain anonymous. Anonymity relates closely to privacy and is the ability to keep unidentified information in the public realm. Ideally, users would like complete anonymity. The problem with anonymity that is attained through means of never providing an associated identity is that it is nearly

impossible to complete a transaction without disclosing at least some personal information. The increase in the sizes of databases, the volume of collected personal data, the loss over personal data control and the possibility of an organization violating or misusing individual privacy, all contribute to a certain apprehension when it comes to participation. This apprehension leads to users declining to share personal information online. Users do realize the benefits of sharing accurate and detailed information and thus it is a trust issue between organization and individual that factors into a pro-con value judgement when it comes to information sharing. Exchange trust is the amount of user confidence that an organization will keep its privacy promises. More exchange trust will mean more information disclosure. Apart from anonymity posing the obvious problem of decreasing reliable information disclosures, it also has the potential to allow users to discriminate; fabricate information or display generally inappropriate behaviour. As an example consider a participant that makes fake reports on a system intended to collect actual data based on factual events or genuine individuals: Malicious participants who are guaranteed of their anonymity could make reports to paint a certain picture of an organization, another individual or even a whole area. This is obviously not desired.

#### 2.1.2.3 Privacy Practices

Various standards, guidelines and practices govern the flow of information. The most comprehensive are the five fair information practices outlined by the US Federal Trade Commission[40]:

1. Notice
2. Choice
3. Access
4. Enforcement
5. Security.

Notice means that a user understands what information he or she is providing and what it will be used for. Choice allows the user to decide how information

is collected and distributed. Access means that users can update or remove their information at any given time. Enforcement is typically legislation around policies that protects a user through the enforcement of law. Security is ensuring that data is accurate and secure and is typically where approaches like k-anonymity or l-diversity (discussed in Section 2.5) are used to protect individuals privacy [35].

Any framework, including a crime reporting framework, would thus have to conform to these five fair information flow practices.

### 2.1.3 Trends in preserving the worlds' privacy

The basic principles of privacy, the interest of individuals, as well as the fair information flow practices that have been put in place, have resulted in several trends that are noticeable in the sphere of online privacy preservation: Sensitive data such as medical data, including online data, is forced by law to be treated with extreme confidence and can only be released with consent from the patient. Apart from the legal framework that protects individuals, users are also starting to employ applications that range from encryption software to anonymizers and system cleaners. These mechanisms are described in the next section.



## 2.2 Protecting against collusions in data collection

One of the means of ensuring anonymity in data communication includes providing assurance between mutually distrustful respondents and an untrusted data miner. These method uses cryptography, mixed networks and forwarding between participants and a data miner to ensure anonymity. This is done in onion-like layers. A brief survey of some of two leading collusion resistant methods, their benefits, drawbacks and applicability to crime reporting is presented in this section.

Anonymity, along with accuracy and efficiency are key factors in data mining activities [2]. It is argued that data collection tasks in an online environment should be collusion resistant as well as adhering to the principles of data mining activities. Collusion attacks could be seen as an agreement between malicious parties to

obtain certain information. Collusion resistant data collection ensures that if all participants (including the data miner) in a data collection exercise are dishonest, they should still not be able to learn the correlation between a respondent and his/her response.

### 2.2.1 Mixed Network Shuffling

The collusion resistant approach by [9] attempts to ensure that participants responses are only used in the aggregate, which is to say it will only be used to form part of a collection of items gathered together to form a total quantity. The proposed protocol has two distinct parts or phases:

1. The respondents first encrypt and subsequently shuffles responses.
2. Shuffle integrity gets verified and the necessary decryption information is provided to the data miner.

Unlike previous approaches where third party shufflers have been used to ensure a safe shuffle, the proposed collusion resistant protocol attempts to eliminate the need for such a third part shuffler. The collusion resistant protocol is argued to be both efficient as well as secure. The protocol makes use of an anonymization game to evaluate effectiveness: If a dishonest participant can win a game with only negligible probability, then the protocol could be considered anonymous.

### 2.2.2 Hybrid Shuffling

Ashrafi et al. [2] proposes an approach that consists of random shuffling and cryptography to ensure the anonymity of responses by participants. A dishonest data miner and N-1 participants that are honest (with at least two honest participants) are provided for in this collusion resistant approach, while maintaining data integrity and adequate efficiency.

The proposed technique consists of five phases, including

1. Preparation by participants of their data and generating a code for partial verification

2. Sending of encrypted responses to data collector and

3. Collector sending all responses to each participant and shuffling by participants;

4. Participants verify each others honesty and

5. Finally the collector generates a random permutation of plaintext responses.

This hybrid approach uses randomized responses and makes use of the probabilistic ElGamal encryption technique, where the same message is encrypted several times. The multiple term algorithm that is utilized in hybrid shuffling ensures that trying to break the encryption using computational power and the cipher text alone will not succeed, seeing as the approach is semantically secure.

### 2.2.3 Criticism against collusion resistant approaches

A few of the drawbacks for anonymous data collection applications with a protocol such as the collusion resistant approaches include:

1. If any participant decides to reply/act dishonestly, the whole process is overthrown. Although a respondent will not be linked with his/her response, no responses will be usable.

2. The protocol also does not cover the possibility of all but one respondents & the miner being dishonest. A so called N-2 number of honest respondents is needed.

3. Although claiming to be much more effective than zero-knowledge proof methods, the collusion resitant protocol still makes use of several rounds of communication and encryption.

Neither the Hybrid Shuffling proposed by [2] nor the Mixed Network Shuffling by [9] distorts any of the values collected or change them in any form. However, the downside is the reliance on cross-communication and verification by participants and the assumption on encryption and decryption in particular order, as well as the fact that plaintext responses which contains information on individuals (private

information) is not considered. For the purpose of providing anonymity in crime data, the required back and forth communication of onion-layered data collection methods, makes it an infeasible solution.

An alternative to layered anonymity through collusion prevention, is privacy preserving distributed data collection and publication, discussed in the next section.

## 2.3 Privacy Preserving Data Mining

As indicated in Section 2.1, the collection, storage and publishing of information for whatever purpose, be it business, health, census, government or social work always entails privacy related consequences. To this extent there is a tradeoffin the distribution of collected data, where utility is often sacrificed for increased privacy, or alternatively more utility is sometimes gauranteed, placing individual privacy at risk [3].

### 2.3.1 Utility and Adversaries

Askari et al.[3] show that utility of a protection mechanism depends on:

1. The type of application of the published dataset and
2. The amount of knowledge a potential user possesses.

Considering this, a protection mechanism is evaluated on both:

1. The information that an adversary has about an original dataset in general,
2. The amount of information about a released dataset that would affect a specific user.

The adversary has the goal of de-sanitizing a published table. An adversary can form an attack with auxiliary/additional information gained from either:

1. Datasets that were publicly released because they were considered safe
2. Previously anonymized data, also called inference attacks

#### 2.3.1.1 Types of data

Adversaries who attempt to gather information from public databases can use several methods to try and do so: Queries posed at databases could aim to obtain information such as names and identification numbers of the individuals represented in the database. Alternatively, if names or identification numbers have been removed, adversaries could attempt to use a method called a linking attack to deduce results. Linking attacks combine external- or background knowledge with publicly available information not considered to be liability to disclose. To this extent, identity disclosure is releasing records with an individuals identity contained therein. Attribute disclosure is releasing data where an attribute can be used to infer information that can lead to identity disclosure. This is also known as inference disclosure. Samarati [31] states that data can be categorized as being:

- Non-identifying information depending on table (Marital Status)
- Explicitly identifying information (ID number, Name)
- Quasi identifier information (Age, Location)
- Sensitive information (Disease, Income, Crime)

Typically, explicitly identifying information should be removed from databases before being made available to third party data users. Sensitive information should be protected by indistinguishability from other sensitive values. Quasi identifier attributes (often just called QI values) can be linked with other tables or background knowledge to infer sensitive values. Table 2.1 presents some examples of each of these types of data values.



Table 2.1: Various Data Types

| Identifying Attributes | | Quasi-Identifier Attributes | | Sensitive Attributes |
|---|---|---|---|---|
| ID Number | First Name | Zip Code | Age | Crime Reported |
| 8909085001081 | Mark | 7701 | 23 | Rape |
| 8603085001081 | Joe | 7701 | 26 | Bulgary |
| 8509083323081 | Pierre | 7708 | 28 | Burglary |
| 7809083385081 | John | 7709 | 35 | Theft |

#### 2.3.1.2 Inference Channel Protection

Protection against inference channels focuses on the basic structures of frequent pattern mining that can be used to build complex models including clusters, classifications and associations. Protection against inference channels attempt to answer the question whether anonymity can be guaranteed amidst the disclosure of a collection of frequent patterns from data mining.

Construction of data inference channels refer to the act or process of deriving logical conclusions from premises about the data known or assumed to be true. Modelling inference channels of data relies on the use of a binary database with Boolean formulas. The problem with inference rules is that adversaries could deduce new or unknown patterns. Inference protection plans attempt to check that a set of published patterns ensures anonymity and if not, how to sanitize said patterns (preserving quality while cleaning data to ensure anonymity).

Inference channel protection strives to detect and block inference channels in patterns. Such detection and blocking would eliminate the need for excessive processing by avoiding unnecessary sanitization. If an inference channel is not detected the processed query can be published and if not, a new mining task can be issued with higher support thresholds (thus adding fake records), or the collection to be published can be sanitized. Some approaches to provide anonymity are tested over public, private and unknown data. Inference channel protection focuses on data where there is assumed to be no knowledge of what is sensitive; It does not consider semantics of attributes and generalization possibilities. All attributes are considered to be quasi-identifiers. Inference channel protection makes use of support counts and confidence measures to ensure anonymity and also considers infrequent mining as a potential attack route, where infrequent mining would be an adversary drawing inferences from data by looking at what is not published or provided.

The advocates for inference channel protection claim that partial data sanitization works better than total data anonymization strategies, partly because data mining is aimed at learning patterns, models and trends [5]. The argument is that these trends are less likely to be discoverable in partial sanitization and publishing

and this is specifically relevant where input data cannot be accessed more than once. However, such advocates use specific empirical evaluations in theoretical settings to prove that sanitizing a mined pattern is better than anonymizing all the source data since only pertinent patterns are anonymized. Inference channels are protected against with privacy preserving data publication approaches described below.

## 2.4 Distributing Collected Data

Various techniques for data publishing with privacy protection have been researched ([3, 9, 16, 20, 32]). Accordingly, the approaches to collected data distribution while preserving privacy can be divided into three broad groups:

1. Publishing micro data and answering research questions about data.
2. Authorization/authentication and restriction of trusted third parties
3. Allow third parties data querying with selective publishing, including:
    - Auditing
    - Output perturbation

A brief discussion of these three groups are given in the next three subsections:

### 2.4.1 Matrix Channels for publishing microdata

Matrix channels fall into the category of publishing microdata. Achieving anonymity is seen as a channel matrix through:

1. A representation of a dataset
2. A method to specify input- and output sets and
3. A way to compute conditional probabilities.

Matrix channel computations are done through mapping from input to output. A channel matrix takes input $a$ and produces output $o$. The matrix makes use of a

noisy channel (sending several inputs with several observable outputs). Bayes risk is used to measure the probability of errors. Finding the correct output is considered a hypothesis testing problem. Continuing along the lines of using probabilities for measurements, the error probability of a guess by an adversary is considered to be the privacy measure. Privacy operation cost is equal to utility loss in the sense that potentially valuable information is lost. A distance metric is used to measure the amount of utility lost and average utility degradation. The mechanism aims to measure both average utility preserved and the utility difference between a released and original dataset. It is important to note that generalizing essential target attribute values is basically destroying utility while generalizing other (non-essential) attributes should be considered more acceptable. The main drawback of matrix channels is the speculation associated with adversary knowledge and the difficulty of modelling that knowledge.

### 2.4.2 Restricted authorization of trusted third parties

In the context of collected data, it may be necessary for a trusted third party to gain unaltered access to original data for the purpose of acting on it. As two practical examples consider a doctor that needs to be able to access medical data of patients or an authorized police officer that needs access to unaltered crime reports. In such cases data should not be anonymized, but instead data access should be strictly controlled. Different levels of clearance should relate to different levels of information access. Typically data access would rely on a two factor authentication protocol with data encryption and biometric- as well as password verification [7, 26]. However, authentication in isolation is not a suitable solution to providing data to trusted third parties. It is infeasible to authenticate and assign clearance levels to every member of the public that requires access to crime or other data.

### 2.4.3 Querying Interfaces

Query auditing falls in the category of selective publishing. Querying interfaces essentially comes down to the trade-off found when limiting usability of a public database in order to preserve privacy of individual details recorded in the ta-

bles. Query auditing can be acheived through the different means of ensuring anonymization, including removing information, altering certain attributes/fields and adding statistical noise (See Section 2.5). Sleeper et al. [32], for instance, evaluated the usability of queried data after privacy policies have been applied to the data. They noted that if removal of search by certain fields is applied as a privacy enforcing mechanism on large data sets, the usability issues that emerge need to be mitigated properly in order not to hinder legitimate use of the data by the public. Often in order to preserve privacy, usability is sacrificed. Usability is limited by policy makers without sufficient evidence that it will significantly improve privacy. Such usability limitations might include removing certain search fields from database query interfaces or not allowing certain types of input [32].

The main criticism against querying interfaces though, especially in the scenario of crime reporting, is not just the usability issues, but the fact that multiple queries can be studied in conjunction by an adversary and used to draw inferences about individuals. In the same way as removing identifying attributes does not solve the anonymity problem, query interfaces cannot gaurantee privacy simply by altering each query result. An effort thus has to be made to properly anonymize data before queries are directed at the database.



### 2.4.4 Section Summary

There are clear benefits and drawbacks to different approaches in making collected data publicly available [24]. A hybrid approach, incorporating micro data publication, authentication and query limiting is a good alternative. However, selective publishing, especially in the context of releasing data to third parties, has proved to be a good way of balancing privacy concerns with data availability. Selective publication of data relies on "sanitization techniques". Some of these techniques are described in Section 2.5.

## 2.5 Privacy Preservation Through Sanitization

Considering the legitimate privacy concerns of those who contribute to on-line data collection, several approaches have been adopted to attempt to guarantee

user privacy. Privacy through anonymity in released data is achieved through various "sanitization mechanisms", which could include techniques of generalizing-, anatomizing and/or randomizing data. In order to better understand the concepts of adversaries mentioned in Section 2.3 and how sanitization can protect against said adversaries, it is best to consider the different data sanitization mechanisms. A brief outline of these mechanisms looks as follows:

### 2.5.1 Suppression

Suppression is an attempt to make rows in a table indistinguishable from each other by changing row and column values to a particular suppressed value, most often "*".

Table 2.2: Example Table

| Year of Birth | Gender | Sensitive |
|---|---|---|
| 1985 | Male | TRUE |
| 1990 | Male | TRUE |
| 1990 | Male | TRUE |
| 1990 | Male | FALSE |
| 1985 | Male | FALSE |
| 1985 | Female | FALSE |
| 1985 | Female | FALSE |
| 1985 | Female | FALSE |
| 1991 | Female | TRUE |



As an example consider a general table such as in Table 2.2 with three columns of which one is a sensitive value (that should not be disclosed). This table would be suppressed by changing informative values to suppressed values, as indicated in Table 2.3. Consequently, Table 2.3 is said to be a suppressed representation of Table 2.2.

### 2.5.2 Generalization

In contrast to suppression, the generalization approach changes values in a table according to some taxonomy or hierarchy into a more general or less informative

Table 2.3: Suppressed Table

| Year of Birth | Gender | Sensitive |
|---|---|---|
| * | * | TRUE |
| * | * | TRUE |
| * | * | TRUE |
| * | * | FALSE |
| * | * | FALSE |
| * | * | FALSE |
| * | * | FALSE |
| * | * | FALSE |
| * | * | TRUE |

value. Generalization thus provides anonymity through indistinguishably while at the same time preserving some information about attributes that might otherwise be lost through suppression. The reasoning is that more general values are more representative and have a higher probability of containing other tuples in their branch nodes[38]. It is quite clear, for example, that the year range 1990-1994 covers more values than 1991 does. Table 2.4 is said to be a generalization of Table 2.2.

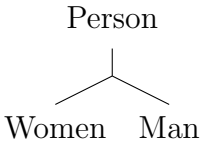
Suppression can thus be seen as a specific case of generalization where values are generalized to their least specific or least informative states. This is because on a taxonomy tree, "*" would typically be a root node.

Table 2.4: Generalized Table

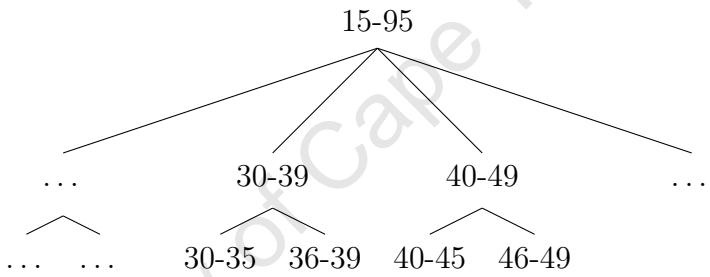
| Year of Birth | Gender | Sensitive |
|---|---|---|
| 1985-1989 | Person | TRUE |
| 1990-1994 | Person | TRUE |
| 1990-1994 | Person | TRUE |
| 1990-1994 | Person | FALSE |
| 1985-1989 | Person | FALSE |
| 1985-1989 | Person | FALSE |
| 1985-1989 | Person | FALSE |
| 1985-1989 | Person | FALSE |
| 1990-1994 | Person | TRUE |

#### 2.5.2.1 Taxonomies of Generalization

A generalization taxonomy of the attribute gender can be understood to look as follows:

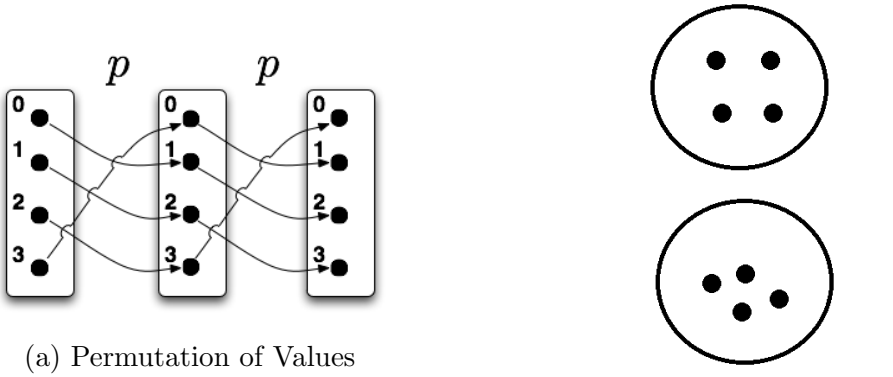



Values such as those found in age, year of birth, income and other integer values would have a taxonomy tree similar to the age tree below:




### 2.5.3 Permutation

Permutation in general terms can be understood as the rearranging of values. One example of using permutation of data for the purpose of ensuring anonymity is the process of finding a suitable model to describe data and generating new data based on the parameters of this model. The generated new data serves as a replacement for the original data and in so doing conceals the original data [1]. Permutation is interesting for theoretical data mining expirementation after anonymization, but in practice it has the disadvantage of producing values that does not exist in the original data. Regenerated data can thus contain invalid tuples, which could in turn provide information to adversaries relating to which values in the anonymized data was changed. As an example, consider an adversary that sees an invalid address that refers to a non-existent street number and consequently




(a) Permutation of Values

(b) Reclustered Values

Figure 2.1: Example of Permutation Process

knows this value was obtained from permutation.

### 2.5.4 Perturbation

Perturbation of data is the process of exchanging or swapping values for others in the original dataset. In doing so, distortion is introduced and provides a measure of anonymity by means of uncertainty.

Although perturbation techniques ensure that adversaries have less certainty about data, perturbed data is also confusing to analyze and draw conclusions from. Inaccuracies in data leads to significant utility loss. An example would be swapping areas in crime reports and consequently making data available to third parties that is inaccurate. Such data could then be wrongly applied to protect against a non-existent threat.

Permutation and perturbation are no longer considered suitable approaches for anonymizing real world data, seeing as these techniques are mainly used for statistical purposes and not for practical use of data after publication.

## 2.6 Determining Privacy Requirements

Sanitization of data values, as described in Section 2.5, occur when some threshold of predefined privacy is not met. One of the best known approaches is k-anonymity.

K-anonymity has been used in numerous applications, studies and scenarios to provide a suitable means of providing privacy with minimal information loss. Applications have ranged from clustering and anonymizing moving objects [4] and streaming data [10, 22] to providing privacy in social network data [8].

### 2.6.1 K-anonymity

To ensure that private information is not leaked in publicly available databases and tables, information such as names and identification numbers, that could serve as unique identifiers, are removed. Even after removing this information, up to 87% of the US population could be uniquely identified using harmless information like birth date, gender and zip codes [25].

As far as the semantics of data values are concerned, two types of values exist:

1. **Sensitive attribute values** (also called identifier attributes) are those values that give out sensitive information about an individual such as income or disease

2. **Non-sensitive attribute values** (also called quasi identifier attributes or quasi values), are attributes that arent sensitive but can be combined and linked to external data to identify individuals. Examples include age or gender.

K-anonymity tries to ensure that quasi identifiers are always generalized according to a hierarchy, duided measured by an information loss metric. This can happen as global or local recoding on a full domain or on sub trees depending on the application [3, 34, 39].

K-anonymity is the anonymization of quasi-identifiers. Quasi-identifiers are sets of non-sensitive attributes such as birth date and gender that could be combined (sometimes with external data) to link sensitive attributes to individuals. A table is k-anonymous if k-1 other records exist with the same quasi-values for each sensitive value. The most important consideration of k-anonymity is that it achieves an acceptable trade-off between preserving privacy and utility by means of a k-threshold, where the k-threshold is the value that k is set to based on a value

selected to ensure the privacy of participants.

As an example, suppose we select k = 5 in a table with a boolean sensitive attribute field and two non-sensitive (quasi) attributes namely date of birth and gender. If the sensitive field is negative, there should exist at least 4 records that has the same value for Year of Birth and Gender as the record listed as negative. This should be true for all sensitive value fields. Such a table could then be seen as k-anonymous, where k = 5.

Table 2.5: 5-Anonymous Table

| Year of Birth | Gender | Sensitive |
|---|---|---|
| * | Person | TRUE |
| * | Person | TRUE |
| * | Person | TRUE |
| * | Person | FALSE |
| * | Person | FALSE |
| * | Person | FALSE |
| * | Person | FALSE |
| * | Person | FALSE |
| * | Person | TRUE |
| * | Person | TRUE |



### 2.6.2 l-diversity

It has been argued that k-anonymity has certain limitations [25]. These include a lack of diversity among sensitive attributes as well as failure to protect against attacker background knowledge. A table that is $k$-anonymous has at least $k$ values in all equivalence classes. The problem with $k$-anonymity is that sensitive values might not be evenly distributed and even if there are $k$ amount of equivalences, an adversary can still deduce that simply because an individual is in a table she also has a particular sensitive attribute. Two types of disclosure can occur:

**Positive disclosure** is where a sensitive attribute can be identified correctly.

**Negative disclosure** is when an adversary can correctly eliminate some possible values of the sensitive attribute.

To counter this problem, the notion of $\ell$-diversity was suggested. Consider $f_1$ to be the most frequent sensitive attribute value of an equivalence class. To satisfy $\ell$-diversity, we say that $f_1 \leq \frac{1}{\ell}$.

$\ell$-diversity is thus an expansion of the application of k-anonymity. $\ell$-diversity defines $\frac{1}{\ell}$ as highest allowable frequency of quasi identifiers values to be included, in order to ensure anonymity. $\ell$-diversity makes use of entropy, which is a measure of the amount of information lost. $\ell$-diversity works on much the same principle as k-anonymity and for each sensitive attribute, an adversary would have to eliminate at least L-1 sensitive attribute values for a positive disclosure.

Thus, $\ell$-diversity can be understood to privacy requirement relating to the most frequent sensitive value in an equivalence class (where an equivalence class is tuples in a dataset that have the same identifier attribute values), which must not have a frequency of more than $\frac{1}{\ell}$. If we have an equivalence class of size 10 and the most frequent sensitive value occurs five times, we can say that the frequency of the most frequent sensitive value ($f_1$) is $\frac{5}{10}$. If we set $\ell = 2$, this constraint is satisfied. If we set $\ell > 2$ it will not be satisfied and thus the most frequent value must be reduced. It also follows that the less frequent sensitive values (e.g. $f_1, f_2, f_3$) will have frequencies that meet the constraint if the most frequent value meets the constraint. Frequency of sensitive values can be reduced by moving tuples to different equivalences classes or creating new equivalence classes.

Of the different forms of data anonymization as outlined in Section 2.5, $\ell$-diversity makes use of generalization and it is achieved by dividing ordered attributes into partitions and partitioning categories higher up a hierarchical ladder.

#### 2.6.2.1 The case for l-diversity

$\ell$-diversity is seen as both an extension and improvement of k-anonymity. This is partly because of possible attacks on k-anonymity that was not initially accounted for. Some unforeseen attacks against k-anonymity include:

- An attack making use of the composition of like parts or similar attributes in a so-called homogeneity attack. Such an attack is possible because there is not enough diversity contained in sensitive attributes. This is a result of k-anonymity ensuring the diversity of quasi-identifiers (non-sensitive attributes) and not diversity in sensitive attributes.

- An attack making use of background knowledge is also possible. K-anonymity does not take into consideration that attackers could have background knowledge when looking for sensitive information. This background knowledge could be:

  **Domain knowledge** is knowing that attribute 2 denotes gender.

  **Demographic background knowledge** is knowing something about someone on a personal level.

  **Instance level knowledge** is the probability of something occurring when something else occurs.

In the past Bayes-optimal privacy approaches have used prior and posterior belief to measure the success of an adversary. Machanavajjhal et al. [25] outlined several problems with Bayes-optimal privacy.

- The fact that the data miner might have insufficient knowledge
- There is no way of knowing the adversaries knowledge of the joint distribution
- Multiple adversaries may exist
- The personal/instance level knowledge mentioned earlier is impossible to model.

These limitations are relevant to crime data because as has been shown, crime data can be considered to be as sensitive as income or medical data. In contrast to Bayes-optimal privacy, $\ell$-diversity ensures privacy even in cases where the publisher is not aware of what knowledge an adversary has. $\ell$-diversity modifies existing k-anonymity approaches to include an improved version of Bayes-optimal privacy. $\ell$-diversity defines $\ell$ amount of sensitive values to be included to ensure anonymity.

$\ell$-diversity makes use of entropy (a measure of the amount of information lost). $\ell$-diversity works on much the same principle as k-anonymity and for each sensitive attribute, an adversary would have to eliminate at least $\ell$-1 sensitive attribute values for a positive disclosure. $\ell$-diversity addresses the four problems listed earlier pertaining to Bayes optimal privacy:

1. $\ell$-diversity does not require full knowledge of the distribution
2. $\ell$-diversity does not require equal amounts of knowledge, seeing as a larger value for l ensures better the privacy
3. This covers multiple adversaries as well
4. Background knowledge is also defended against, seeing as a higher $\ell$ value will also treat background knowledge as a ruling out problem.

Machanavajjhala et al. [25] argue that $\ell$-diversity is practical, implementable, understandable and sufficient. It also preserves privacy while retaining utility. The final property is that it takes advantage of the principle of monotonicity: if a generalized/sanitized table preserves privacy, then every generalization of that table also preserves privacy. However, [13] studied the tuple minimization problem in order to find the optimal minimal generalization of a dataset with a factor $c \ln l$ for a constant $c > 0$. It was found to be impossible.

In addition, [13] also investigated the approximation of $\ell$-diversity. As an extension of the tuple minimization problem it is clear that no algorithm better than $m \ln n$ can be found to estimate an approximation of an $\ell$-diverse table. If one considers $|T_s|$ to be the number of all distinct sensitive values in a dataset, then the time complexity of the approximation algorithm to achieve $\ell$-diversity is $O(n^{2|T_s|+1})$. This time complexity makes it infeasible for large real world datasets. The practical implication is that in crime data anonymization, optimal $\ell$-diversity will not be obtainable with large datasets.

A suggested solution is parametrized complexity, which is the process of fixing constants on both ends of eligible anonymization values in order to determine if it has an impact on the time complexity of an approximation algorithm. Dondi et

al. [13] found that $\ell$-diversity with constraints $(|T_s|, m)$, with a constant value for $m$ is indeed fixed parameter tractable or fixed parameter controllable. $\ell$-diversity thus offers a promising prospect for increased privacy, but an optimal solution of data anonymization will have to rely on fixed anonymization parameters.

### 2.6.3 Extending l-diversity to (t,l)-diversity

As an extension of the disclosures mentioned in Section 2.6.2, two types of probabilistic disclosures exist:

- The first is where an adversary can determine, with high probability, that a target does not have a particular sensitive attribute value. This is called negative disclosure.
- The second type of disclosure is positive disclosure and entails an adversary that is able to deduce with high probability that a target is within a relatively small set.

$\ell$-diversity techniques often puts users in the position of having to make a trade-off between utility and privacy. It is possible to generalize sensitive values though, as well as extending the original $\ell$-diversity approach to control the frequencies of sensitive values in equivalence classes (A table that is $k$-anonymous has at least $k$ values in all equivalence classes). The effectiveness of this approach is measured by implementing the technique with a heuristic algorithm to find a table of suitable quasi identifier attributes. This table takes into consideration the different importance levels of QI values. The technique is called functional $(\tau,\ell)$-diversity [36].

Functional $(\tau,\ell)$-diversity makes use of taxonomies where weights are assigned to leaf nodes. Leaf nodes are the base values in a hierarchy. This can be used to describe the level of background knowledge that an adversary has. Base tuples are initially divided into equivalence classes and then moved to different equivalence classes based on quasi identifier values that have the biggest impact on frequency. If no suitably big class exists (one which can satisfy the $(\tau,\ell)$-diversity requirement), then sensitive attributes are generalized. Sensitive atrributes are generalized based

on covering the most dominant base sensitive attribute in the equivalence class. Where more than one such base attribute exists, the one with the least general value is selected for generalization. The process is repeated until the $(\tau,\ell)$-diversity requirement is satisfied.

#### 2.6.3.1 The case for (t,l)-diversity

$(\tau,\ell)$-diversity is implemented with the goal of avoiding excessive protection. An expansive technique is needed for various forms of data publishing, including the often mentioned medical record data anonymization, but also other forms of data publishing such as insurance history of individuals. Different data has also been considered for anonymization, this ranges from social network data and streaming data to attempting to present knowledge models of data.

It is argued that $\ell$-diversity poses the problem of having "narrow eligible ranges" for anonymization and thus result in datasets of lower utility. A dataset can be said to maintain some form of utility if it is not presented in its most general form. To obtain a table that offers some measure of utility, the largest possible value for $\ell$ is equal to $\frac{1}{f_1}$. It is intuitive to see that as $\ell$ increases, so too does $f_1$ decrease. A narrow eligible range refers to the varying degrees of anonymization routes that is available to follow considering a certain privacy requirement like $\ell$-diversity.

The contradiction here is that to preserve privacy and avoid the risk of either a positive or negative disclosure, a user is usually forced to specify a high value for $\ell$, yet this leads to data with low utility. Several measures to improve $\ell$-diversity have been explored, but few have looked at generalizing sensitive values as well. $(\tau,\ell)$-diversity is described as being more flexible and expansive than existing $\ell$-diversity techniques. It does however assume that taxonomies are always readily available or easily obtained. This is not always the case. Furthermore, a very specific privacy requirement might very well cause other approaches with their respective algorithms to fare worse comparatively.

### 2.6.4 Anonymity in different applications

It is not as easy as might initially seem to apply privacy preserving algorithms to different forms of data. This is evident if considered in the light of, for example, attempting to anonymize social network data or location data. The need to anonymize location information of participants from different applications, including gps data, gaming data and social network data, is becoming increasingly clear [27]. Crime reporting data could also potentially rely on relaying location data. Such data is required for purposes of a different nature -research or commercial.

Traditional privacy preserving techniques consider data tuples to be independent of each other, if not independent from external knowledge.

Participants might not want to reveal information about the police station, hospital or other private places they visit, as well as not having their residential address published to the general public.

Various forms of data can be anonymized using $\ell$-diversity as privacy requirement, provided that data is represented in a suitable format, an adversaries background knowledge is properly accounted for as well as paying attention to application specific challenges like those in modelling possibly interrelated crime report data.

## 2.7 Summary

It is evident that there is no one standard for mobile devices in data collection, anonymity, privacy and data distribution. However, some approaches, such as generalization have proven to be more effective and scalable than others such as onion layered reporting for anonymity. Chapters 3 and 4 outlines the integration of some these technologies and approaches to address the anonymous crime reporting scenario.

# Chapter 3

# Mobile Crime Reporting Framework



Considering the first research question on how to address the challenge of providing a platform for those who wish to participate in crime reporting, this chapter aims to introduce a framework that could serve as such a platform. To this extent a systems requirements analysis was conducted. A brief outline of the system requirements is provided in Section 3.1. Section 3.2 then gives an architectural overview of the integrated framework's different modules, whereas Section 3.3 describes the various modules in more detail. The anonymization procedure and selection of an anonymization method is explained in more detail in Chapter 4 on page 46.

## 3.1 System Requirements

The objective of our crime reporting framework is to ensure that data is retrieved from a participant's mobile device and stored in a database where an anonymization algorithm processes the data to remove all privacy violating attributes.

As part of the system requirements definition stage, the following questions were addressed:

1. What types of devices exist to report crimes on and what do users prefer?
2. What information is vital when analysing a transgression description?

3. What mobile service can be made available to collect data or what is the best approach to providing a mobile crime reporting service?

4. What anonymization algorithm is best suited to ensure participants' privacy?

The crime reporting environment consists of potential participants who own different mobile devices (with differing levels of capability) that could be used to make reports.

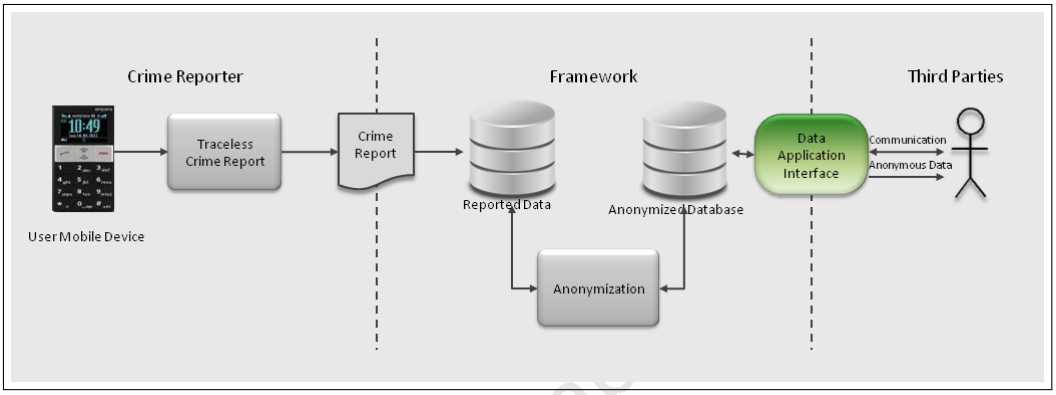



Figure 3.1: Crime Reporting Requirements Analysis

## 3.2 Integrated Framework

The system requirements discussed in section 3.1 highlight the need for different modules dedicated to generating privacy-preserving reports from the reported crime data. Summarily, our proposed framework is comprised of four modules, namely an information retrieval module, database module, a user authentication module and an anonymization module. The combination of these modules in one framework is illustrated in figure 3.2.

### 3.2.1 Framework Information Flow Process

A report is captured by running a Python script containing Extensible Markup Language. This script captures the reporters' description of the crime and transfers the information to a back end web server where the data/information is stored in a database for processing. A trusted party (such as a police officer who needs to

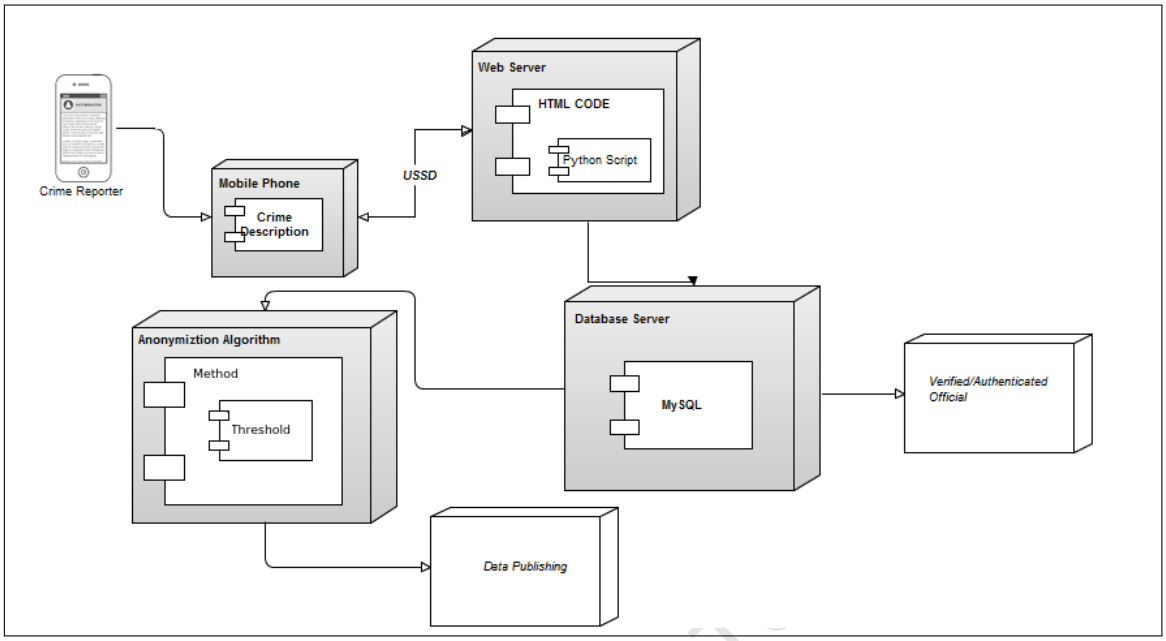



Figure 3.2: Framework Layout

act on reported crimes) can access the data in its unanonymized form. When data is needed for statistical or other purposes, the system can publish data to third parties with the optimal anonymization algorithm based on performance.

## 3.3 Crime Reporting Modules

Given the presented overview of the framework, we are now ready to describe the functions of the components and show how they jointly achieve the goal of enabling anonymous crime reporting.

### 3.3.1 Capturing Information From Mobile Devices

The information capture module corresponds to the "Crime Reporter", "Mobile Phone" and "Web Server" elements of the framework layout as illustrated in Figure 3.2. Our solution for mobile data collection is based on the Unstructured Supplementary Service Data (USSD) protocol. This is a simple text exchange service that allows for menus to be displayed on a mobile phone and enables participants

to interact with these menus. USSD enables a particularly interactive user experience by virtue of being synchronous - reports are recorded as they are made and participants receive immediate responses. In contrast to other services like short message service (SMS), these menus can enable one to limit/control participant responses so they are usable and manageable and not unintelligible. Abbreviating responses in a potentially expansive and elaborative application such as an online crime report is particularly useful.

#### 3.3.1.1 Information Retrieval & Storage:

When a participant dials the USSD number for reports from a mobile station (MS), the MS contacts the web server. The mobile station then interacts with an application interface. The application interface provides uniform resource locator (URL) text strings that guides the user on the options of submitting subsequent responses.



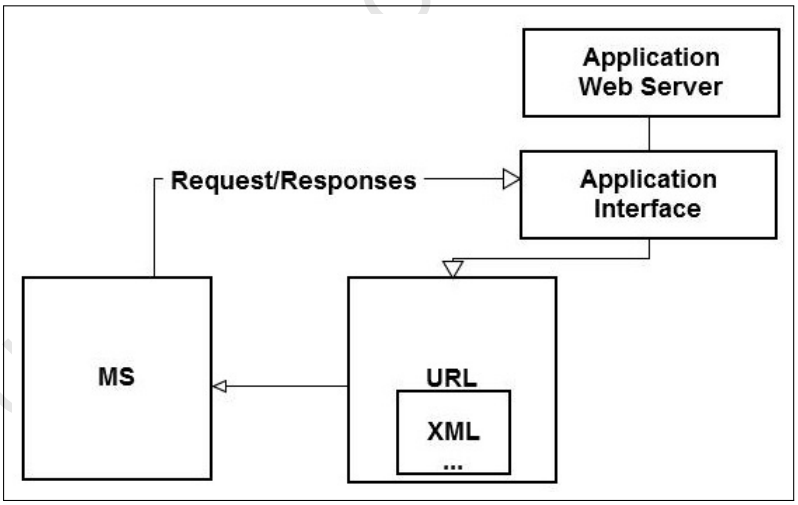



Figure 3.3: USSD Architecture

As illustrated in Figure 3.3, the application designed for a particular USSD purpose creates a menu structure for the mobile user and will create a response URL based on the subscriber's selected menu option.

#### 3.3.1.2 Application Interface XML

When the USSD request URL is called, the Extensible Markup Language (XML) located at the URL is read. The interpretation of the XML code relays to a mobile station what the different response options are and parses the participant input to the application interface. Listing 3.1 illustrates extensible markup languange written for handling crime reports. In this example a participant has selected violent crimes as a crime category, as outlined in line 3 in the headertext. A participant must then make a subcategory selection, with the options comprising of those listed in lines 6, 8, 10, 12, 14 and 16.

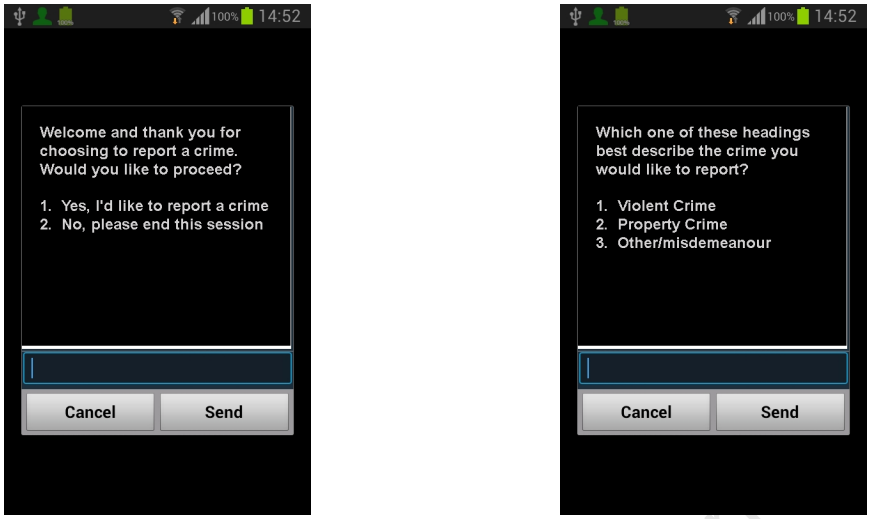
Listing 3.1: XML Setup

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <request>
3 <headertext>Which option best describes the violent crime?</headertext>
4 <options>
5 <option command="1" order="1" callback="http://markpc.cs.uct.ac.za/menu2?cmd=1"
6 display="true">Assault</option>
7 <option command="2" order="2" callback="http://markpc.cs.uct.ac.za/menu2?cmd=2"
8 display="true">Rape</option>
9 <option command="3" order="3" callback="http://markpc.cs.uct.ac.za/menu2?cmd=3"
10 display="true">Murder</option>
11 <option command="4" order="4" callback="http://markpc.cs.uct.ac.za/menu2?cmd=4"
12 display="true">Robbery</option>
13 <option command="5" order="5" callback="http://markpc.cs.uct.ac.za/menu2?cmd=5"
14 display="true">Family/Domestic</option>
15 <option command="6" order="6" callback="http://markpc.cs.uct.ac.za/menu2?cmd=6"
16 display="true">Other</option>
17 </options>
18 </request>
```



#### 3.3.1.3 Crime Report USSD Implementation

Given the various benefits of USSD for crime reports and the simple working structure of XML in application interfaces, we set up a USSD Crime Reporting service




(a) Start Screen (b) A Directed Response

Figure 3.4: Example screenshots of USSD menus

in order to get prototype data for the anonymization module[1].

For our application a USSD number was set up at *120*12021# from any South African mobile phone. As an experimental setup, postgraduate students were asked to submit pseudo crime reports from their mobile phones, relating to crime incidents they have been confronted and/or had family members confronted with. The ability to direct responses proved invaluable in gathering data from a selected group of participants at the University of Cape Town, where it is envisaged that this project will be implemented for the benefit of the larger student population. An example of a directed response is available in Figure 3.4. These directed responses are the practical application of the XML code highlighted in Listing 3.1.



### 3.3.2 Database Module

This section aims to explain how the database module was configured to interact with the application interface by interpreting XML responses and parsing it to MySQL tables. The database module corresponds with the "Database Server"

[1]This data and the extrapolation thereof is discussed in detail in Chapter 4 on page 5.1.3.1

element of the framework layout illustrated in Figure 3.2 and highlights how this element interacts with other elements in the framework.

#### 3.3.2.1 Database Interaction

The responses logged from the application is parsed to a database, in our case divided into the different categories in a crime report. Based on data hierarchies described in Section 4.2.1, these categories include amongst others the reporter category, location, crime type and sub type.

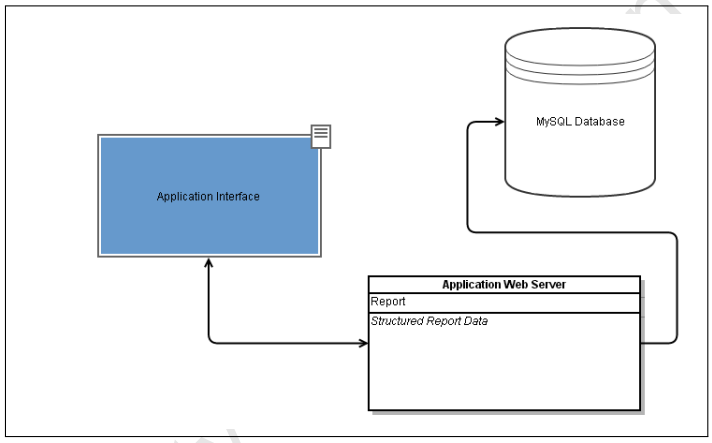



Figure 3.5: DB interaction with API

An example of the setup for Python code execution to interact between the database server and the application interface is presented in Listing 3.2. A main menu structure is outlined and the user response is defined by the XML code embedded in the Python execution. Different XML content, such as the content found in Listing 3.1 on Page 39, is applicable to different steps in the crime report. The full setup details for the server can be found in Appendix 4.

Listing 3.2: Python API Management

```
1 from flask import Flask
2 from flask import request
3 from flask import Response
4 import nltk
```

```
import MySQLdb
import time

app = Flask(__name__)

@app.route("/main")
def mainapp():
    content = """<?xml version="1.0" encoding="utf-8" ?>
<request>
<headertext>Please select a crime category?</headertext>
<options>
<option command="1" order="1" callback="http://markpc.cs.uct.ac.za/menu1?cmd=1"
display="true">Violent Crimes</option>
<option command="2" order="2" callback="http://markpc.cs.uct.ac.za/menu1?cmd=2"
display="true">Property Crimes</option>
<option command="3" order="3" callback="http://markpc.cs.uct.ac.za/menu1?cmd=3"
display="true">Misdemeanors/Other</option>
</options>
</request>
"""
    r = Response(content, content_type="text/xml");

    return r
```



#### 3.3.2.2 Database Layout

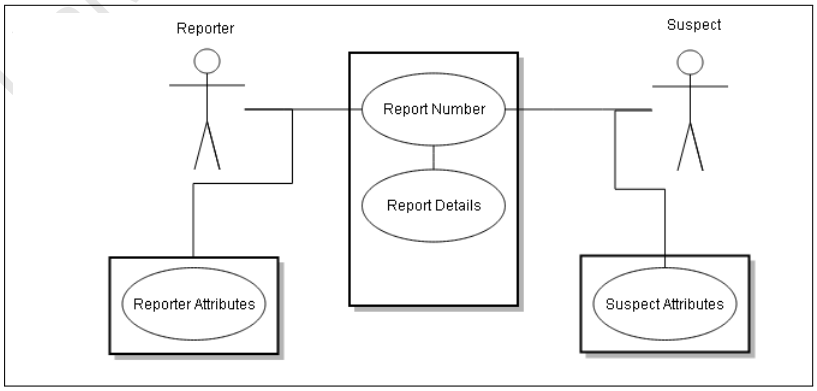



Figure 3.6: Use Case Database Diagram

The information parsed from the application interface to the database is set up to be stored in a MySQL database, with table fields suited to crime report data. Each crime report has a reporter and possibly a suspect. Each suspect will be matched to a report number. Figure 3.6 further clarifies this breakdown.

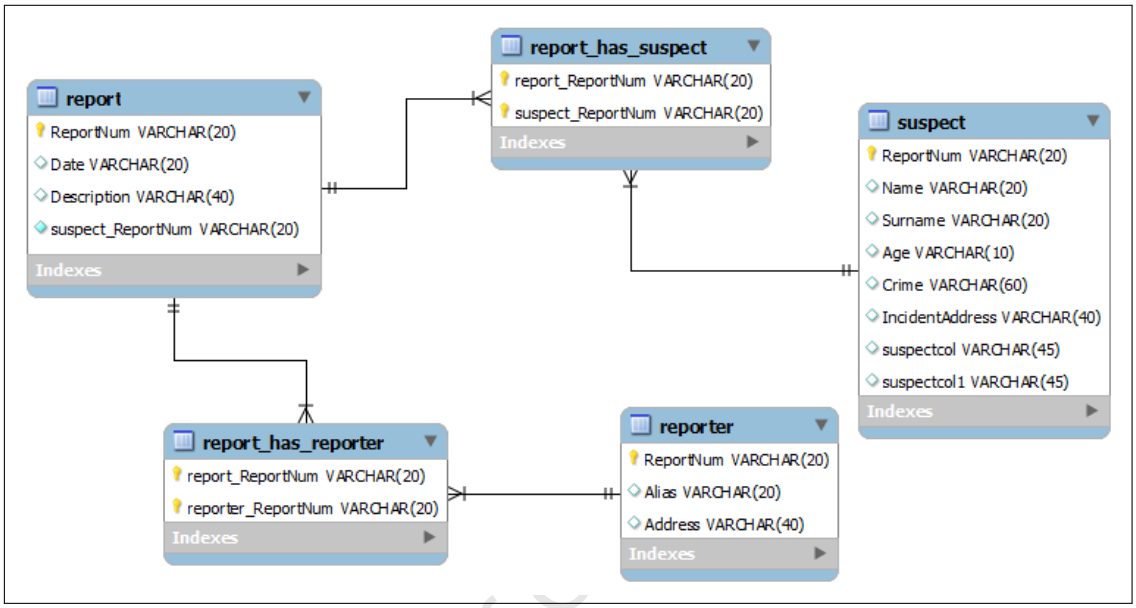



Figure 3.7: Database layout

Fields for the Report, suspect and reporter with various attributes related to each are illustrated in Figure 3.7. Each suspect has fields for a name, surname, age, crime and incident address. These broad fields were determined based on the Federal Bureau of Investigation (FBI) Uniform Crime Reporting Manual [15]. Each reporter on the other hand has an alias and a home address.

### 3.3.3 User Authentication

The user authentication module corresponds with the "Verified/Authenticated Official" element of the framework layout in Figure 3.2 on page 37. A hierarchical scheme that is linked to the anonymity level was designed and built into the java implementation. In this mechanism a qualified official can get access to unchanged data as a super user and third parties can get access to anonymized data.

Data access relies on a password protected access control mechanism that sets k=0 for an authenticated official and higher levels of k for different levels of access. Standard password verification is used to ensure that only trusted individuals with the appropriate privileges have access to information via this access control scheme. We suggest that the access control scheme could be supported by a two factor authentication mechanism with data encryption and biometrics. In addition, privacy policies of an organization implementing the framework could dictate who gets access on a case by case, need to know basis.

An illustration of the authorization request is presented in Figure 3.8.

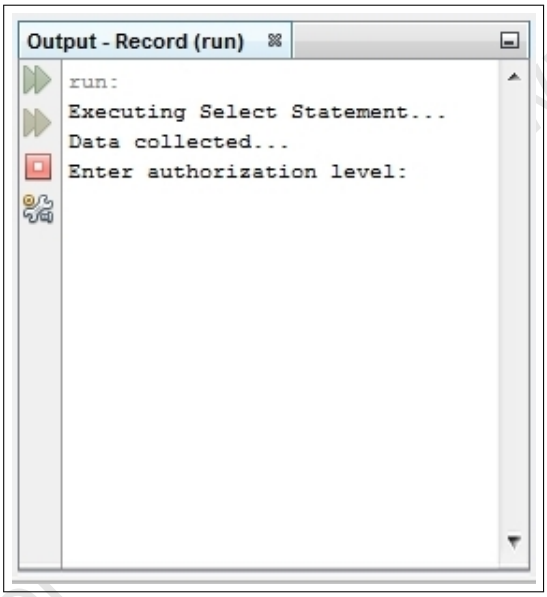



Figure 3.8: Authorization Request

### 3.3.4 Anonymization of Data

After data has been recorded in a database using the platform outlined in previous sections of this chapter, the anonymization module relies on a defined threshold to provide privacy protection in data. A threshold can be understood to be a privacy requirement that refers to setting the size of k in k-anonymity or $\ell$ in $\ell$-diversity. This decision is usually made based on how high the security level needs to be, i.e. how many identical tuples need to exist to ensure anonymity for a particular value [38]. At its most basic level k can be seen as the pre defined trade off between

privacy of individuals represented in the database and utility of the published dataset.

The privacy preserving algorithm that forms part of the anonymization module is described in detail in the following Chapter.

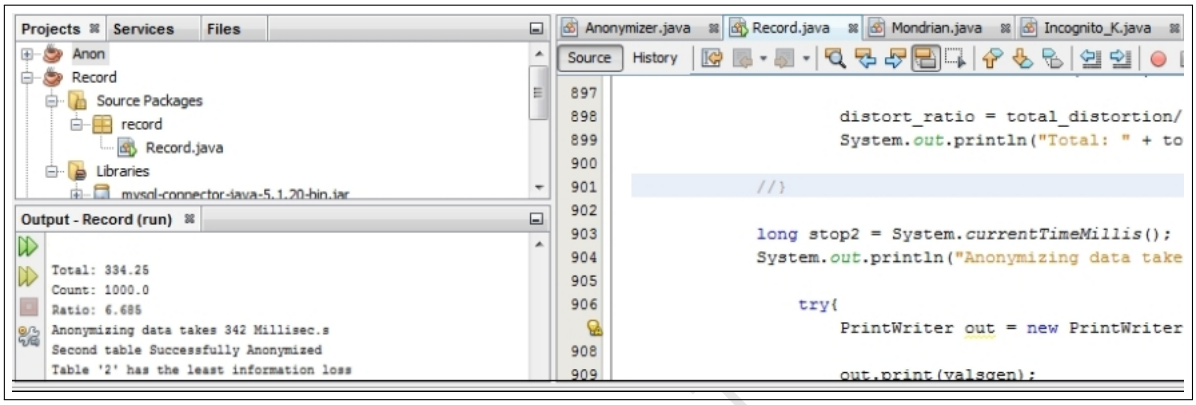



Figure 3.9: Completed Anonymization Process

Figure 3.9 illustrates the completed anonymization process for a third party requesting anonymized data, with the results from the anonymization process indicated in the bottom left outuput results.

# Chapter 4

# Implementation

Data anonymization and usability of mobile applications are topics that are typically studied separately. In the mobile crime reporting scenario however, both topics need to be studied in conjunction. The previous chapter introduced a framework for crime reporting, which includes a module for anonymization of data. This chapter is dedicated to expanding on the problem of distributing collected data by highlighting existing algorithms used in privacy preserving data publishing as well as introducing a hybrid algorithm termed "Crimemod" that was specifically developed to answer the research question: *What is the best way to go about anonymizing crime report data for use in various applications, including crime statistics databases and crime area avoidance applications?*



## 4.1 Anonymity in Crime Data Publishing

Of particular interest to publishing crime data are anonymization algorithms that "sanitize" datasets before or during publication. Such algorithms include ones that are based on the k-anonymity technique and special cases of this technique include algorithms that are based on enforcing $\ell$-diversity [25]. The k-anonymity approach is applicable to publishing crime data because it is an automated process and does not require human aided mechanisms to produce safe- and private/anonymized data. In addition unlike querying interfaces, k-anonymity algorithms do not disclose information by omission and so are secure to inference attacks.

| Age | Gender | Crime |
|---|---|---|
| 33 | Male | Arson |
| 25 | Male | Rape |
| 65 | Female | Drug related |
| 39 | Male | Vandalism |
| 39 | Male | Vandalism |
| 53 | Male | Rape |
| 47 | Female | Drunken Driving |
| 21 | Female | Burglary |
| 19 | Female | Drug related |
| 30 | Female | Theft |

Figure 4.1: Extract from Crime Report Data

As a crime data specific example, suppose we select k = 2 in a table similar to that in Figure 4.1 with a sensitive attribute field "Crime" and two non-sensitive (quasi) attributes namely "Age" and "Gender". Suppose the sensitive field could be any value as listed in Figure 4.1, while "Age" and "Gender" could be the normal values for these fields. Then if the "Crime" attribute is "Vandalism" for example, there should exist at least 1 other record that has the same value for "Age" and "Gender" as the record listed with this particular sensitive value. This should be true for all sensitive value fields. Such a table could then be described as k-anonymous[2].

$\ell$-diversity introduces an additional requirement, which states that equivalence classes only have tuples of $\frac{1}{\ell}$ frequencies for sensitive values. It is argued that the problem with k-anonymity is that sensitive values might not be evenly distributed and even if there are k amount of tuples in an equivalence class, an adversary can still deduce that simply because an individual is in a table he/she also has a particular sensitive attribute[23, 25]. To counter this problem, the notion of $\ell$-diversity was suggested [25]. Although it is an interesting restriction especially in medical datasets, it is not immediately relevant with regards to crime data. The "crime" attribute is considered sensitive value practical purposes, but the goal in anonymizing crime data is to consider all attributes to be subject to inference attacks and hence even the "sensitive" value should be subject to generalization.

T-closeness is another privacy restriction based on the idea that the distribution of a sensitive attribute in an equivalence class should be close to the distribution of the attribute in the overall table [23]. As datasets expand based on more reports and the crime reporting framework is introduced on a larger scale across different locations, it is conceivable that these latter restrictions could perhaps become more relevant, especially if fields like the crime reporter category is labelled the sensitive field.

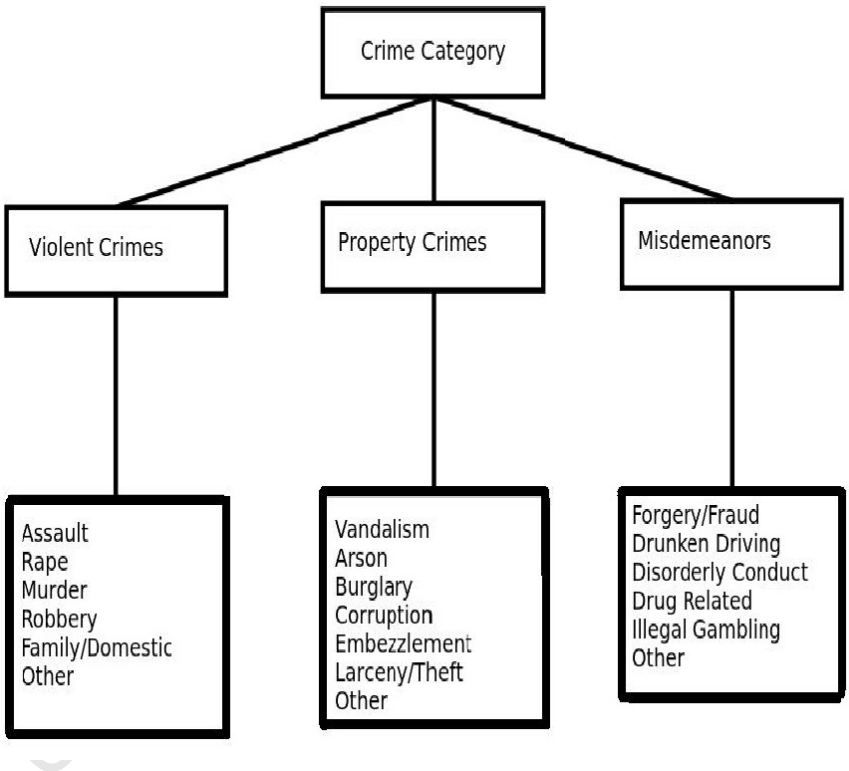



Figure 4.2: Crime Hierarchy

## 4.2 The anonymization process

In order to successfully anonymize data to conform to a specified privacy requirement, three main elements are needed:

1. Hierarchies/taxonomies for values to be generalized

2. A frequency count of tuples in a dataset

3. A suitable anonymization algorithm that can interpret elements 1 and 2 to successfully and minimally anonymize data

These three elements are described in more detail below.

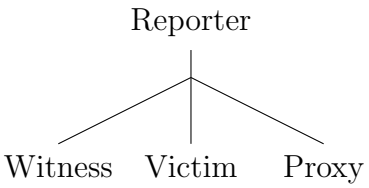


### 4.2.1 Hierarchies

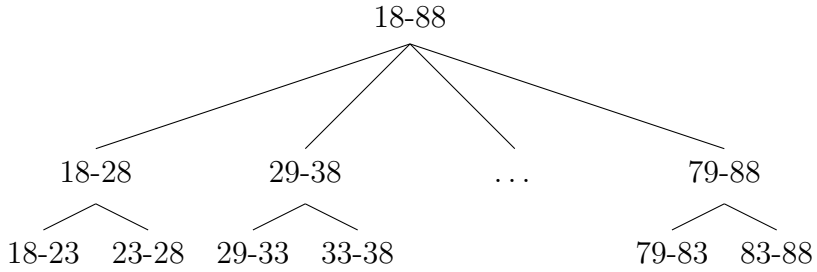
According to the FBI Uniform Crime Reporting Manual [15], crimes can be divided into three main categories with several subcategories for each category. Figure 4.2 illustrates these categories and subcategories.

Considering the FBI guidelines [15] and reports specific to data, as well as general hierarchies used in anonymizing data, the different categories for representation in our proposed framework includes the fields below.

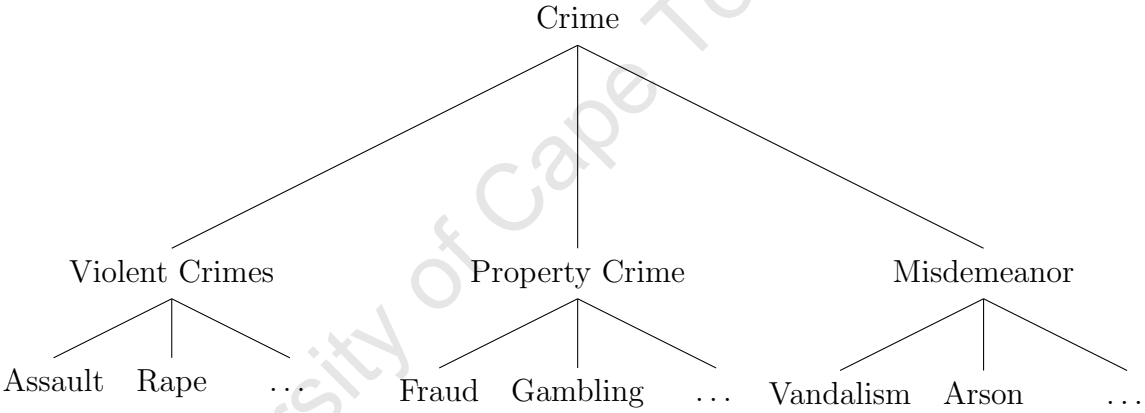
A crime reporter can be one of three types:



Age is divided into several intervals:

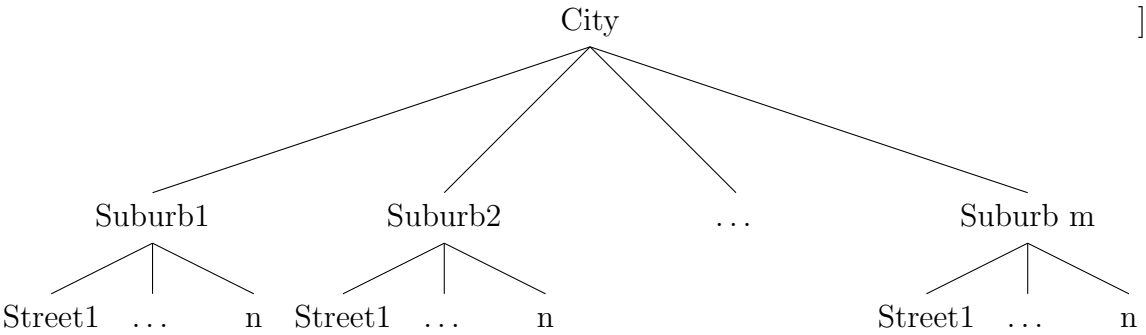



The crime category taxonomy looks as follows:






Location taxonomies are represented by the following tree:



Generalization of crime related tuples that would form part of the quasi identifier or sensitive value attributes can be divided into three distinct groups

- Numeric values
- Limited domain categorical values
- Extended domain categorical values

Values such as age would typically fall under numeric values and is part of a continuous taxonomy that can be divided up into intervals. Categorical values on the other hand can either be part of a taxonomy that takes on a limited hierarchical structure, such as a crime reporter or crime type, or it can be part of a larger domain that is more difficult to capture in a small tree structure. An example of the latter type of categorical value would be one of many street numbers that form part of one of many streets, which in turn forms part of one of many suburbs of a city.

In order to generalize numerical values such as age we need to define a simple procedure for mapping first from a specific value to a small interval (one that covers 5 numerical values for example) and thereafter to a slightly bigger interval and so on until a particular value is in its most general form.

In order to generalize a limited domain categorical value, we simply look up the value in an array and map it to its corresponding array value one level higher in the taxonomy. This continues until a value is finally suppressed when it reaches its most general form.

In order to be able to generalize an extended domain categorical value we need to ensure that when a value is in its most specific form it also contains the information for its more general forms, since the taxonomy for potentially massive domains might not always be known. As a practical example consider a particular street address: street1. In the database street1 should appear in the form City1>Suburb1>Street1. In this way for each generalization step a part of the value need only be removed. For the first generalization that would mean that street1 would become City1>Suburb1.

### 4.2.2 Frequency Counts

In order for an algorithm to determine which values to generalize, it is necessary to first estimate the amount of distinct tuples in a dataset, as well as the frequency of each tuple. Algorithm 1 outlines a basic approach to determining the frequency of tuples in a dataset. Each tuple in an arraylist created from a database query is compared to all subsequent tuples and if they are found to be equal in content, the frequency count of both is increased. In doing so, equivalence classes of tuples are created, where similar tuples will have the same frequencies for associated values. The frequency count method outlined in Algorithm 1 makes use of a nested for-

**Algorithm 1** Basic Frequency Count

```
1:  for i ← 1 to arraySize do
2:      attI_1 ← ArrayList[i_1]
3:      for j ← i + 1 to arraySize do
4:          attJ_1 ← ArrayList[j_1]
5:          if (attI_1 = attJ_1) then
6:              freq ← freq + 1
7:              array[i] ← freq
8:              array[j] ← freq
9:          end if
10:     end for
11:     freq ← 0
12: end for
```

loop declared in lines 1 & 3. Lines 2 & 4 assign each attribute to an arraylist for comparison. In line 6 arraylist values are compared and the frequency for each is increased in lines 8 & 9. A table would thus be read into an arraylist and tuples compared for similarity to determine attribute frequencies.

### 4.2.3 Anonymization

In order to enforce anonymity, we propose using an anonymization algorithm that is based on the concept of k-anonymity. As outlined in Section 4.1, there are various arguments listed in favour of k-anonymity as a means of providing sufficient privacy through indistinguishability. In k-anonymity, the value of k depends on the dataset size, the amount of privacy required and the amount of utility needed.

It is obvious that k cannot be bigger than the total dataset size. The value of k will depend on the application and the third party that data is released to.

Considering the definition of k-anonymity given in Section 2.6.1, a brief description of the existing most popular algorithms that provide k-anonymity are:

#### 4.2.3.1 Existing algorithms

**Datafly** adopts an approach whereby a frequency list is generated from a data table. This frequency list contains within it the sequences of values that are distinct in the table. Where a k-anonymity threshold is set, sequences are checked to determine whether they occur less than k times. If a sequence occurs less than k times but it accounts for more than k tuples, it needs to be generalized. This is done by selecting from these candidate tuples the ones with the highest number of distinct values and generalizing them. Sequences that occur less than k times and account for less than k tuples are automatically suppressed[1] or generalized[33, 34]. As an example consider Table 4.1. Tuple 2 would be generalized so that "Age" changes from a fixed integer to a larger interval and hence is indistinguishable from tuple 6. Tuple 10 would be suppressed to hide the Age and Gender fields.

**Incognito** requires that any and all possible generalizations of all the quasi-identifiers are generated. Each of the possible generalization domains is checked to see whether it meets the k-anonymity threshold. For each generalization that meets the threshold an information loss indicator is calculated and compared to other generalizations. The generalization with the least amount of information loss is selected [16, 21, 29, 38]. It is important to note that different information loss metrics exist and can generally be manipulated to indicate some form of attribute preference - where less information loss is tolerated in certain fields/attributes than in others.

**Mondrian** makes use of a top-down anonymization approach that first generalizes all the data in a table and then iteratively partitions data according to a

[1] Generalization entails moving values up on a defined hierarchy while suppression means to make values uniform, e.g. star (*) symbol

particular attribute until there is no feasible partition available[20, 29, 30, 38]. The data in Table 4.1 would be generalized and then partitioned according to "Gender". After partitioning according to "Gender", "Age" would be partitioned. This will continue until a table violated a k-anonymity threshold. It is important to note that Mondrian generates its own generalization hierarchy for attributes. The generalization hierarchy is based on converting values to integers and dividing values into intervals. The intervals size is then increased to represent a more general value in a hierarchy. A detailed description of this process can be found in [20].

Table 4.1: Original Table

| Age | Gender | Crime |
|---|---|---|
| 33 | Male | Arson |
| 25 | Male | Rape |
| 65 | Female | Drug related |
| 39 | Male | Vandalism |
| 39 | Male | Vandalism |
| 53 | Male | Rape |
| 47 | Female | Drunken Driving |
| 21 | Female | Burglary |
| 19 | Female | Drug related |
| 30 | Female | Theft |



#### 4.2.3.2 Crimemod

Different algorithms have different approaches to segment data for anonymization. However, existing approaches such as Mondrian rely on internal hierarchies. As outlined in Section 4.2.1, clear external hierarchies do exist for crime data. In addition, other existing implementations of generalizations algorithms, such as Incognito and Datafly, rely on explicit conversion of values to numeric representations for generalization. In crime data, it is not always clear how to assign numeric values to crime- or location categories. The Crimemod algorithm accepts values in their categorical form.

Furthermore, any value could be generalized under existing approaches because a record did not meet a certain frequency threshold. Existing approaches rely on

theoretical applications and do not often consider real world problems. However, it is possible that generalizing one value in a tuple will result in it being moved to a different equivalence class and consequently the tuple will have the required frequency, but a lot of information will have been lost. It is intuitive in crime data that less information will be lost by slightly increasing interval sizes for numeric values such as age, whereas a significant amount of information will be lost by generalizing a location value which only has a 3 level taxonomy height. The latter (categorical) values should only be anonymized where no candidate numerical value exists. Crimemod ensures that information loss on categorical values is weighed heavier than numerical values.

A basic approach used in our adaptation to achieve this goal, termed "crimemod", is presented in Algorithm 2.

**Algorithm 2** Crimemod

**Require:** Frequency Count Method; Generalization Hierarchies
**Ensure:** Equivalence classes $\geq k$

**function** FREQ COUNT($Array$) ▷ Sends values to Alg1
    $FreqCountMethod \leftarrow Array(Att_1, Att_2, Att_n)$
    **return** $Frequency$
**end function**

**while** $Frequency < k$ **do**
    **for** $i \leftarrow 1$ to $arraySize$ **do**
        $Element \leftarrow FreqCount[i]$ ▷ Obtains frequency from Function
        **if** $(Element < k)$ **then**
            **if** $Att_j(i).isInteger$ **then** ▷ $Att_j$: denotes any attribute
                $Att_j(i) \leftarrow (NumericGeneralization - 1)$
            **else**
                $Att_1(i) \leftarrow (GeneralizationHierarchy_1Val - 1)$
                $Att_2(i) \leftarrow (GeneralizationHierarchy_2Val - 1)$
                ...
                $Att_n(i) \leftarrow (GeneralizationHierarchy_nVal - 1)$
            **end if**
        **end if**
    **end for**
**end while**

The frequency counts for each element (line 2), obtained from Algorithm 1 is compared to the minimum privacy requirement $k$ (line 4). First numeric values are considered in lines 5 & 6 and subsequently the different categorical attributes are considered in lines 8 through 11. Values that are smaller than the threshold are moved one step up on the generalization taxonomy. This is done for all values smaller than $k$, before the frequency is recomputed in lines 15 to 17. The process continues until all tuples meet the privacy requirement, or all values are in their most general form (line 1 & 19).

Table 4.2: Anonymized Table

| Age | Gender | Crime |
|---|---|---|
| * | * | * |
| * | Male | Rape |
| * | Female | Drug related |
| 39 | Male | Vandalism |
| 39 | Male | Vandalism |
| * | Male | Rape |
| * | * | * |
| * | * | * |
| * | Female | Drug related |
| * | * | * |



As an example to outline the anonymization process, consider Table 4.1. Anonymized data will require equivalence classes of size $k$. For all integer values in equivalence classes smaller than $k$, a numeric generalization will take place. In the first iteration of generalization this means converting values to a small interval range. Further iterations, as determined by the while loop in line 1 & 19 of Algorithm 2, would increase the interval size until the interval is spread across all values. For all non integer values, the hierarchy index for a particular value is looked up and the value is generalized to one level higher in the hierarchy. This will continue until the privacy threshold is met or the table is fully generalized. It is intuitive that the $k$ threshold should be smaller than the total table size. Setting k = 2[1] with the table in Table 4.1 will result in an anonymized set as indicated in Table 4.2.

[1] k = 2 is used for the purpose of concept explanation

#### 4.2.3.3 L-Diversity Extension

As a extension of our application, consider $\ell$-diversity: suppose there is a sensitive value field pertaining to a person making the crime report who could fall into the category victim, witness or proxy. Victims would typically be those who suffered a crime of some sort, a witness would be one who saw a crime happening and a proxy someone reporting a crime on behalf of a witness or a victim. If we set $\ell = 2$ that would mean that the most frequent sensitive value should occur $1/\ell$ times in each equivalence class. For arguments sake let us suppose this is witnesses. That would mean that each equivalence class of size k should contain at least 50% tuples with sensitive value equal to witness. This ensures diversity in the sensitive values. For theoretical purposes, sensitive value generalization was included to show how crime data can be manipulated to ensure $\ell$-diversity. In this extension age and crime categories can be considered quasi identifiers with age a numerical field and crime representing the three categories representing violent crimes, property crimes and misdemeanours as well as representative values for subcategories.

**Algorithm 3** L-Diversity Extension

1: **for** $i \leftarrow 1$ to $arraySize$ **do**
2: $freqCount1 \leftarrow FreqCount[i]$
3: $freqCount2 \leftarrow FreqCountDiversity[i]$
4: $DividedCount \leftarrow freqCount2/freqCount1$
5: **if** $(DividedCount > \ell - Threshold)$ **then**
6: **if** $Att_j(i).isInteger$ **then**
7: $Att_j(i) \leftarrow (NumericGeneralization - 1)$
8: **else**
9: $Att_1(i) \leftarrow (GeneralizationHierarchy_1 Val - 1)$
10: $Att_2(i) \leftarrow (GeneralizationHierarchy_2 Val - 1)$
11: ...
12: $Att_n(i) \leftarrow (GeneralizationHierarchy_n Val - 1)$
13: $Att_{sens}(i) \leftarrow (GeneralizationHierarchy_{sens} Val - 1)$
14: **end if**
15: **end if**
16: **end for**

By expanding on Algorithms 2 & 4 we can extend basic k-anonymity to adhere to $\ell$-diversity. This requires that in Algorithm 4 the frequency count is updated to let $attJ_{k+1}$ and $attI_{k+1}$ denote a sensitive value. The updated method is called FreqCountDiversity. Generalization is similar to Algorithm 2, with the the additional diversity ensuring procedure outlined in Algorithm 3 included before the while loop will end. The main difference between Algorithm 2 and Algorithm 3 is illustrated in lines 3 & 4 of Algorithm 3. The additional privacy requirement is introduced by calculating not only the frequency of an equivalence class, but also the frequency of a sensitive attribute in that equivalence class so that

$$Frequency = \frac{SV_c}{EC_c}$$

with $SV_c$ denoting the sensitive value count and $EC_c$ denoting the entire equivalence class count. In addition to further generalization of values, the sensitive value could also be generalized in order to let equivalence classes be $\ell$-diverse.

Table 4.3 illustrates a dataset where records 1 to 4 form an equivalence class with the most frequent value occurring more than $\frac{1}{\ell}$ with $\ell = 2$. This class needs to be generalized. On the other hand, records 5 to 7 form an equivalence class where the most frequent value occurs less than $\frac{1}{\ell}$ with $\ell = 2$. This equivalence class need not be generalized.

Table 4.3: Equivalence Classes Illustration

| Age | Crime | Reporter |
|---|---|---|
| 23 | Robbery | Victim |
| 23 | Robbery | Victim |
| 23 | Robbery | Victim |
| 23 | Robbery | Proxy |
| 39 | Drug related | Witness |
| 39 | Drug related | Victim |
| 39 | Drug related | Proxy |
| 25 | Murder | Witness |

#### 4.2.3.4 Various Attributes

Lastly, Crimemod is designed to handle various sensitive and identifier attributes. Location, age and gender could all be considered quasi-identifier attributes. In the same way, it is conceivable that both the reporter type as well as the crime could be considered sensitive attributes. Algorithm 4, mentioned in Section 4.2.3.3, illustrates how the frequency count of various sensitive attributes is generated. In Algorithm 4 $Attribute[I_1]$ refers to the first attribute of a record. $Attribute[I_2]$ is the second and $Attribute[I_k]$ is the sensitive attribute. On a practical level that would mean $Attribute[I_1]$ represents the age value for the first record, $Attribute[I_2]$ represents the gender for the first attribute and $Attribute[I_k]$ represents the crime. In Algorithm 4 the frequency count for tuples with all attributes equal to each other is computed.

**Algorithm 4** Frequency Count for Various Attributes

**for** $i \leftarrow 1$ to $arraySize$ **do**
    $attI_1 \leftarrow ArrayList[i_1]$
    $attI_2 \leftarrow ArrayList[i_2]$
    ...
    $attI_k \leftarrow ArrayList[i_k]$ ▷ ...: any number of attributes
    **for** $j \leftarrow i + 1$ to $arraySize$ **do**
        $attJ_1 \leftarrow ArrayList[j_1]$
        $attJ_2 \leftarrow ArrayList[j_2]$
        ...
        $attJ_k \leftarrow ArrayList[i_k]$
        **if** $(attI_1 = attJ_1) \wedge (attI_2 = attJ_2) \wedge \cdots \wedge (attI_k = attJ_k)$ **then**
            $freq \leftarrow freq + 1$
            $array[i] \leftarrow freq$
        **end if**
    **end for**
    $freq \leftarrow 0$
**end for**

# Chapter 5

# Results and Evaluation



## 5.1 Experimental Evaluation

The success of the framework is determined by the ability of the framework to record useful and valuable data; anonymize collected data; and to publish such data while preserving privacy. The framework will be considered to work properly if data can be retrieved from a mobile device, stored in a table and an algorithm can ensure that data is properly anonymized. In this regard, testing experiments were set up to measure performance, usability and security. Performance is measured in terms of turnover time for the running of the algorithm to ensure that a table is anonymous. For usability, information loss and classification accuracy serves as gauging measures. Security is measured by determining if the minimum security threshold is reached for all tuples in a data table.

### 5.1.1 System specifications

Our proposed framework was implemented on an Intel Core i7-2600 3.40GHz machine with 8GB physical memory. The operating system used was Ubuntu 12.9 and we used the University of Texas Dallas toolbox[1] in Java with NetBeans version 7.2.

[1]The toolbox can be found at: http://cs.utdallas.edu/dspl/cgi-bin/toolbox/

### 5.1.2 Comparison metrics for evaluation of algorithms

Finding a common basis for evaluating different different notions of privacy and the algorithms that guarantee them is still a problem that needs to be addressed in order to make a fair assessment of different approaches. Categorical data that exists in the University California Irvine (UCI) machine learning repository [6] has been used for anonymization in the past [36, 37], but numerical data is yet to be evaluated. It is also not always clear how much information an adversary has at her disposal and how much she would need to defeat a particular privacy notion. In computer security we assume that an adversary has infinite resources and to this extent certain metrics do exist to measure different data publication approaches, based on usability and privacy protection This section highlights the different metrics used to evaluate and compare privacy preserving algorithms.



**Computational complexity** Computational complexity is one way of comparing different anonymization techniques. The complexity of the optimal distribution of k-anonymous equivalence classes is an NP-hard problem [13, 37]. The practical implication is that an algorithm's computational time will take exponentially longer as datasets increase. Within this constraint, a scalable solution should still be pursued: if an algorithm takes hours to produce a result on larger sets, then such an algorithm is not a scalable solution for a scenario where continuous access to data is needed, as would be the case with crime data. Algorithms that provide anonymity by means of generalization share the following computational complexity properties:

If an entire dataset table is $|T|$ and the hierarchical height of a taxonomy of an identifier attribute is $H = \prod_{i=1}^{d} h_i$, with $h_i$ denoting the taxonomy height of an identifier attribute $A_i$. Then we can say that the time complexity for the an algorithm is defined as the operational cost (O) so that:

$$TimeComplexity = O((H) \cdot |T^n|$$

with $n$ denoting the amount of different sets of generalization possibilities [37].

**Computational complexity** Execution time is used as an extension of complexity. It is used as both a good measure of performance as well as a good measure of practical algorithm scalability, especially on increasingly larger datasets. While considering computational complexity is important, it is also important to measure actual application execution time, seeing as execution time could vary at implementation level based on different privacy requirements. With algorithms such as Incognito that were designed for different privacy specifications, brute force searches would have to be carried out starting from $\ell = 2$, whereas others do not need to do a brute force search. However, efficiency in isolation means very little, so data needs to conform to privacy requirements while retaining utility.

**Privacy protection:** Privacy protection is another necessary comparison measure. For example, in [36], $(\tau,\ell)$-diversity is used to evaluate different algorithms. Considering that no single agreed upon measure exists for gauging privacy, this is understandable if not fair. The problem with selecting one privacy measure is that different algorithms might not be ideally suited to provide very particular privacy protection, while other algorithms might have been designed for that exact privacy measure.

While it is hoped that an algorithm will provide adequate privacy, where adequate privacy can be understood to mean meeting specified privacy threshold such as $(\tau,\ell)$-diversity, said algorithm should not deliver excessive protection. Excessive protection is the measured distance of sensitive attribute distribution from the initial to the published dataset.



**Utility:** A third metric to compare algorithms is the utility of published datasets. This is measured with count queries, classification accuracy determining measures and information retained.

**Count queries:** Count queries are complex queries of which the answers are evaluated for meaningfulness. Count queries give an indication of the overlaps between anonymized and original datasets in terms of taxonomies.

**Classification accuracy:** Classification accuracy is the process of shaping tree rules based on the original data and thereafter on the anonymous data. Learning rules/trees from the anonymized data requires removing generalized values and replacing them with random particular values from their taxonomies. The accuracy is measured by dividing the accuracy of decision trees of anonymous data with the accuracy of decision trees learned from the original data so that: $accuracy = \frac{ca_a}{ca_o}$, with $ca_a$ denoting anonymous tree accuracy and $ca_o$ denoting original tree accuracy. Instance Based(IB) classification is frequently used to measure accuracy of datasets after anonymization, which is an indication of how useful datasets are after anonymization, as opposed to before anonymization. Instance Based classification is a classification accuracy method where automatic classification is done based on nearest neighbours. In the case of data anonymization the nearest neighbour is most often set to 1, otherwise known as IB1 classification. To better understand the process of determining classification accuracy in privacy preserving data publication, consider Figure 5.1. Each value that is a more general form than the leaf node will be replaced by a randomly selected value from the leaf nodes. In Figure 5.1 the three different values each have a 33% chance of being the correct selection (that corresponds with the original data). It is easy to see that the more general a value is and the more sub- and leaf nodes it has, the higher the likelihood that an incorrect replacement will be selected.



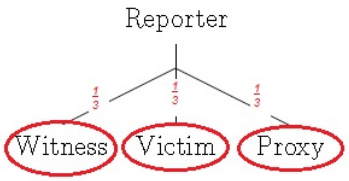



Figure 5.1: Taxonomy Value Likelihood

**Information Loss:** Information loss is usually measured in terms of how many leaves a certain generalized value has [18, 37]. Informational content is considered to be the amount of leaf nodes that any value has. Intuitively, the more leaf nodes a value has, the more information was lost when it was decided

to move up to this particular value in the hierarchy. Therefore:

$$\text{Information}(v) = \frac{1}{|\text{leaves}(v)|}$$

In the same way, the information ($I$) for a whole tuple $t$(that consists of several values) is:

$$\text{Information}(t) = \sum\nolimits_{\text{A}\in A} I(t[\text{A}]),$$

where $A$ denotes the set of possible values and $t[\text{A}]$ is the attribute value of a particular tuple. If we wish to calculate the information detained by an entire table, we merely add all the different tuples' informational content. Both *I(G)* and *I(O)* are calculated in the same fashion so that:

$$\text{Information}(G) = \sum\nolimits_{t\in G} I(t),$$

$$\text{Information}(O) = \sum\nolimits_{t\in O} I(t),$$

Where *Information(G)* is the information in the generalized set and *Information(O)* is the information in the original set. The utility of a dataset after anonymization is the amount of information left over, compared to the original dataset. This is calculated by dividing *Information(G)* by *Information(O)* so that $Utility(G) = \frac{I(G)}{I(O)}$. It makes sense that the original dataset has all values at their most specific level and the anonymized set might have more general values. Weights are generally not assigned to sensitive and identifier attributes in theoretical approaches like Datafly, Mondrian and Incognito, because it is argued that it unclear how these should be assigned and is more useful for modelling in real world applications. This is a particularly important point for those wishing to anonymize crime, medical, insurance or other real world data.



### 5.1.3 Pilot Study Experiments

To construct the anonymization module, we implemented three popular anonymization algorithms mentioned in Section 4.2.3.1: Datafly, Incognito and Mondrian to compare results and decide on a suitable approach to anonymize crime data.

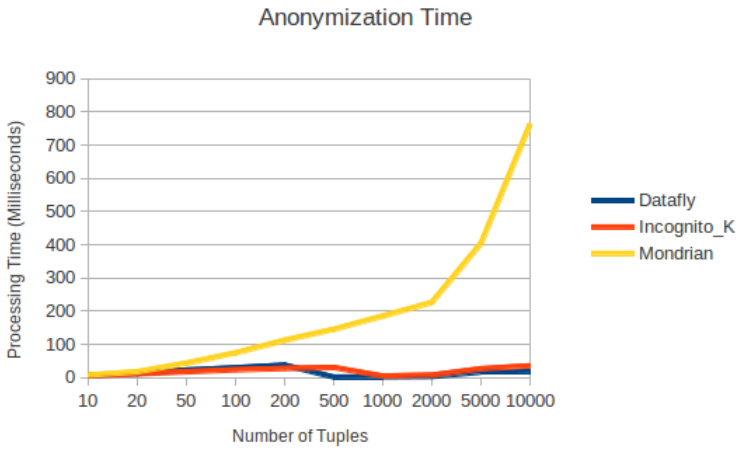



Figure 5.2: Processing Time

#### 5.1.3.1 Dataset

For the pilot study the "Adult" dataset from the UCI Machine Learning repository [6] was used. Salary was selected as the sensitive attribute and age as well as gender as the quasi-identifier attributes.

#### 5.1.3.2 Performance

As mentioned in Section 5.1.2, performance is an indication of data accessibility, especially where data is needed for services such as crime area avoidance applications that typically rely on continuous access to data.

There is a markable difference between algorithms in processing times as the amount of tuples increase. As is evident from Figure 5.2, it takes significantly longer for the Mondrian algorithm to ensure anonymity. The Mondrian results indicate that it could become a liability for larger datasets and is thus not a feasible option.

Datafly decreases its processing time as tuples increase and performs marginally better than Incognito with k as privacy threshold. The decreases in processing time can be attributed to less generalizations that are necessary as more tuples

are available.

As an example, consider there are 500 individuals instead of 200 in a dataset: this would mean that there is a higher likelihood that individuals would share an age (e.g. 46) and therefore are indistinguishable from each other. If values are indeed the same and consequently meet the required security threshold, it is not necessary to generalize these values to a higher range. This saves processing time.

Table 5.1: Proccessing times (Milliseconds)

| | Datafly | Incognito | Mondrian |
|---|---|---|---|
| 10 | 6 | 5 | 7 |
| 20 | 12 | 11 | 17 |
| 50 | 22 | 17 | 43 |
| 100 | 29 | 23 | 74 |
| 200 | 37 | 27 | 112 |
| 500 | 1 | 30 | 145 |



#### 5.1.3.3 Classification Accuracy

Taking into consideration the poor time performance for Mondrian, another pilot study experiment was conducted where both datasets were anonymized with the k-threshold increasing by intervals of 5 per iteration, but only with the Datafly and Incognito algorithms. The anonymized sets were compared to the original sets using Instance Based(IB) classification.

The UTD toolbox allows a user to use IB1 classification with a Weka implementation and outputs the results alongside the anonymized data. To test the classification accuracy of various algorithms, a batch processing program was set up in Java which automatically increases the K-privacy threshold by 5, starting at 0 and increasing until 200. Equivalence classes thus needs to be equal to whatever K is set to. On the "Adult" dataset (which comes as the default set with the UTD toolbox) classification accuracy of around 89% was achieved.

The classification accuracy of the Datafly and Incognito algorithms for the "Adult" dataset is presented in Figure 5.3. Note that initially there was fluctuation in both

algorithms for different values of k, but from k > 180 the accuracy stabilized. Results indicate that this stays the same for bigger values of k. Once again Datafly performed marginally better than Incognito on the "Adult" dataset.

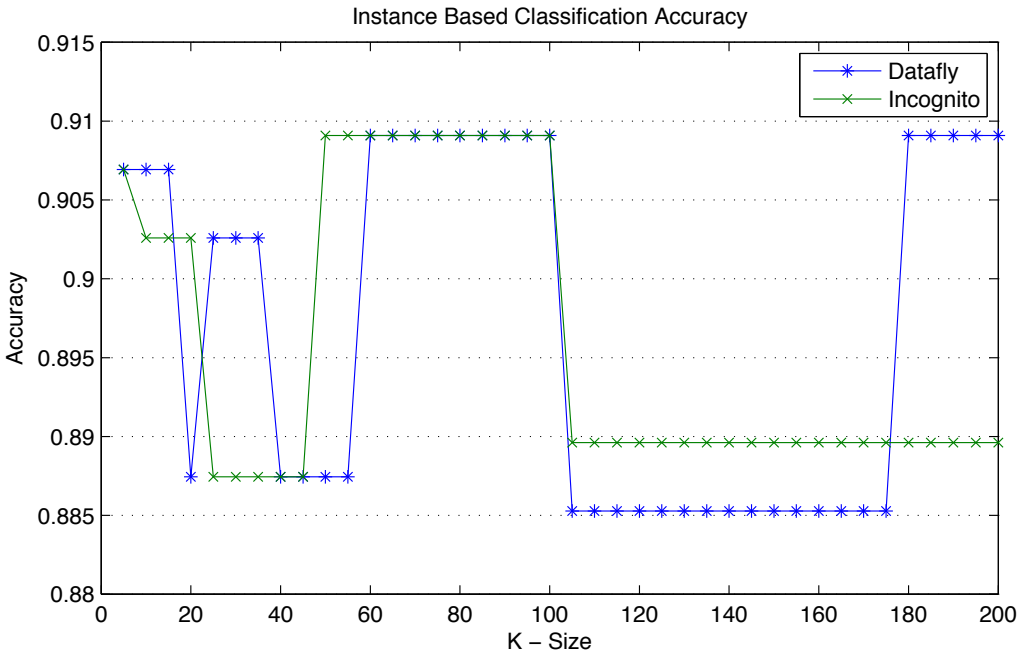



Figure 5.3: Adult Dataset Classification Accuracy



#### 5.1.3.4 Results Indication

Based on the fact that the Datafly algorithm performed the best on different metrics, this was selected as the optimal anonymization algorithm for the framework. Datafly was implemented in Java with several extensions applicable to anonymizing crime data and termed "crimemod". The simplified version of this extended implementation is outlined in Algorithm 2. This data specific anonymization method has been made publicly available at https://sourceforge.net/p/crimeanon/.

## 5.2 Experimental Evaluation - Crimemod

The crimemod algorithm discussed in Section 4 was implemented and tested alongside both the Incognito and Datafly algorithms, also in Java using Netbeans 7.2. Crimemod was programmed making use of the algorithms for frequency count and anonymization described in Section 4.2.2 and Section 4.2.3.2 in Chapter 4.

### 5.2.1 Dataset

The three algorithms were tested on a reported crime dataset. The dataset is a self-generated set to replicate possible crime reports from a mobile device, titled "Reports".

#### 5.2.1.1 Dataset Generation

In order to construct a dataset representative of actual crimes, the prototype mobile device reporting tool outlined in Section 3.2 was used to gather synthesized crime reports from a selected group of student participants. The data was used to generate a similar set of values with more tuples in a table. The fields for crime category, age, location and optional fields for name and surname were included in the dataset. In order to anonymize data, identifier attributes, including name and surname were removed. Remaining fields were used as candidates for anonymization. Seeing as actual crime data has not yet been anonymized in similar approaches (to the best of our knowledge), it is impossible to norm data against real world crime report data, yet is important to consider that a functioning anonymization module will ensure that data is adequately anonymized regardless of whether it is synthesized or original.



### 5.2.2 Results Discussion

Two relevant non sensitive attributes were specified (age and gender) as well as one sensitive attribute. In one experiment, the "Reports" dataset was divided into subsets of different sizes, starting at 10 tuples and stopping at 10 000 tuples[1]. The

[1]Tuples represent individuals that made reports in the data

intuition behind this decision being that a crime reporting service will start out with a small number of reports, but this can and likely will grow exponentially as the system is implemented and used.

#### 5.2.2.1 Performance Evaluation

For the purpose of testing algorithms on datasets that start with a tiny amount of tuples, the threshold was set as k=2. This is because setting k > 2 in a dataset of only 10 tuples requires that any tuple must be indistinguishable from more than 2 values and will almost surely result in all values being changed to their most general form. For measuring processing time, attribute age was selected as quasi-identifier and the crime category as the sensitive value. Location, with a defined hierarchy of more generalized locations (or any other non-sensitive attribute for that matter), can also be used as a quasi-identifier to be generalized in order to ensure anonymity. However, using age as quasi-identifier results in a published table that contains generalized values for age, e.g. values change from 46 to being in the age range [45-49].

It is clear from Figure 5.4 that the differences between the Crimemod and Incognito algorithms are marginal.

#### 5.2.2.2 Classification Accuracy

The same classification accuracy metric described in Section 5.1.2 and used in the pilot study to determine classification accuracy was used in the experimental evaluation of crimemod alongside Incognito and Datafly. In order to evaluate the classification accuracy of Crimemod, manually generating and transforming a result set for each size of $k$ was required. This was done by parsing the data table from the anonymization process into a suitable Weka readable file with appropriate headers (see Appendix B for an example). The results obtained from these expiements were compared to IB1 classification of Incognito and Datafly. These results can be seen in Figure 5.5. The impressive result in Figure 5.3 of the pilot study in Section 5.1.3.3 is because the sensitive value in the adult dataset is the salary of each individual represented in the set and that is only either more than 50 000 or less than 50 000. So classification only has to occur between two values. In the

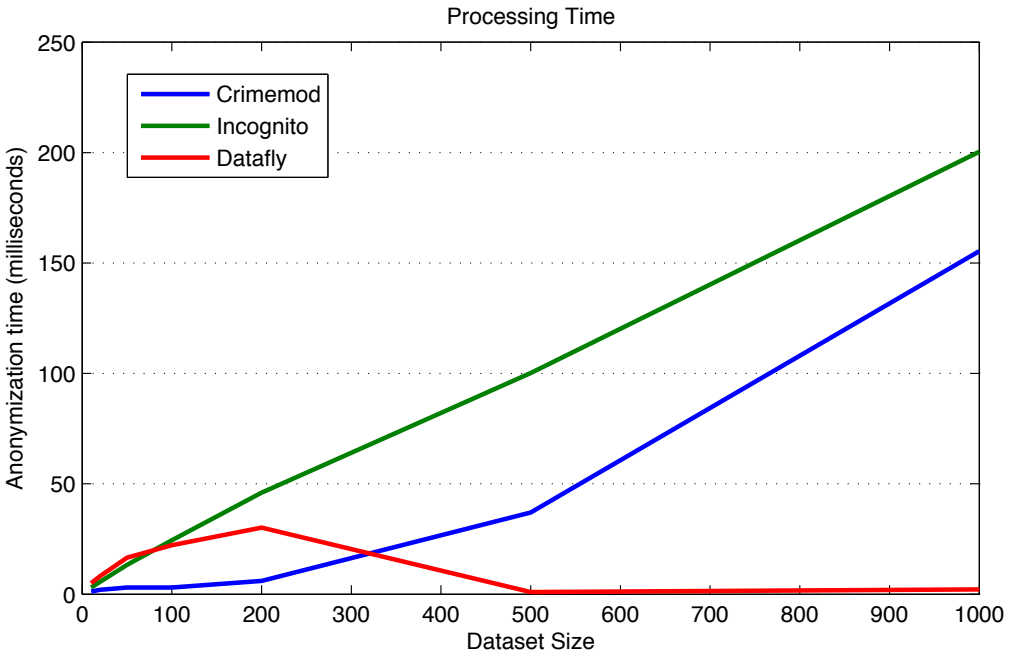



Figure 5.4: Anonymization Times

same way if the sensitive values input of the "Record" dataset (the self generated set with particular crime related data) is only one of two crime categories, then the accuracy is equal to the same percentage.

When considering various sensitive values, as would be the case with real crime reports that has various crime categories and sub categories, the accuracy of Datafly and Incognito significantly decreases. It only achieves an accuracy of between 4-7% if measured with IB1 classification. This becomes clear in Figure 5.5.

Using the crimemod algorithm specifically suited to multiple sensitive values and crime data in particular with k = 2, the accuracy is better than Incognito and Mondrian. This is especially true for smaller values of k, with accuracy as high as out to around 38%. The result is significantly better for specifically crime data on this system.

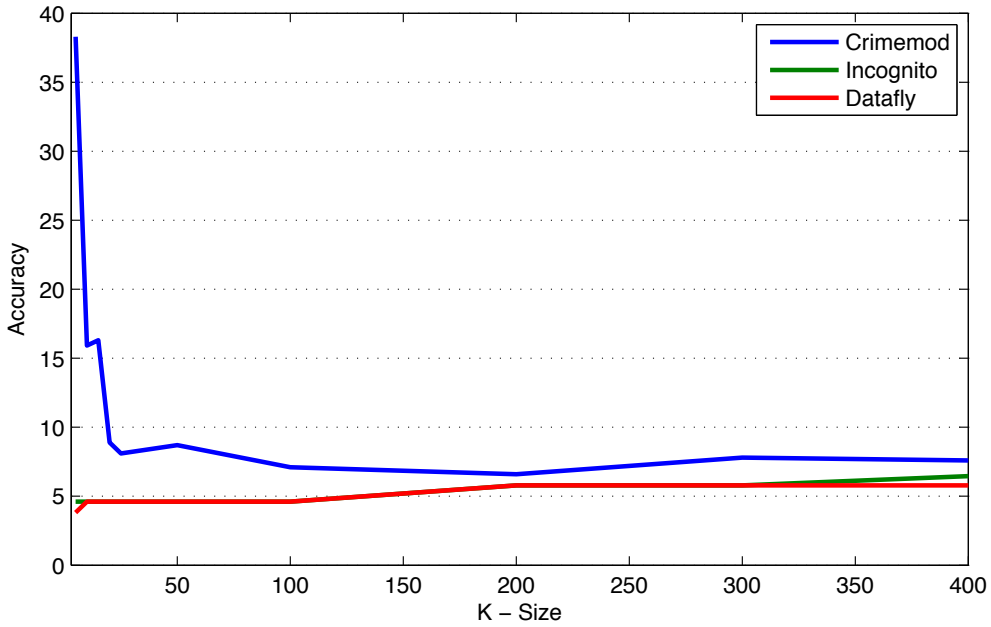



Figure 5.5: Multiple Sensitive Records Accuracy

#### 5.2.2.3 Information Loss

To measure information loss we made use of the Information Loss metric described in Section 2.6.3. To implement the metric we copied the anonymized data to a spreadsheet and converted intervals to integers. This was done so that the smaller of the two values could be subtracted from the larger. The result was divided by the total interval size. The average for all tuples was divided by the number of tuples to give an information loss count. It is clear from Figure 5.6 and Figure 5.7 that Crimemod significantly outperforms Incognito and Datafly. Because of the specific hierarchies defined and the generalization of only attributes that guarantee the least information loss from equivalence classes below a certain threshold, anonymized data is produced by Crimemod that provides the maximum utility under privacy constraints. This becomes clear in Figures 5.6 & 5.7

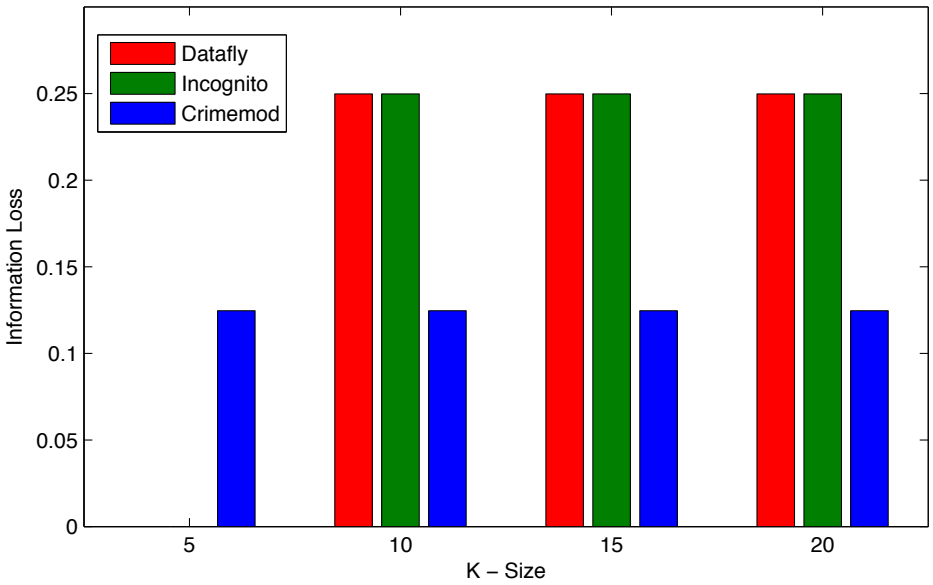



Figure 5.6: Infoloss (K = 5-25)

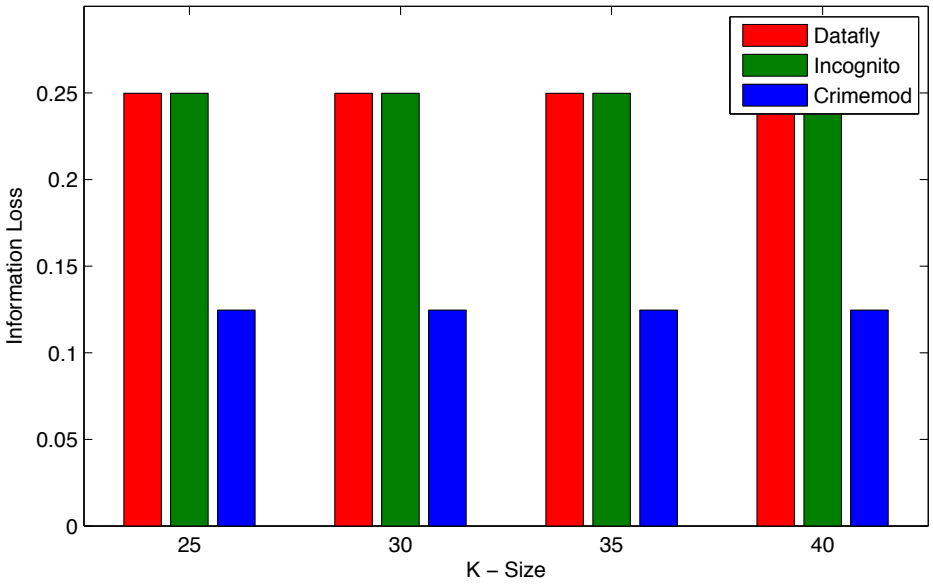



Figure 5.7: Infoloss (K = 25-50)

# Chapter 6

# Conclusions and Future Work

This thesis presented an architectural overview of a crime reporting framework and an implementation of the mentioned system. The main research questions were addressed: The first related to what services can be provided as a crime reporting platform. The second was whether anonymity can solve the privacy protection problem for participants and the third was how to handle anonymizing crime reports.



## 6.1 Summary

To answer the first of the three questions, an integrated framework provides a new crime reporting system, whereby users can safely and conveniently report crimes they might not otherwise have. To this extent mobile phones were used in the framework and the utilization of a simple mobile interface assumed.

The second and third questions relate to anonymity in data publishing. There are various strategies and approaches that can be employed when constructing a privacy preserving crime data distribution service. For example, the decision can be based on the findings of previous studies and applications, with the assumption that the findings are transferable. Alternatively a careful analysis and comparison of all anonymity providing approaches can be used to determine which of them is the most suitable for the anoymization task at hand. Finally, a hybrid approach can be used where applicable existing techniques are evaluated and improved on

for the specific crime data anonymization task.

There are advantages and disadvantages to all three of these approaches: The first approach comes at the cost of making an assumption of transferable findings, especially if approaches like medical data anonymization is adopted bona fide without considering neccessary changes to such approaches in order to make it applicable to crime data. Using this approach, an optimal solution is less likely to be found. With the second approach the most suitable technique for the anonymization task can be identified, but it comes at the cost of considering a large number of techniques. Not all techniques may be applicable and some have been found to not provide the needed privacy requirements such as k-anonymity. The final approach, used in this thesis, was to consider the most applicable algorithms, measure-, compare- and adapt these algorithms to the hierarchical value crime reporting scenario. The advantage of this approach is that suitable algorithms could be evaluated in depth with a particular crime data application in mind.



## 6.2 Contribution

USSD is suggested as a novel approach of collecting data from participants in the developing world and based on our prototype implementation seems to be a scalable solution. The USSD interface was developed and tested on a select group of students. In addition, with *crimemod* as the anonymization algorithm in the framework, data can be recorded and subsequently published for a range of applications and uses. An optimal algorithm ensures that long delays are not experienced when requesting a report from the system. As reports and subsequently tuples in the table increase, the framework can render a useful service that functions effectively and ensures that participants are assured of their anonymity.

## 6.3 Results

A set of experimental results demonstrating classification accuracy and processing time justifies the choice of Datafly as the candidate anonymization algorithm for

extension to protect participants making reports. A data specific implemented approach termed "crimemod" was introduced and the associated process explained. Crimemod was compared to two of the most frequently used algorithms to ensure anonymity: Incognito and Datafly. Results indicate that Crimemod is capable of dealing with crime related data in a way that minimizes information loss, displaying as little 12.4% loss of information on average for different sizes of $k$. Crimemod displays classification accuracy that is comparable to existing sophisticated algorithms and can anonymize datasets within an acceptable time frame.

## 6.4 Future Work

Future work on reporting crimes in the developing world can build either on the framework in general, or the anonymization module in particular.

### 6.4.1 Framework expansion

On the side of collecting crime reports, it is necessary to acknowledge that the interactive, simple and understandable interface associated with USSD led to the assumption that usability will not be a barrier to participation. Optimized user interfaces and human interaction with these interfaces, particularly for crime reporting, is an interesting avenue of research for those interested in building on the existing framework. To this extent, usability analysis metrics could be implemented. Related to this is whether participants, after all possible measures have been taken (from anonymization to friendly interfaces), would choose to participate. It is a complicated question which spans the fields ranging from system marketing to psychology. However, it is an interesting and important consideration in all privacy preserving related work.

Additionally, regarding the authentication mechanism, data access currently relies on a password protected access control mechanism that sets k=0 for a trusted official. An interesting possible area of expansion is to research how a trusted official can get authenticated access via two or more authentication mechanisms

to unchanged crime data. This needs to be done in a manner that ensures the anonymity of participants, perhaps encryption of data is a possible solution. As mentioned in Section 3.3.3, trusted officials should be able to get access, but even trusted officials with unlimited access could put participants off from making crime reports. It is conceivable that access should only be granted on a need to know basis for qualified officials (such as police officers) and access rights granted and revoked on an individual, case-by-case basis. In this way, users are protected by both the anonymization module and the privacy policies around data release.

Lastly, with regards to the framework, it is perhaps important to consider the scaling of the project. Although the framework was developed in a University setting tailored for a finite amount of students, it will be necessary to consider large databases and associated transaction costs if a centralized system was set up to utilize the framework on a national or even continental level. Theoretically though, MySQL is set up to handle large datasets and the more complicated question would not necessarily be anonymization but rather authenticated and trusted access.

### 6.4.2 Crimemod expansion

Although crimemod has the ability to generalize several quasi identifier attributes, future work on this framework will entail expanding the anonymization technique to various sensitive values, for example including location of a crime. It is not immediately clear how t-closeness and other extensions of anonymity requirements can be introduced as additional privacy restrictions, but this exists as an avenue of further research. Finally, further research can also expand on exactly how much privacy (size of k) is sufficient, particularly for crime data.

# Appendix A

Table 1: Processing time for Anonymization Methods

| Datafly | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset Size** | **Anon1** | **Anon2** | **Anon3** | **Anon4** | **Anon5** | **Average** |
| **10** | 25 | 35 | 20 | 25 | 20 | 25 |
| **20** | 35 | 45 | 45 | 35 | 45 | 41 |
| **50** | 75 | 80 | 85 | 75 | 100 | 83 |
| **100** | 145 | 115 | 95 | 95 | 105 | 111 |
| **200** | 150 | 165 | 165 | 145 | 130 | 151 |
| **500** | 5 | 5 | 5 | 5 | 5 | 5 |
| **1000** | 10 | 10 | 10 | 10 | 15 | 11 |

| Mondrian | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset Size** | **Anon1** | **Anon2** | **Anon3** | **Anon4** | **Anon5** | **Average** |
| **10** | 21 | 25 | 25 | 25 | 25 | 24.2 |
| **20** | 50 | 62 | 69 | 47 | 71 | 59.8 |
| **50** | 115 | 110 | 120 | 160 | 156 | 132.2 |
| **100** | 251 | 259 | 273 | 200 | 197 | 236 |
| **200** | 382 | 282 | 307 | 292 | 292 | 311 |
| **500** | 497 | 539 | 484 | 452 | 462 | 486.8 |
| **1000** | 667 | 622 | 647 | 615 | 642 | 638.6 |

| Incognito_K | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset Size** | **Anon1** | **Anon2** | **Anon3** | **Anon4** | **Anon5** | **Average** |
| **10** | 30 | 35 | 34 | 35 | 40 | 34.8 |
| **20** | 30 | 30 | 31 | 35 | 25 | 30.2 |
| **50** | 115 | 90 | 102 | 70 | 80 | 91.4 |
| **100** | 105 | 90 | 93 | 140 | 85 | 102.6 |
| **200** | 160 | 130 | 158 | 125 | 200 | 154.6 |
| **500** | 10 | 10 | 12 | 10 | 15 | 11.4 |
| **1000** | 30 | 25 | 27 | 25 | 25 | 26.4 |

| Crimeproc | | | | | | |
|---|---|---|---|---|---|---|
| **Dataset Size** | **Anon1** | **Anon2** | **Anon3** | **Anon4** | **Anon5** | **Average** |
| **10** | 6 | 0 | 0 | 0 | 0 | 1.2 |
| **20** | 5 | 5 | 0 | 0 | 0 | 2 |
| **50** | 5 | 0 | 0 | 5 | 5 | 3 |
| **100** | 10 | 0 | 0 | 0 | 5 | 3 |
| **200** | 10 | 5 | 5 | 5 | 5 | 6 |
| **500** | 35 | 30 | 40 | 40 | 40 | 37 |
| **1000** | 147 | 147 | 159 | 147 | 177 | 155.4 |

# Appendix B

The University of Texas Dallas Toolbox comes with an xml configuration file for different privacy settings and adjusting privacy requirements. Listing 1 illustrates an example configuration file. In this configuration, Incognito is selected as the anonymization algorithm and the size of K is set as 10. The identifying attribute is set as age and acceptable generalization intervals for age is defined. Crime category is set as sensitive attribute and a numerical value assigned to each category. Note that the toolbox requires all values, whether identifying- or sensitive attribute values, to be in numerical form and where generalization is neccessary, taxonomies have to be converted to representative integers.



Listing 1: UTD Configuration Sample

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?><!--
    Attribute 1 is part of the QID, attribute 2 is sensitive. k =
    10--><!-- Name attributes of "att" nodes are not used, included
     just for reference.-->
2 <config c="0.2" k="10" l="3" method="Incognito_K" t="0.2">
3   <input filename="dataset/regenMOD.data" separator=","/> <!-- If
      left blank, separator will be set as comma by default.-->
4   <output filename="census-income_Anon_all.data" format="genVals"
     /> <!-- Format options = {genVals, genValsDist, anatomy}. If
      left blank,
5   output format will be set as genVals by default.-->
6   <id> <!-- List of identifier attributes, if any, these will be
```

```
        excluded from the output -->

    </id>
    <qid>
        <att index="1" name="age">
            <vgh value="[18:88)">
                <node value="[18:53)">
                    <node value="[18:36)" /> <!--No need to list the
                        leaves, as the domain is continuous.-->
                    <node value="[36:53)" />
                </node>
                <node value="[53:88)">
                    <node value="[53:70)" />
                    <node value="[70:88)" />
                </node>
            </vgh>
        </att>
        <att index='3' name ='location'>
            <vgh value='[0:100)'>
                <node value='[0:50)'>
                    <node value='[0:25)' /> <!--Represents converted
                        addresses.-->
                    <node value='[25:50)' />
                </node>
                <node value='[50:100)'>
                    <node value='[50:75)' />
                    <node value='[75:100)' />
                </node>
            </vgh>
```

```
33          </att>
34      </qid>
35      <sens>
36          <att index="2" name="Sensit">
37              <map>
38                  <entry cat="Assault" int="1"/>
39                  <entry cat="Rape" int="2"/>
40                  <entry cat="Murder" int="3"/>
41                  <entry cat="Robbery" int="4"/>
42                  <entry cat="Family_or_Domestic_Violence" int="5"/>
43                  <entry cat="Other" int="6"/>
44                  <entry cat="Vandalism" int="7"/>
45                  <entry cat="Arson" int="8"/>
46                  <entry cat="Burglary" int="9"/>
47                  <entry cat="Corruption_or_Embezzlement" int="10"/>
48                  <entry cat="Theft" int="11"/>
49                  <entry cat="Forgery_or_Fraud" int="12"/>
50                  <entry cat="Drunken_Driving" int="13"/>
51                  <entry cat="Disorderly_conduct" int="14"/>
52                  <entry cat="Drug_related" int="15"/>
53                  <entry cat="Illegal_gambling" int="16"/>
54              </map>
55          </att>
56      </sens>
57  </config>
```



Listing 2 illustrates another example configuration file. In this configuration, Incognito is again selected as the anonymization algorithm, but with L = 2 diversity as an additional privacy requirement. Furthermore, two attributes are listed as identifying attributes for experimental purposes. Age and crime are assigned

representative numerical values for generalization, while the type of witness is considered the sensitive attribute.

Listing 2: UTD LDiversity Sample

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><!-- Sample configuration file. Attributes 12 and 0 are part of the QID, attribute 41 is sensitive. k = 32--><!-- Name attributes of "att" nodes are not used, included just for reference.-->
<config c="0.2" k="2" l="2" method="Incognito_L" t="0.2">
    <input filename="dataset/ldivexputd.data" separator=","/> <!-- If left blank, separator will be set as comma by default.-->
    <output filename="out.data" format="genVals"/> <!-- Format options = {genVals, genValsDist, anatomy}. If left blank, output format will be set as genVals by default.-->
    <id> <!-- List of identifier attributes, if any, these will be excluded from the output -->
    </id>
    <qid>
        <att index="0" name="age">
            <vgh value="[18:88)">
                <node value="[18:53)">
                    <node value="[18:36)"/>
                    <node value="[36:53)"/>
                </node>
                <node value="[53:88)">
                    <node value="[53:70)"/>
                    <node value="[70:88)"/>
                </node>
            </vgh>
        </att>
```

```
21        <att index="1" name="crime">
22            <vgh value="[0:18)">
23                <node value="[0:6)">
24                    <node value="[0:3)"/> <!-- Represents crime
                          categories converted to numeric values.-->
25                    <node value="[3:6)"/>
26                </node>
27                <node value="[6:12)">
28                    <node value="[6:9)"/>
29                    <node value="[9:12)"/>
30                </node>
31                <node value="[12:18)">
32                    <node value="[12:15)"/>
33                    <node value="[15:18)"/>
34                </node>
35            </vgh>
36        </att>
37    </qid>
38    <sens>
39        <att index="2" name="Sensit">
40            <map>
41                <entry cat="Witness" int="1"/>
42                <entry cat="Proxy" int="2"/>
43                <entry cat="Victim" int="3"/>
44            </map>
45        </att>
46    </sens>
47 </config>
```

# Appendix C

The original (unchanged) data was set as the training set and the various anonymized sets as the testing sets with IB1 classification. Figure 1 shows an example of using Weka for classification.

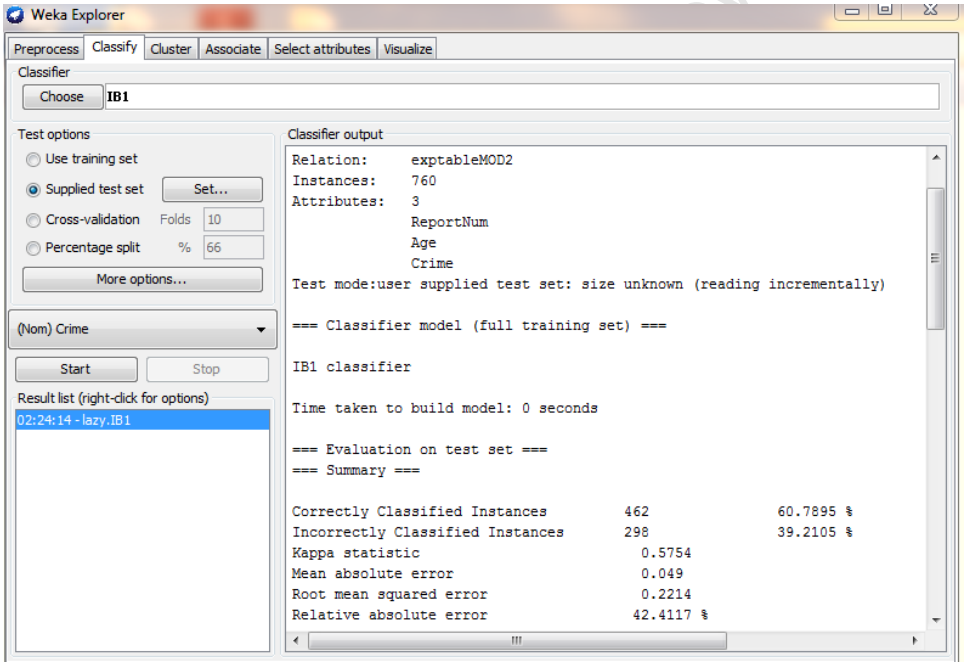



Figure 1: Appendix 3: Weka Classification Accuracy

In a weka configuration file, the table name is defined as "@relation". Different attributes are defined as "@attribute" and the last attribute is the class type, meaning weka will attempt to classify the last attribute based on the combination of complimenting attributes defined before the class. "@data" comprises of comma-separated representations of the data table records.

Listing 3: Weka Header

```
@relation record1000

@attribute Location numeric
@attribute Age numeric
@attribute Crime {Murder,Other,Disorderly_conduct,Assault,
    Family_or_Domestic_Violence,Theft,Vandalism,Drug_related,Rape,
    Corruption_or_Embezzlement,Forgery_or_Fraud,Burglary,
    Illegal_gambling,Arson,Drunken_Driving,Robbery}

@data
751,64,Robbery
752,86,Vandalism
753,20,Arson
754,23,Assault
755,51,Rape
756,80,Corruption_or_Embezzlement
757,42,Burglary
758,70,Other
759,37,Drunken_Driving
760,24,Assault
761,43,Disorderly_conduct
762,27,Family_or_Domestic_Violence
763,56,Murder
764,32,Robbery
765,28,Rape
766,35,Corruption_or_Embezzlement
767,23,Murder
```

# Appendix D

Listing 4 illustrates the Python script used for communication between mobile devices and a back-end database. Note how the main menu is set up in line 10 and a user is directed to different response URL's based on the options in line 16, 18 and 20.

Listing 4: USSD Server

```
1 from flask import Flask
2 from flask import request
3 from flask import Response
4 import nltk
5 import MySQLdb
6 import time
7
8 app = Flask(__name__)
9
10 @app.route("/main")
11 def mainapp():
12     content = """<?xml version="1.0" encoding="utf-8" ?>
13<request>
14<headertext>Thank you for choosing to report a crime. Please select
     a category?</headertext>
15<options>
```

```
<option command="1" order="1" callback="http://markpc.cs.uct.ac.za/menu1?cmd=1"
display="true">Violent Crimes</option>
<option command="2" order="2" callback="http://markpc.cs.uct.ac.za/menu1?cmd=2"
display="true">Property Crimes</option>
<option command="3" order="3" callback="http://markpc.cs.uct.ac.za/menu1?cmd=3"
display="true">Misdemeanors/Other</option>
</options>
</request>
"""

    r = Response(content, content_type="text/xml");

    return r


@app.route("/menu1")
def m1():
    cmd = request.args.get('cmd', None)
    if cmd == "1":
        content = """<?xml version="1.0" encoding="utf-8" ?>
<request>
<headertext>Which one of these headings best describe the violent crime?</headertext>
<options>
<option command="1" order="1" callback="http://markpc.cs.uct.ac.za/menu2?cmd=1"
```

```
40 display="true">Assault</option>
41<option command="2" order="2" callback="http://markpc.cs.uct.ac.za/
    menu2?cmd=2"
42 display="true">Rape</option>
43<option command="3" order="3" callback="http://markpc.cs.uct.ac.za/
    menu2?cmd=3"
44 display="true">Murder</option>
45<option command="4" order="4" callback="http://markpc.cs.uct.ac.za/
    menu2?cmd=4"
46 display="true">Robbery</option>
47<option command="5" order="5" callback="http://markpc.cs.uct.ac.za/
    menu2?cmd=5"
48 display="true">Family/Domestic</option>
49<option command="6" order="6" callback="http://markpc.cs.uct.ac.za/
    menu2?cmd=6"
50 display="true">Other</option>
51</options>
52</request>
53"""
54      elif cmd == "2":
55          content = """<?xml version="1.0" encoding="utf-8" ?>
56<request>
57<headertext>Which one of these headings best describe the property
    crime?</headertext>
58<options>
59<option command="1" order="1" callback="http://markpc.cs.uct.ac.za/
    menu2?cmd=1"
60 display="true">Vandalism</option>
61<option command="2" order="2" callback="http://markpc.cs.uct.ac.za/
```

```
        menu2?cmd=2"
display="true">Arson</option>
<option command="3" order="3" callback="http://markpc.cs.uct.ac.za/
        menu2?cmd=3"
display="true">Burglary</option>
<option command="4" order="4" callback="http://markpc.cs.uct.ac.za/
        menu2?cmd=4"
display="true">Corruption</option>
<option command="5" order="5" callback="http://markpc.cs.uct.ac.za/
        menu2?cmd=5"
display="true">Embezzlement</option>
<option command="6" order="6" callback="http://markpc.cs.uct.ac.za/
        menu2?cmd=6"
display="true">Larceny/Theft</option>
</options>
</request>
"""
    elif cmd == "3":
        content = """<?xml version="1.0" encoding="utf-8" ?>
<request>
<headertext>Which one of these headings best describe the
        misdemeanor/other crime of ?</headertext>
<options>
<option command="1" order="1" callback="http://markpc.cs.uct.ac.za/
        menu2?cmd=1"
display="true">Forgery/Fraud</option>
<option command="2" order="2" callback="http://markpc.cs.uct.ac.za/
        menu2?cmd=2"
display="true">Drunken Driving</option>
```

```
<option command="3" order="3" callback="http://markpc.cs.uct.ac.za/menu2?cmd=3"
display="true">Disorderly Conduct</option>
<option command="4" order="4" callback="http://markpc.cs.uct.ac.za/menu2?cmd=4"
display="true">Drug related</option>
<option command="5" order="5" callback="http://markpc.cs.uct.ac.za/menu2?cmd=5"
display="true">Illegal Gambling</option>
<option command="6" order="6" callback="http://markpc.cs.uct.ac.za/menu2?cmd=6"
display="true">Other</option>
</options>
</request>
"""
    else:
        content = """<?xml version="1.0" encoding="utf-8" ?>
<request>
<headertext>That was an invalid response. Please dial the number *120*12021# again and enter a valid response.</headertext>
</request>
"""

    r = Response(content, content_type="text/xml");

    return r

@app.route("/menu2")
def m2():
```

```
107     content = """<?xml version="1.0" encoding="utf-8" ?>
108<request>
109<headertext>Please enter the victim name. Enter your own name if you
        are the victim. If unsure just enter zero(0).</headertext>
110<options>
111<option command="1" order="1" callback="http://markpc.cs.uct.ac.za/
      menu3?cmd=1"
112 display="false">KThxBye</option>
113</options>
114</request>
115 """
116     r = Response(content, content_type="text/xml");
117
118     return r
119
120 @app.route("/menu3")
121 def m3():
122     content = """<?xml version="1.0" encoding="utf-8" ?>
123<request>
124<headertext>Please enter the suspect or perpetrator name if you know
        it. If unsure or unknown, just enter zero(0).</headertext>
125<options>
126<option command="1" order="1" callback="http://markpc.cs.uct.ac.za/
      menu4?cmd=1"
127 display="false">KThxBye</option>
128</options>
129</request>
130 """
131     r = Response(content, content_type="text/xml");
```

```
132
133     return r
134
135 @app.route("/menu4")
136 def m4():
137     content = """<?xml version="1.0" encoding="utf-8" ?>
138 <request>
139 <headertext>Please briefly describe the crime.</headertext>
140 <options>
141 <option command="1" order="1" callback="http://markpc.cs.uct.ac.za/
      final?cmd=1"
142 display="false">KThxBye</option>
143 </options>
144 </request>
145 """
146     r = Response(content, content_type="text/xml");
147
148     return r
149
150 @app.route("/final")
151 def mfinal():
152     content = """<?xml version="1.0" encoding="utf-8" ?>
153 <request>
154 <headertext>Thank you for your trouble to dial the UCT crime
      reporting service and your report.</headertext>
155 </request>
156 """
157     r = Response(content, content_type="text/xml");
158
```

```
159     return r
160
161 if __name__ == "__main__":
162     app.run(host = '0.0.0.0', port = 80)
```

# References


[1] CC Aggarwal and PS Yu. A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):609–623, 2009. 25

[2] Mafruz Zaman Ashrafi and See Kiong Ng. Collusion-resistant anonymous data collection method. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 69, 2009. 14, 15, 16, 47

[3] Mina Askari, Reihaneh Safavi-Naini, and Ken Barker. An information theoretic privacy and utility measure for data sanitization mechanisms. *Proceedings of the second ACM conference on Data and Application Security and Privacy - CODASKY '12*, page 283, 2012. 1, 17, 20, 27

[4] Roland Assam. Preserving privacy of moving objects via temporal clustering of spatio-temporal data streams. *Workshop on Security and Privacy in GIS and LBS*, page 9, 2011. 9, 27

[5] Maurizio Atzori, Francesco Bonchi, Fosca Giannotti, and Dino Pedreschi. Anonymity preserving pattern discovery. *The VLDB Journal*, 17(4):703–727, November 2006. 19

[6] K. Bache and M. Lichman. UCI machine learning repository, 2013. 61, 65

[7] Randolph C. Barrows and Paul D. Clayton. Privacy , Confidentiality : and Electronic Medical Records Abstract The enchanced Goals of Informantional Security In Health Care. 1996. 21

[8] Aaron Beach, Mike Gartrell, and Richard Han. Social-K: Real-time K-anonymity guarantees for social network applications. *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 600–606, March 2010. 27

[9] Justin Brickell and Vitaly Shmatikov. Efficient anonymity-preserving data collection. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, page 76, 2006. 2, 15, 16, 20

[10] Jianneng Cao and Barbara Carminati. Castle: Continuously anonymizing data streams. *IEEE Dependable and Secure Computing*, 8(3):337–352, 2011. 27

[11] Rajarshi Chakraborty, Claire Vishik, and H. Raghav Rao. Privacy preserving actions of older adults on social media: Exploring the behavior of opting out of information sharing. *Decision Support Systems*, pages 1–9, January 2013. 8, 10, 12

[12] Jay Chen, Saleema Amershi, Aditya Dhananjay, and Lakshmi Subramanian. Comparing web interaction models in developing regions. *Proceedings of the First ACM Symposium on Computing for Development - ACM DEV '10*, page 1, 2010. 3

[13] Riccardo Dondi, Giancarlo Mauri, and Italo Zoppis. The -Diversity problem: Tractability and approximability. *Theoretical Computer Science*, (Mfcs 2011), May 2012. 31, 32, 61

[14] Nick Doty, Erik Wilde, and U C Berkeley. Geolocation Privacy and Application Platforms Position Paper. pages 65–69, 2010. 9

[15] Federal Bureau of Investigation. Uniform Crime Reporting Handbook. *U.S. Department of Justice*, 2004. 43, 49

[16] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. A framework for efficient data anonymization under privacy and accuracy con-

straints. *ACM Transactions on Database Systems*, 34(2):1–47, June 2009. 20, 53

[17] Dara Hallinan, Michael Friedewald, and Paul McCarthy. Citizens' perceptions of data protection and privacy in Europe. *Computer Law & Security Review*, 28(3):263–272, June 2012. 8, 10, 11, 12

[18] Vijay S. Iyengar. Transforming data to satisfy privacy constraints. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, page 279, 2002. 63

[19] Steinar Kristoffersen and Fredrik Ljungberg. Mobile use of IT. *The Proceedings of Information Systems Research Seminar (IRIS22), Jyväskylä*, 1999. 3, 9

[20] K LeFevre. Mondrian multidimensional k-anonymity. *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, 2006. 20, 54

[21] K LeFevre, DJ DeWitt, and R Ramakrishnan. Incognito: Efficient full-domain k-anonymity. *2005 ACM SIGMOD international conference on Management of data*, (05305002), 2005. 53

[22] J Li, BC Ooi, and W Wang. Anonymizing streaming data for privacy protection. *IEEE Conference on Data Engineering, ICDE*, 2008. 27

[23] N Li, T Li, and S Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. *23rd International Conference on Data Engineering (ICDE 07)*, (2), 2007. 47, 48

[24] T. Y. Lin and N. Cercone, editors. *Rough Sets and Data Mining: Analysis of Imprecise Data*. Kluwer Academic Publishers, Norwell, MA, USA, 1996. 22

[25] Ashwin Machanavajjhala and Daniel Kifer. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007. 27, 28, 30, 31, 46, 47

[26] Charles Martel, Glen Nuckolls, Premkumar Devanbu, Michael Gertz, April Kwong, and Stuart G. Stubblebine. A general model for authenticated data structures. *Algorithmica*, 39(1):21–41, 2004. 21

[27] Shinya Miyakawa, Nobuyuki Saji, and Takuya Mori. Location l-Diversity against Multifarious Inference Attacks. *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, 1:1–10, July 2012. 34

[28] Norshidah Mohamed and Ili Hawa Ahmad. Information privacy concerns, antecedents and privacy measure use in social networking sites: Evidence from Malaysia. *Computers in Human Behavior*, 28(6):2366–2375, November 2012. 8

[29] Mehmet Ercan Nergiz, Christopher Clifton, Senior Member, and Ahmet Erhan Nergiz. Multirelational k-Anonymity. *IEEE 23rd International Conference on Data Engineering (ICDE)*, 21(8):1104–1117, 2009. 53, 54

[30] Hyoungmin Park and Kyuseok Shim. Approximate algorithms for K-anonymity. *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD '07*, page 67, 2007. 54

[31] P. Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001. 18

[32] Manya Sleeper, Divya Sharma, and Lorrie Faith Cranor. I Know Where You Live : Analyzing Privacy Protection in Public Databases Public Databases. 2011. 20, 22

[33] Latanya Sweeney. Datafly: a system for providing anonymity in medical data. *Database Security XI: Status and Prospects*, 1998. 53

[34] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, October 2002. 27, 53

[35] Stefano Taddei and Bastianina Contena. Privacy, trust and control: Which relationships with online self-disclosure? *Computers in Human Behavior*, 29(3):821–826, May 2013. 9, 11, 12, 14

[36] Hongwei Tian and Weining Zhang. Extending l-Diversity for Better Data Anonymization. *2009 Sixth International Conference on Information Technology: New Generations*, pages 461–466, 2009. 32, 61, 62

[37] Hongwei Tian and Weining Zhang. Extending l-diversity to generalize sensitive data. *Data & Knowledge Engineering*, 70(1):101–126, January 2011. 61, 63

[38] K Wang. Privacy Preserving Data Publishing. *ACM Computing Surveys*, 42file(4):1–113, 2010. 24, 44, 53, 54

[39] Weiping Wang, Jianzhong Li, Chunyu Ai, and Yingshu Li. Privacy protection on sliding window of data streams. *2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007)*, pages 213–221, November 2007. 27

[40] Kuang-Wen Wu, Shaio Yan Huang, David C. Yen, and Irina Popova. The effect of online privacy policy on consumer privacy concern and trust. *Computers in Human Behavior*, 28(3):889–897, May 2012. 7, 8, 11, 12, 13

[41] Yanfang Wu, Tuenyu Lau, David J. Atkin, and Carolyn a. Lin. A comparative study of online privacy regulations in the U.S. and China. *Telecommunications Policy*, 35(7):603–616, August 2011. 8, 11