A THREE-DIMENSIONAL APPROACH TOWARDS MEASURING SENDER ANONYMITY

A Thesis by

Umesh Marappa Reddy

Bachelor of Technology, Jawaharlal Nehru Technological University, 2008

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2010

A THREE-DIMENSIONAL APPROACH TOWARDS MEASURING SENDER ANONYMITY

The following faculty members have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

_____
Neeraj Jaggi, Committee Chair

_____
Rajiv Bagai, Committee Member

_____
Hamid M. Lankarani, Committee Member

_____
Bing Tang, Committee Member

DEDICATION

To Venkateswara swamy, my family, and my country

ACKNOWLEDGEMENTS

First, I would like to thank my family and friends for their unending support and love from my childhood to now.

Dr. Neeraj Jaggi has been an excellent advisor to me. Over the two years that I have worked with him, he shared his ideas and time generously, and has staunchly supported my research. I gratefully acknowledge his support as an advisor.

I would also like to thank rest of the thesis committee Dr. Rajiv Bagai, Dr. Hamid Lankarani, and Dr. Bin tang for their precious time and comments.

I am very fortunate to be part of the anonymity research group of Dr. Rajiv Bagai, which helped me to explore more in the field of anonymity. I thank each and every member of the research group for sharing their wisdom in web anonymity. I have learned so much from all of you.

This work would not have been possible without the support of my friends, Nikhil, Vamshi, and Sreenu. You're always there for me, when I need help with my research and when I need moral support.

Last but not least, I would like to thank Wichita State University and United States of America for giving me this opportunity.

ABSTRACT

Privacy plays an indispensable role over internet transactions. There are existing methods in place which enhance the level of privacy over World Wide Web (WWW). Anonymity intends to hide the identity of the user during web transactions. Sender anonymity attempts to hide the identity of the sender of a message. Various anonymous systems exist today which provide different levels of anonymity to their users. Comparing various anonymous systems on the Web by quantifying the degree of anonymity provided by them is a daunting and challenging task. This thesis illustrates with examples that existing measures in literature are not sufficient to fully characterize the anonymity provided by a system and introduces a new isolation measure. The new isolation measure is proposed based on the existence of outliers in a distribution which is critical towards quantifying the anonymity provided by the system.

The need for three distinct aspects of anonymity is justified, important from the perspectives of a user, a system designer and an attacker, leading to a three-dimensional approach towards measuring sender anonymity. A 3-tuple metric is proposed, and various properties of metric are also discussed. The interpretation of proposed metric depends on the desired characteristics of the system. Two anonymous systems can be compared in terms of the degree of anonymity provided, using the proposed 3-tuple metric and appropriate weights reflecting the attributes desired in the system. The proposed metric is applied to quantify the degree of anonymity provided by various existing anonymous systems.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS (continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The number of users employing the services of internet is spiraling day by day. At the same time, the number of foes trying to assail the privacy of users is also increasing day by day. The ability to maintain one's privacy on the web has always been an important concern [1], [2]. There are various types of threats possible during web transactions which include eavesdropping, identity theft, etc. Different methods have been proposed to make the user transactions more secure against different attacks by the adversaries. The technique of encryption is proposed to make the communication more secure and make it insulated from eavesdropper. Introducing delay in the communication avoids timing attacks from the adversary. However, if the attacker succeeds in identifying the user he intended to attack, then he can use this information to perform attacks like denial of service.

With the increase in identity theft over World Wide Web, anonymity is emerging as one of the important research areas. Anonymity on the Web is critical to enable applications such as e-voting, auctions, payments and to provide necessary cyber-security measures [3]. Anonymity is becoming increasingly important in sensor networks [4] and also towards routing in ad hoc networks [5]. With renewed emphasis on identity protection, privacy awareness and selective information sharing on social networks [6], [7], the ability to protect online activities from possible misuse is gaining importance [8], [9].

1.1 Anonymous Systems

Anonymous systems allow a user to access the Web (for surfing, chatting etc.) while hiding the user's identity from potential attackers. Note that anonymity is different from data privacy, which could be achieved using end-to-end encryption. Since anonymity cannot be provided end-to-end, the complete network must get involved in the design of anonymous system, making it very difficult to provide complete anonymity to the users.

Various approaches have been proposed towards providing anonymous transactions on the

Web. A widely employed technique for providing anonymity is by introducing proxy server(s) between a sender-receiver pair. The goal of an anonymous system includes providing one or more of - sender anonymity, receiver anonymity, and unlinkability of sender and receiver. This thesis focuses solely on sender anonymity.

Potential eavesdroppers or attackers could reside anywhere in the network, and protecting the identity of the sender of a message from malicious nodes is the key concern in most anonymous systems. The following are some of the existing anonymous systems.

1.1.1 Mixes

Mix network, also called digital mixes, was invented by David Chaum [10] to provide anonymous e-mail solution. A simple mix network consists of proxy server(s) between the sender and the receiver. The proxy server(s) is(are) intended to receive messages from multiple users. The received messages are dispatched to destination in such a way to avoid correspondence between incoming messages and outgoing messages. Figure 1.1 shows a mix network with three proxy servers employed between source and destination.

In mix network, public key cryptography is adopted to make the communication secure. The message to be sent is encrypted in multiple layers with the public encryption key of the proxy servers. Once the proxy server receives the encrypted message, it decrypts the top most layer of encryption with its private key, and forwards the resulting message to next hop irrespective of the order the message was received. Anonymity is achieved as each node appears to be a proxy server to adjacent node.

Mixes adopt different strategies to avoid correspondence between outgoing and incoming messages, such as -

- Threshold mix: In threshold mix [12], each proxy server waits for X messages to arrive before it decrypts the messages and forwards them to the next proxy server. So an attacker observing the behavior of a proxy server cannot correlate the logical relationship between incoming and outgoing messages.

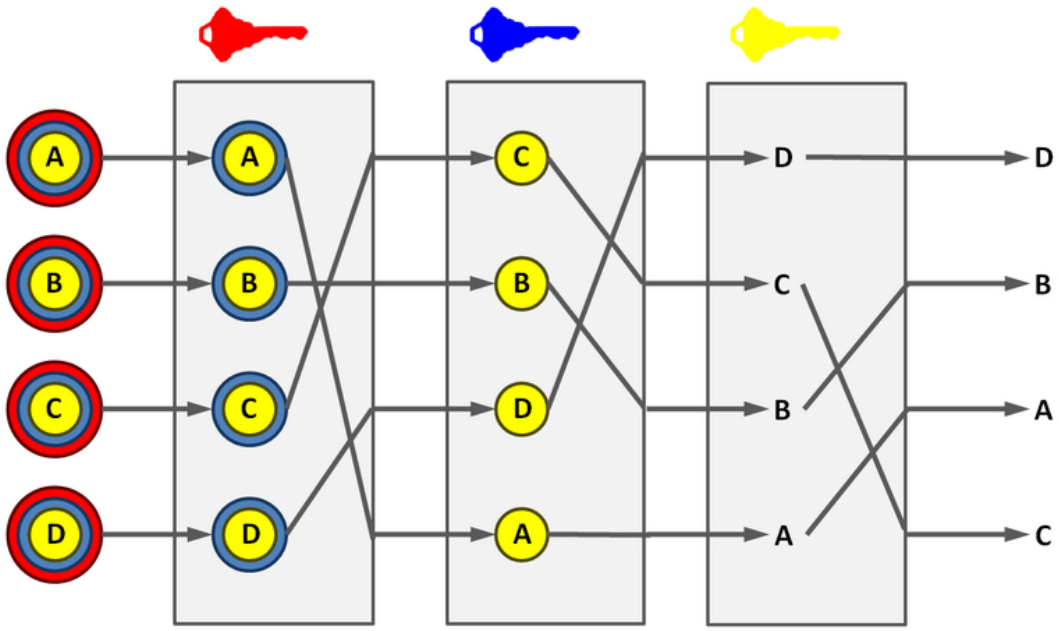

Figure 1.1 – Mix network with three proxy servers [11].

- Stop and Go mixes: In such mixes [12], each received message is delayed for a random amount of time in the proxy server before it is forwarded. This avoids the attacker from gaining any information by performing timing attack.

- Pool mix: In pool mixes [12], a proxy server waits till a threshold number of messages are received. Once the threshold is reached, it randomly stores some of the messages in its buffer, and sends dummy messages in place of genuine messages. The buffered messages are forwarded randomly in the subsequent operations.

1.1.2 Crowds

Crowds [13] is another anonymous system which adopts the concept of implementing proxy servers between sender and receiver[1] to achieve anonymity. The forwarding agents are the users in crowds. In a crowds system, if a sender wants to send data message to a receiver, it first sends the data message to a jondo[2], which is randomly chosen. The jondo forwards the received

[1] A receiver here refers to an end server.
[2] In a crowds system, every user is referred to as a jondo including the sender of the message.

data message either to the next jondo with probability p_f or to the receiver with probability *(1-p_f)*. The forwarding jondo has no information about the initiator of the message. It always assumes that the node which forwarded the data message to it is another jondo. Even if the node which is forwarding data to a jondo is the initiator, the jondo will always assume it to be a jondo. This mechanism of forwarding data ensures anonymity in the system.

There is always a tradeoff between anonymity that can be achieved in a crowds system and end to end latency in delivering a data message. As the forwarding probability p_f increases, data message is forwarded multiple number of times between jondos before it actually reaches the receiver. This results in increased latency. But due to the increased number of jondos that forward the message, anonymity is also increased considerably.

The data message forwarding between any two jondos always takes place using symmetric key cryptography[3]. An eavesdropper[4] cannot interpret the data message unless he has the exact key used to encrypt it. Even though a jondo can always observe the contents of the entire message, it has absolutely no information regarding who the actual sender is. The meta data[5] exposes just the identity of the forwarding jondo from which the data message has been received, but not its predecessor.

After the Data message has been received at the receiver from the last forwarding jondo, the receiver initiates a response message and forwards it to the last jondo from which the data message was received. The jondo then forwards the response message to the jondo that forwarded the data message to it. In this way, the response message reaches the sender via the reverse path along which the data message was forwarded.

The attack model in a crowds system includes compromised jondos, which share the data message amongst themselves. Such compromised jondos are called collaborating jondos. The primary goal of the collaborating jondos is to identify the sender of a particular message, thus compromising the anonymity provided by the system.

[3]Both the jondos, share an identical key which is used to encrypt and decrypt the data message at both ends.
[4]An eavesdropper tries to observe the ongoing communication between any two jondos.
[5]The Data appended by the network layer to the Data message.

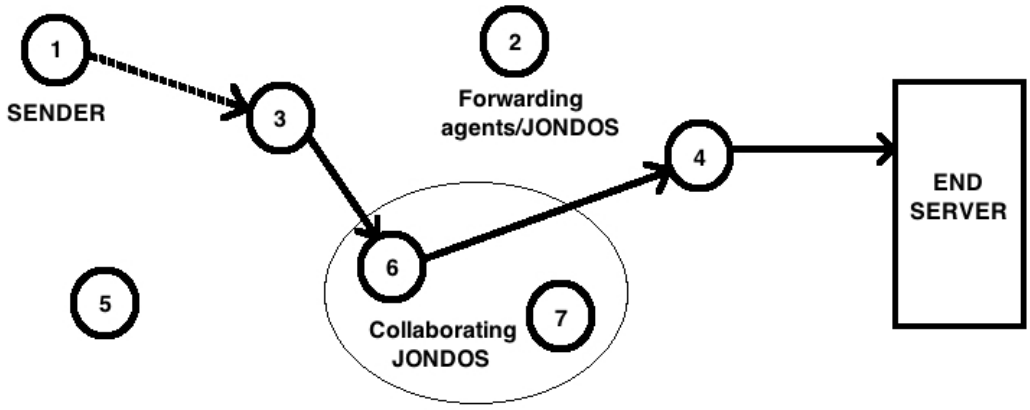

Figure 1.2 – Crowds anonymous system [14].

1.1.3 Onion Routing

Onion routing [15], [16] is yet another anonymous system which resembles a combination of mixes and crowds. In an onion routing system, the data message is forwarded across different onion routers before it reaches the receiver. An onion routing system employs public key cryptography[6] to encrypt/decrypt the messages in the system. Every onion router knows the public keys of all the other onion routers in the system. Initially the sender sends the data message to an onion router. The onion router then randomly selects a path of onion routers through which the data message would traverse before it reaches the receiver. Every router in the path is only aware of the onion router to which it needs to forward the data message.

Figure 1.3 explains the behavior of onion routing. Let O_1, O_2, O_3, ...O_N be the onion routers in the path and P_1, P_2, P_3.... P_N be the public keys of respective onion routers. O_1 encrypts the data message repeatedly with the public keys in the order of P_N, P_{N-1},P_{N-2}, ...P_2. At each level of encryption, the encryption header contains the address of next onion router. O_1 then forwards the data to O_2. O_2 strips out the top layer of encryption with its private key. The decrypted header yields the next onion router's address to O_2. O_2 then forwards the message to the next onion

[6]Each onion router distributes a public key used to encrypt the message, which requires a unique private key to decrypt it.

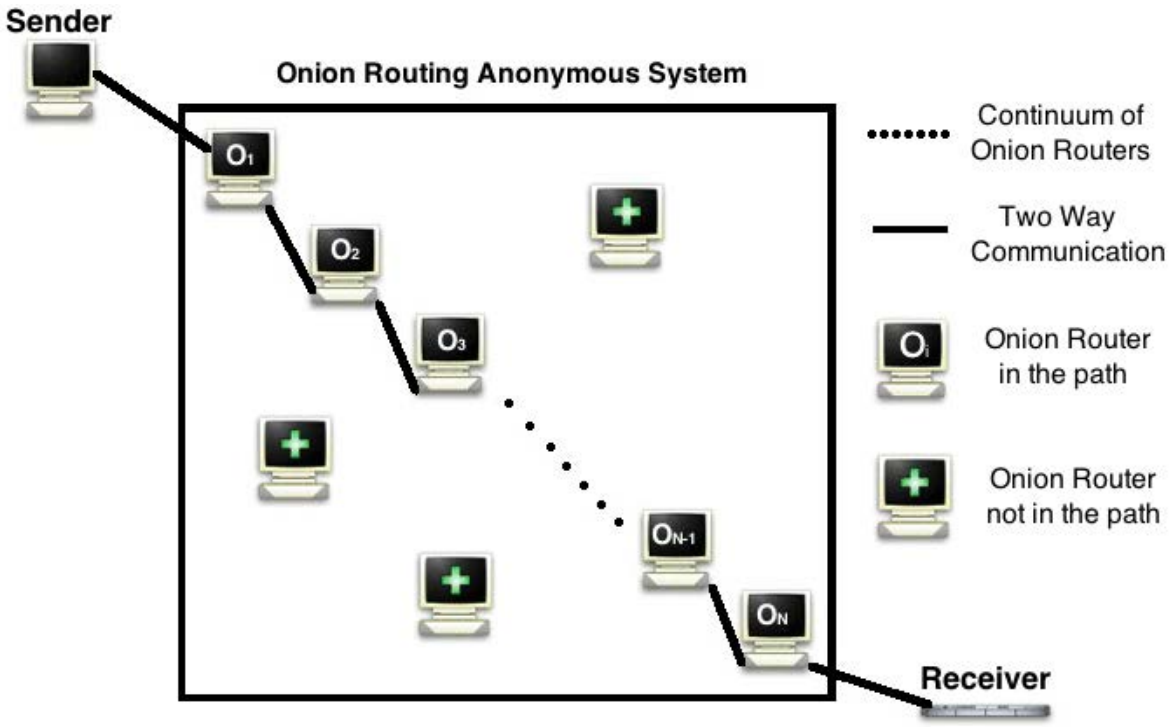

Figure 1.3 – Onion routing anonymous system.

router (O_3) and the process continues until the message reaches the last onion router (O_N) on the path, from where the actual data message is sent to the receiver.

When the receiver receives the data message, a response message is sent to the last onion router(O_N) that forwarded data message to it. O_N then encrypts the response message repeatedly with the public keys in the order of P_1, P_2, P_3, ...P_{N-2}, P_{N-1}. At each level of encryption, the encryption header contains the address of next onion router. O_N then forwards the data to O_{N-1}. O_{N-1} strips out the top layer of encryption with its private key. The decrypted header yields the next onion router's address to O_{N-1}. O_{N-1} then forwards the message to the next onion router (O_{N-2}) and the process continues until the message reaches the onion router (O_1), which is the successor of sender on the forward path, from where the response message is sent to the actual sender of the data message.

1.2 Thesis Structure

The rest of the thesis is organized as follows. Chapter 2 presents an overview of the research done so far in the area of measuring anonymity. Chapter 3 shows that the existing measures of S, d and θ are not sufficient to completely characterize the sender anonymity provided by the system, or to compare and distinguish two systems appropriately in terms of anonymity provided. Chapter 4 discusses the need to quantify the isolation of a user (or a set of users) in the distribution, and proposes a new measure to quantify this isolation, based upon presence of outliers in a distribution and discusses its properties. Then, a three dimensional approach towards measuring sender anonymity is proposed in Chapter 5 and the justification for the three distinct aspects in this approach, considering the perspectives of a user, system designer and attacker respectively, is provided. The interpretation of the proposed 3-tuple metric in terms of desired characteristics of the system is discussed in Chapter 5, and various properties of the proposed metric are outlined. Chapter 5 also models the desired attributes of an anonymous system using weights, and argues that the overall anonymity of the system should be viewed in relation to these weights assigned to the three distinct aspects.The comparison of anonymous systems using the proposed 3-tuple metric and predetermined weights is also performed in Chapter 5. Finally, the proposed metric is applied to existing anonymous systems in Chapter 6, and the conclusions are summarized in Chapter 7.

The research results in this thesis have also been submitted for publication [17].

CHAPTER 2

RELATED WORK

2.1 Anonymity Preliminaries

Anonymity is defined [18] as the state of being non-identifiable within a set of subjects, the anonymity set, where anonymity set A corresponds to the set of users who could potentially be suspected by the attacker. In case of sender anonymity, which is important in most applications, the attacker attempts to identify the originator or sender of a particular message from possible senders belonging to the anonymity set A.

The objective of the anonymous system design is to prevent such identification under different models of attacks possible in the system. Once an attacker has gained sufficient information using a particular attack, he/she attempts to identify the sender in a probabilistic manner. The attacker assigns a probability p_i to each user i in the anonymity set A, where p_i is a measure of suspicion with which the attacker considers user i to be the actual sender of the message. Let N denote the size of anonymity set A ($N = |A|$). A probability distribution is valid if $\forall i \in \{1, ..., N\} : 0 \leq p_i \leq 1$ and $\sum_{i=1}^{N} p_i = 1$. Note that anonymity of the system is completely compromised when attacker assigns p_j = 1 for some $j \in A$ and user j is the actual sender of the message. On the other hand, the system apparently achieves its goals when attacker assigns p_j = 0 corresponding to the actual sender (user j). Given the probability distribution **p**, the anonymity set A and the sender $j \in A$, quantifying the lack of sender identification information available to the attacker amounts to measuring the level or degree of anonymity provided by the system.

2.2 Existing Metrics

Various approaches have been presented to quantify the degree of anonymity provided by an anonymous system. One of the approaches considers just the probability assigned to the actual sender j [13], [19]. Among these, [13] provides conditions for probable innocence of the actual sender in the crowds system. Particularly conditions on the cardinality of anonymity set (N) with

respect to the number of compromised collaborating intermediate nodes is outlined such that the probability assigned to the sender of a message, $p_j \leq \frac{1}{2}$. [19] performs a probabilistic analysis of onion routing in a black-box model based upon this measure and presents results on expected and worst-case anonymity achieved under given usage patterns of all users. Note that in order to minimize the probability assigned to the actual sender, one needs to consider an attack model and a typical attack scenario.

2.2.1 Simple Entropy Based Measure (S)

Entropy is one of the information theoretic measures that can be used to quantify the level of anonymity in a system. Entropy is the measure of uncertainty in a system. A simple entropy based measure is given by [12]:

$$S = -\sum_{i=1}^{N} p_i log_2(p_i) \tag{2.1}$$

Consider a system with N number of users. The maximum anonymity of $log_2 N$ is achieved in a scenario, where the attacker has no information on who the actual sender is and therefore assigns equal probabilities p_i[7] $= \frac{1}{N}\ \forall i \in \{1, 2, ..., N\}$. Let D denote this probability distribution. In this scenario, S is given by,

$$S = -\sum_{i=1}^{N} \frac{1}{N} log_2 \frac{1}{N} = -log_2 \frac{1}{N} = log_2 N \tag{2.2}$$

The anonymity provided by the system in a scenario is 0 when the attacker can assign p_i = 1 for any $i \in \{1, 2, ..., N\}$. S in this scenario is given by,

$$S = -log_2 1 = 0$$

Therefore, the entropy measure S, in an anonymous system lies in the range $[0,\ log_2 N]$.

[7] p_i is the probability that i^{th} user is the suspected sender of message. p_i=1 indicates that the attacker suspects the i^{th} user to be the sender with probability 1.

2.2.2 Normalized Entropy Based Measure (d)

Normalized entropy based measure is also an information theoretic measure. Simple entropy measure S depends on the size of anonymity set N, whereas normalized entropy measure does not. Entropy measure S is normalized against the optimal situation, in which the attacker has no additional information about the sender of the message. The normalized entropy measure is given by [14],

$$d = \frac{S}{log_2 N} = -\frac{\sum_{i=1}^{N} p_i log_2(p_i)}{log_2 N} \tag{2.3}$$

Note that, when the sender j is completely identified (p_j = 1), d equals zero. And d = 1 under distribution D, since the sender j is as likely to have sent the message as any other user in A in the eyes of the attacker and is thus relatively (completely) unidentifiable. Note that a distribution **p** with p_j = 0 is less anonymous under this measure than the distribution D, even though the attacker's suspicion towards sender j is minimal. Such a behavior could be justified using the overall system performance. The system achieves maximum anonymity when none of the users is suspected more than the others. In case a non-sender user k is suspected to be sender with a high probability, even though p_j = 0, the attacker could act upon the suspicion and could cause denial-of-service or other attacks on user k, which is undesirable from the system's perspective. Moreover, such a scenario is more likely to result in user k being exposed, the next time user k sends a message. Since the performance of the anonymous system is measured by its ability to hide the identity of the sender each time a message is sent (and not by how much the attacker is misled), the distribution D is considered to provide the maximum anonymity according to this measure. In general, the average entropy computed using probability distributions corresponding to different messages should be considered. [14] applies the normalized entropy based measure d on existing anonymous systems such as mixes [10], onion routing [15] and crowds [13].

2.2.3 Local Anonymity (θ)

Shortcomings of both simple and normalized entropy based measures are outlined in [20], where counterexamples are constructed to show that two distributions could have the same measure but provide practically different anonymities. Also, it shows that non-desirable (or less anonymous) distributions could measure arbitrarily close to maximum anonymity, as measured using d. [20] introduces local anonymity θ as:

$$\theta = \max_{1 \leq i \leq N} p_i. \tag{2.4}$$

The authors argue that θ is more important than d from the perspective of a user and derive appropriate relations between θ, S and d. Note that since the probability assigned to the actual sender is bounded by θ, a low value of θ is sufficient to hide the actual sender's identity. However, considering θ alone is not sufficient to characterize the anonymity of the distribution, since the probability assigned to the sender must be viewed in relation with the probabilities assigned to other users. Thus the normalized entropy d is important as well.

Various approaches consider specific attack models such as local eavesdropper or collaborating intermediate nodes to analyze the properties of the system. However, the approach presented in [14], [20], and [12] is slightly different, and considers only the posterior probabilities assigned by an attacker to various users who are suspected of having sent a particular message, after the information has been gained from the attack. This thesis model is also based upon the latter approach, and is thus suitable to analyze and compare anonymous systems under any general attack model.

Other related research on anonymity metrics includes [21], [22], [23]. [21] structured different metrics based on the layer at which the attack is considered to be occurred and proposed a way to merge these metric structures such a way to achieve comprehensive anonymity metrics including both network and application layer. [22] proposes a combinatorial approach towards measuring anonymity based upon the network graph of the system. Instead of considering a single user or message, [22] introduces a system wide metric based on the permanent of the graph's ad-

jacency matrix. [23] revisits the combinatorial approach, generalizes the metric by modeling user relations and provides an algorithm to compute the refined metric.

CHAPTER 3

DRAWBACKS OF EXISTING MEASURES

3.1 Drawbacks and Counter Examples

In this section, different examples of anonymous systems with different anonymity properties are considered, and it is shown that even though an attacker assigns quite different probability distributions to these systems, the existing anonymity measures declare the two systems to be equally anonymous. In addition, it is shown that simple and normalized entropy measures could even provide misleading information in some cases, which suffice to illustrate that the current measures are insufficient to completely characterize the anonymity properties of a system.

Let D_1 and D_2 denote the a posteriori probability distributions assigned in the two anonymous systems, and let the sizes of the anonymity set be denoted N_1 and N_2 respectively.

3.1.1 Example 1

Consider two systems with N_1 = 7 and $N_2 = N$ + 1, where the distributions are given by:

$$D_1 : p_1 = 0.4,\ p_2 = 0.4,\ p_k = 0.04,\ \forall k \in \{3,4,....,\ 7\}.$$
$$D_2 : p_1 = 0.4,\ p_k = \tfrac{0.6}{N},\ \forall k \in \{2,3,....,\ N+1\}.$$

Here, it is apparent that the attacker has more information about the actual sender in system 2 compared with system 1. This is because one user has been assigned a significantly larger probability than the others (particularly if N is large). Since the attacker is able to isolate a single user in the anonymity set and suspect that user to be the potential sender much more than others, the anonymity provided by system 2 should be lower than that provided by system 1. The normalized entropy for system 1 is computed as:

$$d_1 = -\frac{\sum_{i=1}^{N_1} p_i log_2(p_i)}{log_2 N_1} = 0.708 \tag{3.5}$$

The normalized entropy for system 2 is computed as:

$$d_2 = -\frac{\sum_{i=1}^{N_2} p_i log_2(p_i)}{log_2 N_2} = -\frac{0.4log_2(0.4) + 0.6log_2(\frac{0.6}{N})}{log_2(N+1)} \tag{3.6}$$

In order to illustrate the drawback in d, N is chosen appropriately, such that both the systems measure equal in terms of normalized entropy. Equating (3.5) and (3.6),

$$0.708 = \frac{0.971 + 0.6log_2 N}{log_2(N+1)}$$

For large values of N, using $log_2(N+1) \approx log_2 N$,

$$\therefore log_2 N = 8.99 \implies N \approx 508$$

Choosing N = 508, d_2 = 0.708 = d_1 (upto 3 places of decimal). The local anonymity measure for both the systems is the same, i.e. θ_1 = θ_2 = 0.4. Note that the simple entropy measure (2.2.1) for the two systems is not the same and is in fact counterintuitive. Computing S_1 = 1.986 and S_2 = 6.364. Thus $S_1 < S_2$, which seems to suggest that system 2 has higher degree of anonymity than system 1, which is counterintuitive [8].

The sizes of anonymity sets are different in the previous example (7 and 509 respectively). In the following example, we show that even though the anonymity set sizes are the same, the existing measures fail to differentiate systems, reiterating the drawback of existing measures.

3.1.2 Example 2

Consider two systems with N_1 = N_2 = 100, where the distributions are given by (for some $N \in \{1, ..., 98\}$):

$$D_1 : p_1 = p_2 = 0.2,\ p_k = \frac{0.6}{98},\ \forall k \in \{3,4,....,\ 100\}.$$

$$D_2 : p_1 = 0.2,\ p_k = \frac{0.5}{N},\ \forall k \in \{2,3,....,\ N+1\},\ p_l = \frac{0.3}{(99-N)},\ \forall l \in \{N+2,....,\ 100\}.$$

[8] Therefore, simple entropy measure is not considered in further examples.

The normalized entropy for system 1 is computed as:

$$d_1 = -\frac{\sum_{i=1}^{100} p_i log_2(p_i)}{log_2 100} = 0.804 \tag{3.7}$$

$$d_2 = -\frac{0.2log_2(0.2) + 0.5log_2(\frac{0.5}{N}) + 0.3log_2(\frac{0.3}{99-N})}{log_2 100} \tag{3.8}$$

N is chosen appropriately, such that both the systems measure (approximately) equal in terms of normalized entropy. Equating (3.7) and (3.8),

$$0.5log_2 N + 0.3log_2(99 - N) = 3.856$$

The integral value of N which satisfies the above equation with minimum error is given by N = 15. Choosing N = 15, d_2 = 0.806 $\approx d_1$.[9] The local anonymity measure for both the systems is the same, i.e. $\theta_1 = \theta_2$ = 0.2. In this example, since the attacker is able to isolate two users in system 1 and one user in system 2, the degree of anonymity provided by the two systems appears to be different. However, the measures d and θ for both the systems are the same.

Next subsection provides an example to show that normalized entropy measure could sometimes be misleading.

3.1.3 Example 3

Consider two systems with $N_1 = N_2$ = N, where the distributions are given by

$$D_1 : p_1 = 0.3,\ p_k = \tfrac{0.7}{N-1},\ \forall k \in \{2,3,....,N\}.$$
$$D_2 : p_1 = p_2 = 0.3,\ p_k = \tfrac{0.4}{N-2},\ \forall k \in \{3,4,....,N\}.$$

Here, for all values of $N > 3$, $\theta_1 = \theta_2 = 0.3$ and $d_1 > d_2$. For eg., for N = 10, d_1 = 0.933 and d_2 = 0.834. However, system 2 seems to provide a higher degree of anonymity than the system 1, since the attacker is able to isolate a single user in the anonymity set in system 1 (particularly for large values of N). Thus the measure d is misleading in this scenario. In order to better understand

[9] Note that a more sophisticated construction of D_1 and D_2 could be performed to achieve $d_1 = d_2$ to any desired level of accuracy.

the construction of this example, ignore p_1 from both distributions and consider distributing a total probability of 0.7 to N - 1 users. The distribution which results in maximum entropy is given by $p_i = \frac{0.7}{N-1}, \forall i \in \{1, ..., N-1\}$, which corresponds to D_1. Thus, $S_1 > S_2$ and since $N_1 = N_2$, $d_1 > d_2$.

The above examples illustrate that the existing metrics are not sufficient to completely characterize the anonymity of a system.

CHAPTER 4

MEASURING ISOLATION IN ANONYMOUS SYSTEMS

4.1 Introduction

In all the examples considered in the Chapter 3, one observation is apparently common. In one of the distributions, exactly one user is being isolated by the attacker i.e. the probability assigned to exactly one user is substantially higher than that assigned to other users in the anonymity set. As a result, the degree of anonymity provided by the corresponding anonymous system seems to be lower than the other system. However, the measures d and θ are unable to satisfactorily quantify this isolation of the user in the anonymity set. In this Chapter, a new measure called Isolation factor (denoted I) is proposed to measure the degree of isolation in the system.

4.2 Role of Outlier in Measuring Isolation

Note that a user in the anonymity set is considered isolated, if the probability assigned to the user is significantly larger than that assigned to other users in the set. Viewing these probabilities as a set of observations, the probability assigned to the isolated user corresponds to an outlier in this set.[10] This concept of outlier is adopted to measure or quantify the isolation provided by the system.

An outlier is a statistical observation which appears to deviate considerably from the rest of the observations. Multiple approaches exist to detect an outlier in a sample of data [24]. Model-based methods identify observations which are deemed unlikely based upon the mean and standard deviation of the distribution and are commonly used. One such method, which is able to identify multiple outliers, is the Peirce's criterion [25], [26], [27]. Here, observations deviating from the mean beyond an appropriately computed threshold are declared outliers. The principle used in Peirce's criterion [25] states that the proposed observations should be rejected when the probability of the system of errors obtained by retaining them is less than that of the system of errors

[10] In statistics, an outlier is an observation which is numerically distant from the rest of the data.

obtained by their rejection multiplied by the probability of making so many and no more abnormal observations.

When the number of observations is relatively small (< 60), a simplified procedure to detect outliers based upon Peirce's criterion could be used [27]. However, a detailed procedure is proposed in [26] which is valid for upto 150 observations. When the number of observations is larger, the algorithm is applied to multiple sets of observations, obtained by dividing the original set into equal sized pieces of < 150 observations each, as suggested in [26].

4.3 Outlier Detection

This subsection discusses the algorithm used to detect outliers in a set of observations, and illustrates it using an example. The algorithm starts by assuming that there is at least one outlier in the set. Using some calculations, this hypothesis is either accepted or rejected. If accepted, the algorithm assumes that there are at least two outliers in the set and so on. The algorithm stops when a hypothesis gets rejected and the number of outliers is declared to be the value corresponding to the previous iteration. Note that, if an initial assumption is made that number of outliers is > 1, it leads to inaccurate results and hence the algorithm must be executed in iterations, until the number of assumed outliers exceeds the number of outliers in the set.

Let N denote the total number of observations and n denote the total number of suspected observations during an iteration of the algorithm. The sample standard deviation is denoted ε, and the error threshold used in outlier detection is given by $x.\varepsilon$. $\lambda.\varepsilon$ is the mean error after n observations have been declared outliers. To declare n observations as outliers, the following inequality should be satisfied [26].

$$\lambda^{N-n} e^{\frac{1}{2}n(x^2-1)} (\psi x)^y < Q^N, \tag{4.9}$$

where $\psi x = \frac{2}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{1}{2}x^2}$

$$Q^N = \frac{n^n (N-n)^{N-n}}{N^N} \tag{4.10}$$

$$\lambda^{N-n} R^n = Q^N. \tag{4.11}$$

$$x^2 = 1 + \frac{N-n}{n}(1-\lambda^2) \tag{4.12}$$

$$R = e^{\frac{1}{2}(x^2-1)} \psi x \tag{4.13}$$

Table 4.1 – Computation of x for given N and n.

log R	9.2000	9.3683	9.3420	9.3457	9.3452
log $(1-\lambda^2)$	9.4629	9.5755	9.5606	9.5628	9.5625
x	1.6560	1.8050	1.7837	1.7868	1.7864
log R	9.3683	9.3420	9.3457	9.3452	9.3451

Consider an example distribution with one outlier and N = 7, given by p_1 = 0.4, p_i = 0.1, $\forall i \in \{2, ..., 7\}$. Assuming n = 1, Q is computed using (4.10). Now, assuming $log_{10}R = 9.2$[11], λ is computed using (4.11) and x is computed using (4.12). Then, R is obtained using (4.13) (an appropriate table can also be used [27]) and this value of R is used to recompute λ and x. This procedure is followed until the value of x converges. The converged value of x is used to define the error threshold. The computation for the example is shown in Table 4.1. For the example distribution, sample standard deviation ε = 0.1134 and the mean $\bar{p}$ = $\frac{1}{7}$. Assuming n = 1, x = 1.7864 from Table 4.1. The error threshold $x.\epsilon$ = 0.2026. The observation i is declared outlier if $|p_i - \bar{p}| > x.\epsilon$. Thus, p_1 is declared outlier in this iteration. In the next iteration, assume n = 2, compute x and check for outliers. In this iteration too (only) p_1 is declared to be an outlier. Thus, the algorithm stops and declares the detection of one outlier.

The algorithm detects all outliers in the set, deviating in either direction from the mean. Since the analysis requires observations whose value is significantly larger than the mean, the output of the algorithm is trimmed and only the outliers which are greater than mean are chosen.

[11] Infact, $log_{10}R$ = 10-9.2, the notation followed here is same as in [25], and [26].

4.4 Isolation Factor

Using the above approach, all outliers in the probability distribution significantly larger than the mean are detected. Let L denote the number of outliers detected. A new measure, Isolation factor (I) is defined in order to measure the extent of additional information the attacker would gain due to the presence of these outliers. If $L = 0$, I is defined to be zero. The desired properties of this measure include:

- Isolation factor should decrease as number of outliers increase, since the information gained by the attacker decreases as the number of isolated users increase.
- Isolation factor should be proportional to the extent of deviation of the outliers, since the larger the deviation, the higher is the perceived suspicion of the attacker towards the corresponding users.
- For the same number and value(s) of outlier(s), the Isolation factor should increase with an increase in the size of the anonymity set, as the attacker's suspicion set becomes a smaller fraction of the total number of users.

For a probability distribution **p** with N users and $L > 0$ outliers, let us assume without loss of generality, that the first L probabilities in **p** correspond to the outliers. The Isolation factor I is defined as:

$$I = \frac{\sqrt{\sum_{i=1}^{N} (p_i - \bar{p})^2} - \sqrt{\sum_{i=L+1}^{N} (p_i - \bar{q})^2}}{max\{1, L\}} \tag{4.14}$$

where $\bar{p}$ is the sample mean with outliers (and equals $\frac{1}{N}$) and $\bar{q}$ is the sample mean without outliers, i.e. $\bar{q} = \frac{\sum_{i=L+1}^{N} p_i}{N-L}$. Note that the definition of the measure satisfies the desired properties mentioned above. The numerator in (4.14) measures the impact of outliers in increasing the standard deviation of the distribution, while the denominator adjusts this impact based upon the number of outliers.

4.5 Properties of IF

Proposition 1: Isolation factor I lies in the range [0, 1].

Proof: When there are no outliers, $L = 0$, $\bar{q} = \bar{p}$ and $I = 0$, indicating no isolation in the distribution. When user j is completely isolated i.e. $p_j = 1$ and $p_i = 0, \forall i \neq j$,

$$I = \sqrt{(1 - \frac{1}{N})^2 + \frac{N-1}{N^2}} = \sqrt{\frac{N-1}{N}}$$

In this case, $I \to 1$ as $N \to \infty$. For all other scenarios, including more than one outlier, $0 < I < 1$, as shown below.

$$\begin{aligned} I &= \frac{\sqrt{\sum_{i=1}^{N}(p_i - \bar{p})^2} - \sqrt{\sum_{i=L+1}^{N}(p_i - \bar{q})^2}}{max\{1, L\}} \\ &\leq \sqrt{\sum_{i=1}^{N}(p_i - \bar{p})^2} - \sqrt{\sum_{i=l+1}^{N}(p_i - \bar{q})^2} \\ &\leq \sqrt{\sum_{i=1}^{N}(p_i - \bar{p})^2} \end{aligned}$$

Therefore,

$$\begin{aligned} I^2 &\leq \sum_{i=1}^{N}(p_i - \bar{p})^2 \\ &\leq \sum_{i=1}^{N}(p_i^2 - \frac{1}{N}) \\ &\leq \sum_{i=1}^{N} p_i^2 \end{aligned}$$

The maximum value of I^2 is given by solution to the following optimization problem, which can be shown to equal 1.

$$\text{max} \sum_{i=1}^{N} p_i^2, \text{ s.t. } \sum_{i=1}^{N} p_i = 1 \text{ and } 0 \leq p_i \leq 1, \forall i \in \{1, 2, ..., N\}.$$
$$\therefore \sum_{i=1}^{N} p_i^2 \leq 1$$

Thus $I^2 \leq 1$ and hence $I \leq 1$.

I = 0 when there is no isolation in the system and $I = 1$ when the degree of isolation in the system is the maximum. Thus, the Isolation factor provides a measure of the extent of additional information the attacker gains due to the presence of the outliers in the distribution. Let the outlier probabilities be denoted $p_1, p_2..., p_L$, for $L > 0$. Let $\bar{l}$ denote the mean of outlier probabilities, i,e.

$$\bar{l} = \frac{\sum_{i=1}^{L} p_i}{L}.$$

Proposition 2: Isolation factor I satisfies the following inequality:

$$I \leq \bar{l}(1 + \frac{1}{\sqrt{N}}).$$

Proof: The proposition holds for $L = 0$, $\bar{l} = 0$ and $I = 0$. For $L > 0$,

$$\begin{aligned} I &= \frac{\sqrt{\sum_{i=1}^{N}(p_i - \bar{p})^2} - \sqrt{\sum_{i=L+1}^{N}(p_i - \bar{q})^2}}{L} \\ &= \frac{\sqrt{\sum_{i=1}^{L}(p_i - \bar{p})^2 + \sum_{i=L+1}^{N}(p_i - \bar{p})^2}}{L} - \frac{\sqrt{\sum_{i=L+1}^{N}(p_i - \bar{q})^2}}{L} \end{aligned}$$

Therefore

$$I \leq \frac{\sqrt{\sum_{i=1}^{L}(p_i - \bar{p})^2} + \sqrt{\sum_{i=L+1}^{N}(p_i - \bar{p})^2}}{L} - \frac{\sqrt{\sum_{i=L+1}^{N}(p_i - \bar{q})^2}}{L} \tag{4.15}$$

The last inequality follows since for $A, B \geq 0$, $\sqrt{A + B} \leq \sqrt{A} + \sqrt{B}$. Expanding,

$$\sum_{i=L+1}^{N}(p_i - \bar{p})^2 = \sum_{i=L+1}^{N}(p_i - \bar{q})^2 + (N - L)(\bar{q} - \bar{p})^2. \tag{4.16}$$

Using (4.15), (4.16) and the relation for $\sqrt{A + B}$ above,

$$\begin{aligned} I &\leq \frac{\sqrt{\sum_{i=1}^{L}(p_i - \bar{p})^2} + \sqrt{(N - L)(\bar{q} - \bar{p})^2}}{L} \\ &= \frac{\sqrt{\sum_{i=1}^{L}(p_i - \bar{p})^2} + \sqrt{N - L}(\bar{p} - \bar{q})}{L} \end{aligned}$$

The equality follows since $\bar{p} > \bar{q}$. Since $p_i > \bar{p}, \forall i \in \{1, ..., L\}$,

$$
\begin{aligned}
I &\leq \frac{\sum_{i=1}^{L} \sqrt{(p_i - \bar{p})^2} + \sqrt{N-L}(\bar{p} - \bar{q})}{L} \\
&= \frac{\sum_{i=1}^{L} (p_i - \bar{p}) + \sqrt{N-L}(\bar{p} - \bar{q})}{L} \\
&= \frac{\sum_{i=1}^{L} p_i - L\bar{p} + \sqrt{\frac{N-L}{N^2}}(1 - N\bar{q})}{L} \\
&= \frac{L\bar{l} - L\bar{p} + \sqrt{\frac{1}{N} - \frac{L}{N^2}} L(\bar{l} - \bar{q})}{L} \\
&= \bar{l} - \bar{p} + \sqrt{\frac{1}{N} - \frac{L}{N^2}}(\bar{l} - \bar{q}) \\
&\leq \bar{l} - \bar{p} + \sqrt{\frac{1}{N}}(\bar{l} - \bar{q}) \\
&= \bar{l} + \sqrt{\frac{1}{N}}(\bar{l} - \bar{q} - \sqrt{\frac{1}{N}}) \\
&\leq \bar{l}(1 + \frac{1}{\sqrt{N}}).
\end{aligned}
$$

Note that, the relations $\bar{p} = \frac{1}{N}$ and $1 - N\bar{q} = l(\bar{l} - \bar{q})$ are used above.

From Proposition 2, when the size of anonymity set (N) is considerably large, the value of Isolation factor does not exceed the mean of outlier probabilities. This is desirable considering the objectives of this new measure. Consider two probability distributions with $L(> 1)$ outliers, with the same mean $\bar{l}$. In one distribution, all the outlier probabilities are the same and equal the mean. However, in the other distribution, one of the outlier probabilities is larger than the others. Since the Isolation factor is expected to measure the additional information gained by the attacker due to the presence of all outliers, the value of Isolation factor in the latter distribution should not get heavily influenced due to one large outlier probability.

CHAPTER 5

THREE DIMENSIONAL MEASURE OF ANONYMITY

All three measures namely, normalized entropy d, local anonymity θ and Isolation factor I, are essential towards measuring the degree of anonymity of an anonymous system. Note that all the three measures lie in the range [0, 1]. Although a higher value of d is desirable from the system's perspective, a lower value of θ and I would be preferred from an end user's perspective.

5.1 Motivation and Perspective

From the definition of anonymity (Section 2.1), the objective of an anonymous system is to guarantee that the sender of a message be non-identifiable within the anonymity set. In the a posteriori probability distribution, this would be achieved when $d = 1$, i.e. $p_i = \frac{1}{N}, \forall i \in \{1, ..., N\}$. Thus, the normalized entropy d is an important measure from the system designer's perspective.

However, as pointed out in [20], a higher value of d may not be sufficient to convince the end user. For a user trying to utilize the services provided by an anonymous system, she would like to have some guarantees on her maximum exposure for any message she sends in the system. This would be achieved when θ is minimized. Thus, the local anonymity θ is an important measure from the end user's perspective.

Now let us consider the attacker's point of view. Once the attacker has performed the attack and has assigned the probabilities based upon the information gained from the attack, the attacker would like to make a guess (his best bet) as to which user(s), from among those in the anonymity set, is (are) most likely to have sent the message. The attacker's task gets easier when there is a single user (or a set of users) which is (are) clearly isolated in the distribution. The larger the value of I, the more successful is the attack from the attacker's perspective. Thus, the Isolation factor I is an important measure from the attacker's perspective. Note that other measures of anonymity attempt to measure the level of anonymity of a system, given an attack. However, Isolation factor attempts to measure the level of success of an attack, given an anonymous system. Thus, including

this measure helps better understand the overall anonymity of the system.

Thus, all the three measures in the 3-tuple metric (d, θ, I) are important measures of sender anonymity provided by a system, albeit one measure may be rendered more important than the others, depending upon the perspective under consideration. At this point, one may be tempted to combine the three measures, for instance $A_1 d + A_2(1-\theta) + A_3(1-I)$, in order to represent the overall anonymity of the system using one metric. However, such combination would lead to loss of information and could also be misleading.

Consider an anonymous system with 3-tuple metric (d, θ, I). For a given N, maximum anonymity is achieved under the probability distribution D, defined in Section 2.2. In this case, $d = 1, \theta = \frac{1}{N}$ and $I = 0$. As $N \to \infty$, $\theta \to 0$ and the maximum possible anonymity corresponds to the 3-tuple metric (1, 0, 0). Representing d, $(1-\theta)$ and (1 - I) on the x-, y- and z- coordinates respectively, the 3-tuple metric corresponds to a point in the 3-dimensional space. The distance of this point from the origin could then be interpreted as the overall anonymity provided by the system, with the maximum anonymity corresponding to a distance of $\sqrt{3}$. A more elaborate method to represent the system's overall anonymity is presented next.

5.2 Role of Weights

In order to characterize the overall anonymity of the system, the desired attributes of the system are represented using weights assigned to each of the three dimensions of the 3-tuple metric, and the overall anonymity of the system is viewed in relation to these weights. Let W_d, W_θ and W_I denote the weights associated with d, θ and I respectively, such that,

$$0 \leq W_d, W_\theta, W_I \leq 1, \quad \text{and} \quad W_d + W_\theta + W_I = 1 \tag{5.17}$$

These weights are designed to reflect the attributes desired in the system. For instance, if an end user is only interested in local anonymity provided by the system, she would view the system (and the 3-tuple metric) using weights W_d = W_I = 0 and W_θ= 1. Once the weights have been assigned,

representing $\sqrt{W_d}.d$, $\sqrt{W_\theta}.(1-\theta)$ and $\sqrt{W_I}.(1-I)$ on the x-, y- and z- coordinates respectively, the 3-tuple metric corresponds to a point in the 3-dimensional unit sphere[12]. The distance of this point from the origin is interpreted as the overall anonymity provided by the system, with the maximum anonymity corresponding to a distance of 1. Specifically, this distance (denoted R) equals

$$R = \sqrt{W_d.d^2 + W_\theta.(1-\theta)^2 + W_I.(1-I)^2}. \tag{5.18}$$

Note that the maximum possible distance,

$$R_{max} = \sqrt{W_d + W_\theta + W_I} = 1.$$

The distribution D with $N \rightarrow \infty$ results in maximum overall anonymity of R = 1, regardless of the weights assigned. Similarly, the distribution with least anonymity, given by p_1 = 1, p_k = 0, $\forall k \in \{2, ..., N\}$ results in minimum overall anonymity of $R = 0$, as $N \rightarrow \infty$ (since $d = 0, \theta = 1$ and $I = \sqrt{\frac{N-1}{N}}$), regardless of the weights assigned.

5.3 Metric Interpretation and Evaluation

Consider two anonymous systems S_1 and S_2. Let the 3-tuple metric for the two systems be given by (d_1, θ_1, I_1) and (d_2, θ_2, I_2) respectively. Let R_1 and R_2 denote their respective distances from origin. If $R_1 > R_2$, system S_1 is considered to be more anonymous than system S_2 and vice versa. Also, system S_1 is considered to be $\frac{R_1}{R_2}$ times more anonymous than system $S2$. Note that this comparison is performed only after the weights have been assigned according to the attributes desired in the system.

The 3-tuple metric is transitive in that, given the weights, if system S_1 is u times more anonymous than S_2 (i.e. $R_1 > R_2$) and S_2 is v times more anonymous than S_3 (i.e. $R_2 > R_3$), then S_1 is u . v times more anonymous than S_3 ($R_1 > R_3$).

Note that weights are designed to model user preferences while comparing the degree of

[12]Note that the square root of weights is considered in order to assign the maximum overall anonymity of 1 to a system with distribution D and $N \rightarrow \infty$, regardless of the weights assigned. Other ways to interpret the 3-tuple metric may also be plausible.

anonymity provided by different anonymous systems. Indeed, with a different choice of weights, the results of such comparisons are expected to differ.

5.4 Counter Examples Revisited

In this section, the examples considered in Chapter 3 are revisited and are interpreted using the proposed 3-tuple metric.

5.4.1 Example 1 Revisited

Consider two systems with N_1 = 7 and N_2 = 509, where the distributions are given by:

$$D_1 : p_1 = 0.4,\ p_2 = 0.4,\ p_k = 0.04,\ \forall k \in \{3,4,....,\ 7\}.$$
$$D_2 : p_1 = 0.4,\ p_k = \tfrac{0.6}{508},\ \forall k \in \{2,3,....,\ 509\}.$$

Here, $d_1 = d_2 = 0.708$ and $\theta_1 = \theta_2 = 0.4$. For D_1, the number of outliers detected $L_1 = 0$ and the Isolation factor $I_1 = 0$. For D_2, the number of outliers detected $L_2 = 1$ and the Isolation factor is computed as $I_2 = 0.398$. Considering weights $W_d = W_\theta = W_I = \frac{1}{3}$, the overall anonymity of system 1 equals $R_1 = 0.788$ and for system 2 equals $R_2 = 0.639$. Thus, system 1 is more anonymous than system 2, under the chosen set of weights. Thus, using the 3-tuple metric allows us to distinguish the two systems in terms of the degree of anonymity provided.

5.4.2 Example 2 Revisited

Consider two systems with $N_1 = N_2 = 100$, where the distributions are given by:

$$D_1 : p_1 = p_2 = 0.2,\ p_k = \tfrac{0.6}{98},\ \forall k \in \{3,4,....,\ 100\}.$$
$$D_2 : p_1 = 0.2,\ p_k = \tfrac{0.5}{15},\ \forall k \in \{2,3,....,\ 16\},\ p_l = \tfrac{0.3}{84},\ \forall l \in \{17,18,....,100\}.$$

Here, $d_1 = 0.804$, $d_2 = 0.806$ and $\theta_1 = \theta_2 = 0.2$. For D_1, the number of outliers detected $L_1 = 2$ and the Isolation factor is computed as $I_1 = 0.136$. For D_2, the number of outliers detected $L_2 = 1$ and the Isolation factor is computed as $I_2 = 0.112$. Considering weights $W_d = W_\theta = W_I = \frac{1}{3}$, the overall anonymity of system 1, $R_1 = 0.823$ and for system 2, $R_2 = 0.832$. Thus, system 2 is

slightly more anonymous than system 1, under the chosen set of weights. In system 2 even though the first user is being isolated, the attacker does not suspect this user a lot more than the next 15 users. On the other hand, in system 1, the attacker suspects first 2 users very highly. Therefore, system 2 turns out to be more anonymous than system 1. Note that, in this example, the choice of weights could play a significant role in distinguishing the two systems. For instance, if W_I = 1, system 2 evaluates to be quite more anonymous than system 1 (R_1 = 0.864 and R_2 = 0.888), since I_1 is 21$\%$ greater than I_2.

5.4.3 Example 3 Revisited

Consider two systems with $N_1 = N_2 = 11$, where the distributions are given by:

$$D_1 : p_1 = 0.3,\ p_k = \tfrac{0.7}{10},\ \forall k \in \{2,3,....11\}.$$
$$D_2 : p_1 = p_2 = 0.3,\ p_k = \tfrac{0.4}{9},\ \forall k \in \{3,4,....11\}.$$

The 3-tuple metric for these systems are given by (0.927, 0.3, 0.219) and (0.821, 0.3, 0.163) respectively. Clearly, $d_1 > d_2$ and $(1 - I_1) < (1 - I_2)$.

Case I: Consider a system designer's perspective and choose W_d = 0.8, W_θ = 0.1 and W_I = 0.1. This results in R_1 = 0.893 and R_2 = 0.811 and system 1 evaluates to be more anonymous than system 2.

Case II: Consider an attacker's perspective and choose W_d = 0.1, W_θ = 0.1 and W_I = 0.8. This results in R_1 = 0.789 and R_2 = 0.823 and system 2 evaluates to be more anonymous than system 1.

Case III: Consider an end user's perspective and choose W_d = 0.1, W_θ = 0.8 and W_I = 0.1. This results in R_1 = 0.734 and R_2 = 0.728 and system 1 evaluates to be slightly more anonymous than system 2 (similar to Case I). Since $\theta_1 = \theta_2$, the degree of anonymity provided by the two systems is similar from the end user's perspective.

Figures 5.1 and 5.2 pictorially depict the two scenarios in cases I and II above. Thus, an anonymous system may be more anonymous from the perspective of system designer, but could

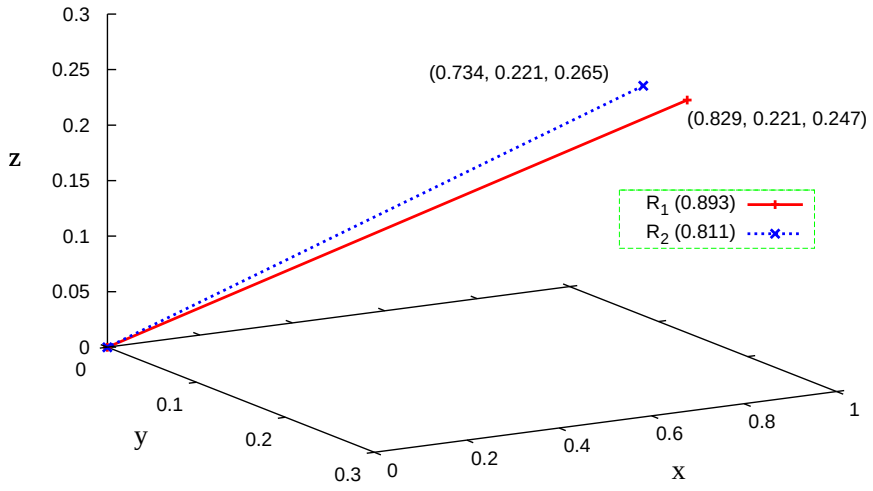

Figure 5.1 – Anonymous system comparison from system designer's perspective.

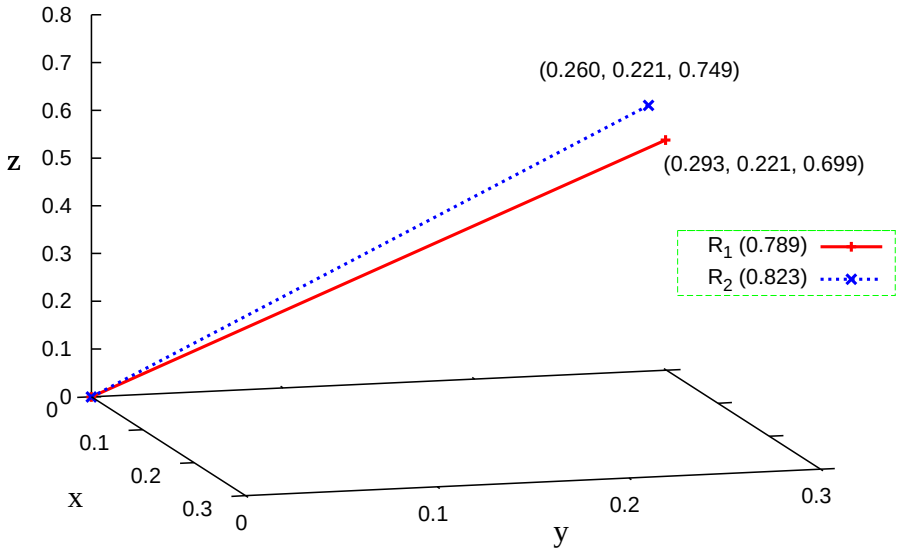

Figure 5.2 – Anonymous system comparison from attacker's perspective.

turn out to be less anonymous for an attacker. Hence, the anonymity of a system should always be viewed keeping the desired attributes of the system in mind.

CHAPTER 6

APPLICATION OF THREE DIMENSIONAL METRIC TO EXISTING ANONYMOUS SYSTEMS

In this chapter, the proposed 3-tuple metric ($d,\ \theta,\ I$) is applied to existing anonymous systems - crowds, onion routing, and mixes. The design choices are also evaluated which could improve the anonymity provided by an anonymous system.

6.1 Crowds

Consider the crowds [13] system with N users, where each user forwards the request to a randomly chosen user with probability p_f and contacts the server directly with probability $(1-p_f)$. [13], [14] consider an attack model with C collaborating jondos (or users) in the system. The size of anonymity set equals N and the probability assigned to each collaborating jondo equals 0. One of the users (predecessor of first collaborating jondo on the path) is suspected with a probability of [13]:

$$p_1 = 1 - p_f\tfrac{N-C-1}{N}.$$

Assuming, all other non-collaborative users are equally suspected,

$$p_i = \tfrac{1-p_1}{N-c-1}=\tfrac{p_f}{N}, \forall i \in \{2, 3, ..., N-C\}.$$

Using this probability distribution, the 3-tuple metric is applied to crowds for various values of p_f, N and C and these systems are compared under equal weights, W_d = W_θ = W_I = $\frac{1}{3}$. Figure 6.1 depicts the overall anonymity (R) for N = 100. As the number of collaborating jondos (C) increases, the anonymity of the system decreases. Further, the decrease in anonymity is linear even when C is considerably large, in contrast with the case when only the normalized entropy (d) is considered [14]. Also, the anonymity increases as the probability of forwarding (p_f) increases for all values of C, with the increase being higher when C is small. Figure 6.2 depicts the anonymity of the system under a fixed number of collaborating jondos (C = 9). It can be seen that the anonymity increases with an increase in N for all values of p_f . However, the anonymity of the system

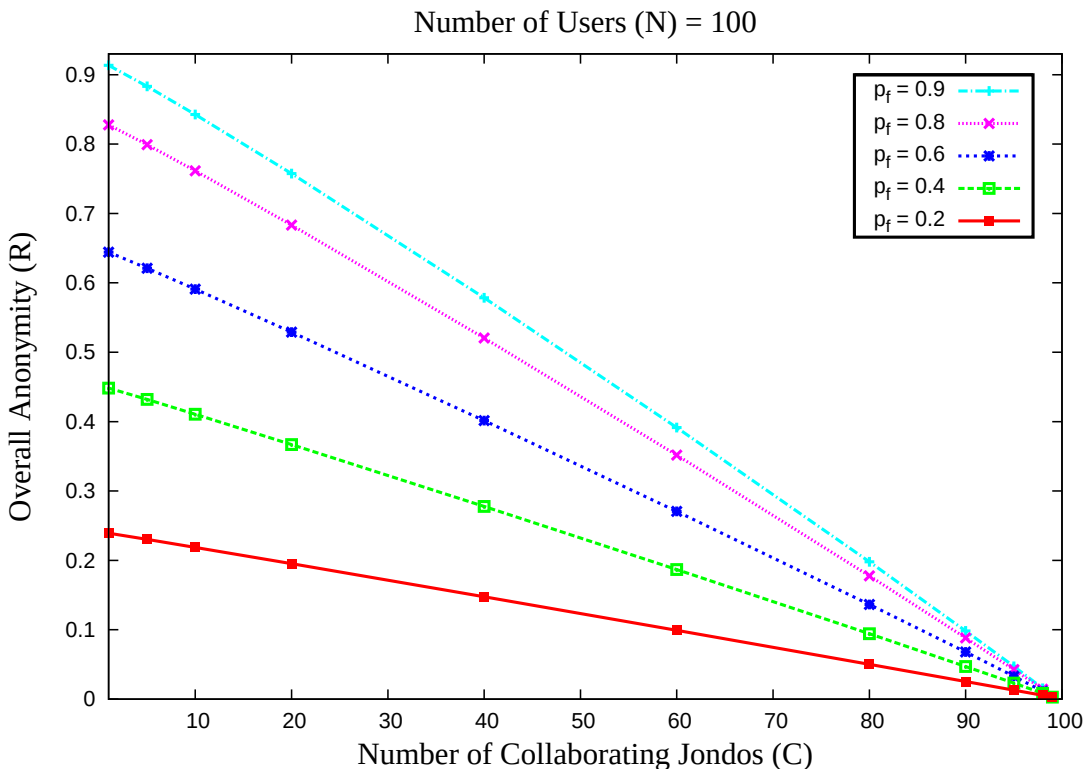

Figure 6.1 – Overall anonymity for crowds.

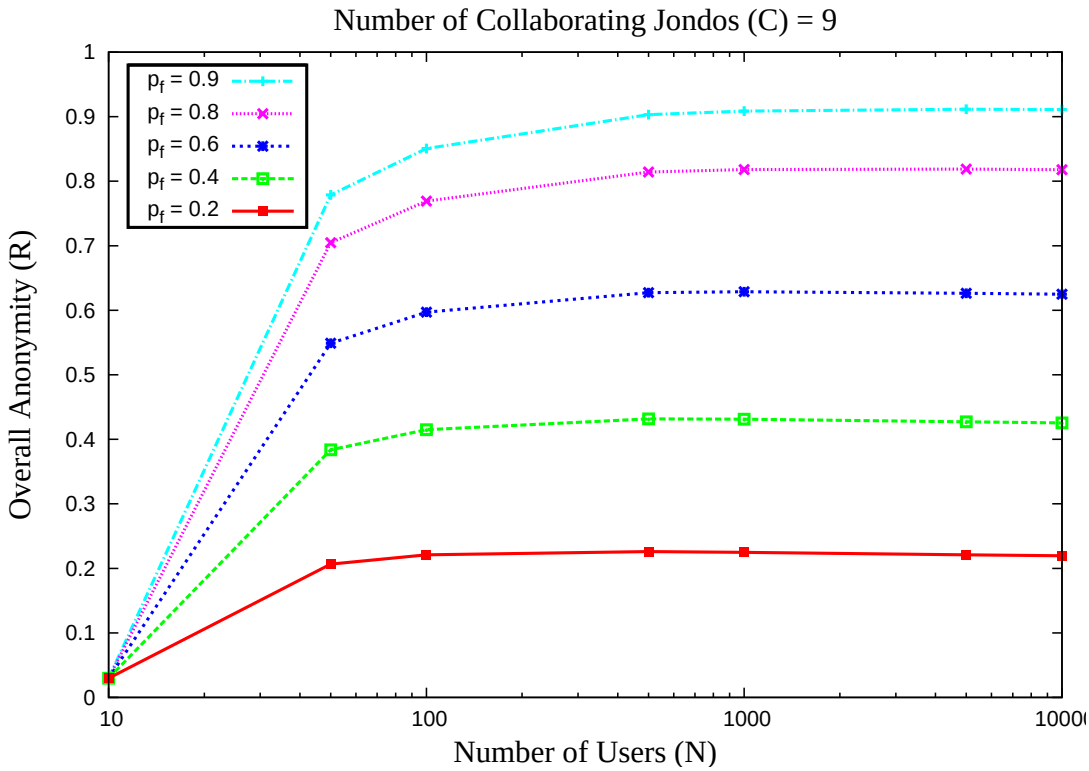

Figure 6.2 – Anonymity saturates as N increases.

saturates as N is increased further, with the saturated anonymity value being quite close to (and greater than) the probability of forwarding (p_f). Thus, in order to achieve an overall anonymity of $>$ 0.75, the following conditions must be satisfied, i.e. either N = 50 and p_f = 0.9 or N = 100 and p_f = 0.8. When C is small, increasing the probability of forwarding (p_f) would substantially increase the anonymity in crowds. However, when C is large, increasing the number of users (N) instead (such that C is a smaller fraction of N) would lead to a higher degree of anonymity.

6.2 Onion Routing

Consider an onion routing system, where data message is sent through the path specified by an initiator or onion router. The data message to be sent is encrypted in layers as per the symmetric keys of the nodes in the path it follows. Each node removes a layer of encryption if the data message is going forward and adds a layer of encryption if the data message is going backward (from receiver to sender).

If an onion routing system consists of N nodes, then the size of anonymity set equals N. The attack model consists of an attacker who tries to narrow down the anonymity set to S from N where $1 \leq S \leq N$. Since the attacker does not have any other information within the anonymity set S, he/she suspects all the S nodes equally to be the initiator of the message. In this scenario, the case when $S = 0$, is considered to be similar to the case where $S = N$. Therefore, the probability distribution is given by:

$$p_i = \frac{1}{S}, \forall i \in \{1, ...S\} \quad \text{and} \quad p_i = 0, \forall i \in \{S+1, ...N\} \quad \text{where} \quad 1 \leq S \leq N$$

Normalized entropy measure d is given by:

$$d = \frac{log_2 S}{log_2 N}$$

Local anonymity is given by:

$$\theta = \frac{1}{S}$$

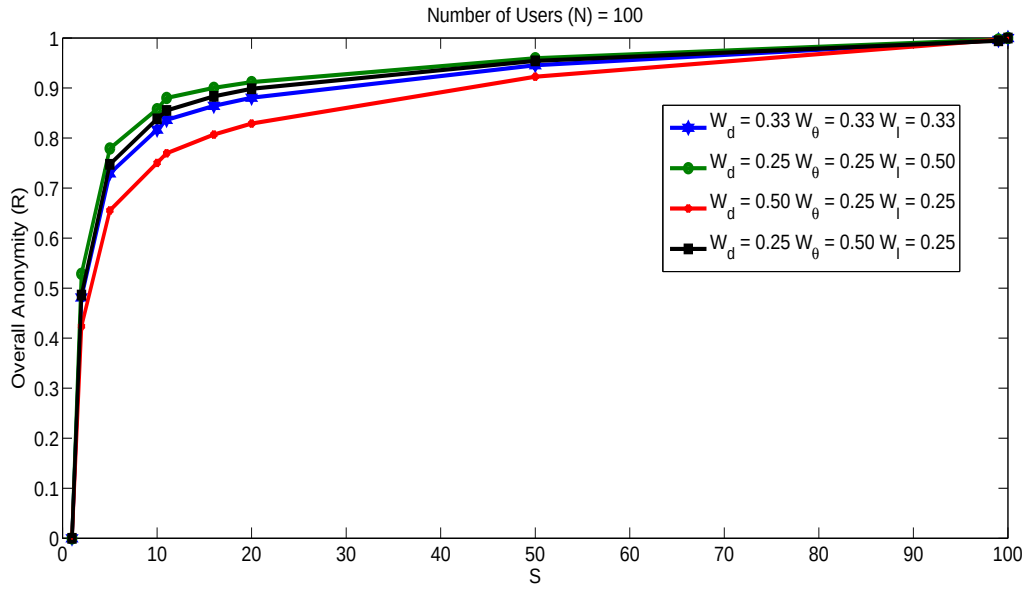

Figure 6.3 – Overall anonymity (R) vs S when N=100.

Based on the above distribution, the 3-tuple metric is calculated for different values of N, S, and the weights W_d, W_θ and W_I.

Figures 6.3 and 6.4 depict the overall anonymity (R) for N=100, N=200 and for different weights. As the value of S increases, the overall anonymity (R) increases to 1 in both the cases. From the plots, it is evident that the shape of curve is similar for different choices of weights and is concave in nature. Varying the weights does not lead to a substantial difference in R in onion routing.

Figure 6.5 shows the behavior of overall anonymity (R) when S equals 10 and for different weights. It can be seen that maximum anonymity is achieved when N equals S. The overall anonymity drops considerably when higher weight is assigned to normalized entropy measure (d). This is because when the value of S is constant, local anonymity θ remains same and with increase in N, the amount of decrease in normalized entropy d is greater than the amount of decrease in factor $(1 - I)$[13].

Figure 6.6 depicts the overall anonymity (R) for various values of S/N. R is zero when S equals 1 and reaches maximum value when S equals N. The shape of curve is concave in nature.

[13]When S is constant, Isolation factor I increases with increase in N.

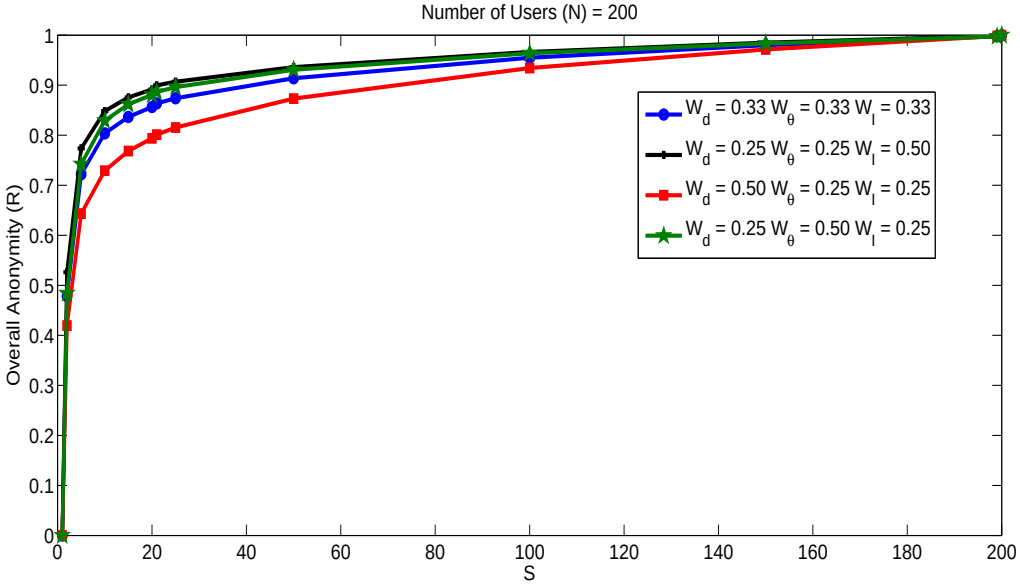

Figure 6.4 – Overall anonymity (R) vs S when N=200.

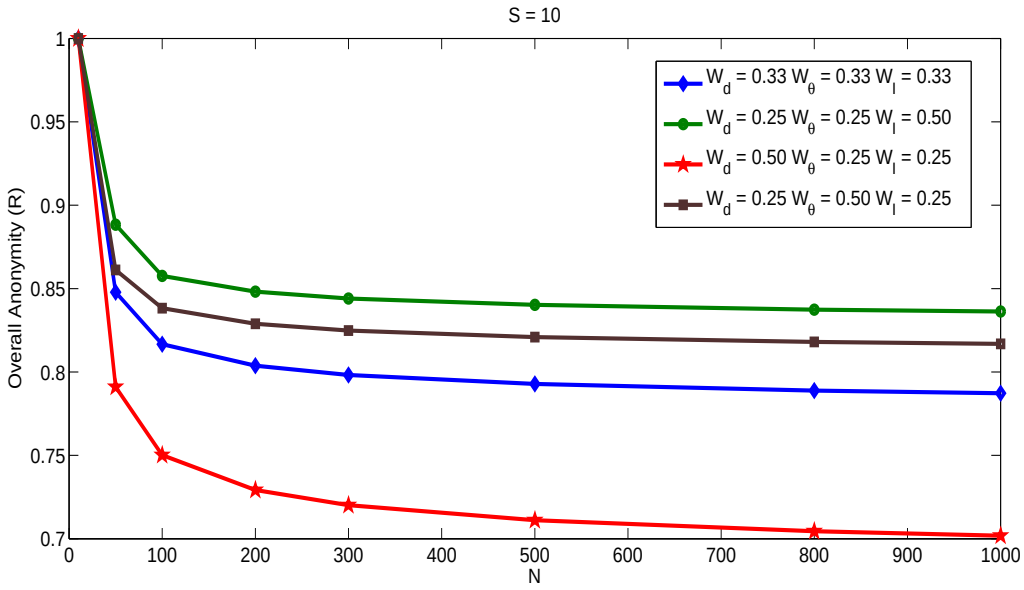

Figure 6.5 – Overall anonymity (R) vs N when S=10.

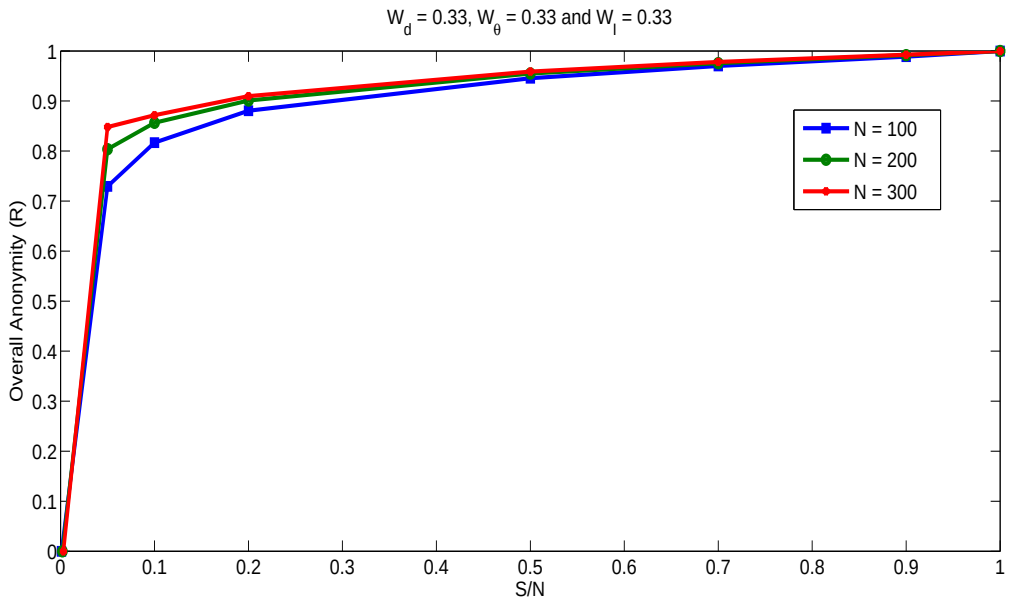

Figure 6.6 – S/N vs Overall anonymity (R).

It can be observed that the behavior of plot is similar for different values of N, which shows that overall anonymity R can be interpreted as a non-decreasing function of S/N. Thus in onion routing system, S/N is more important in determining the level of anonymity provided, than S or N individually.

6.3 Mixes

The architecture of mix network includes proxy servers between the sender and the receiver. The proxy servers receive data messages from group of users and forward the messages on behalf of the user, avoiding any correlation between received messages and dispatched messages.

A passive attack model is considered where, the attacker can listen to the communication in the system, but cannot control any of the systems. Consider a system with N users, i.e., the size of anonymity set is N. The attacker after performing an attack, assigns p probability to a set of users of size C to be initiator of a message, and 1-p to the rest of the users ($N - C$). Since the attacker does not have any additional information, p is assigned equally to C users and (1-p) is assigned equally to ($N - C$) users.

$$p_i = \frac{p}{C} \; \forall i \in \{1, 2, ..., C\} \quad \text{and} \quad p_i = \frac{1-p}{N-C} \; \forall i \in \{C+1, ..., N\}$$

From the above distribution, maximum anonymity is achieved when equal probability ($\frac{1}{N}$) is assigned to all the N users in the system, i.e.

$$\frac{p}{C} = \frac{1-p}{N-C} = \frac{1}{N}$$

i.e., $C = N.p$. Consider $p = 0.5$ and $N = 500$. In this scenario, maximum overall anonymity is achieved when C is 250.

Figure 6.7 depicts the behavior of overall anonymity R with C for different values of p and when $N = 500$, weights $W_d = 0.33$, $W_\theta = 0.33$ and $W_I = 0.33$. Consider the case where $p = 0.2$. Here, when $C = 100$ equal probability ($\frac{1}{500}$) is assigned to all the 500 users in the system. When $p = 0.5$ and for $C = 250$ equal probability ($\frac{1}{500}$) is assigned to all the 500 users in the system. From the above cases, it can be summarized that the overall anonymity R depends on C and p. For increased values of p, the max value of overall anonymity R is achieved at higher values of C.

Figure 6.8 depicts the behavior of overall anonymity R with C for different weights and when $N = 500$, $p = 0.5$. Consider the scenario where equal probability is assigned to all the users in the system (d = 1, θ = 0 and I = 0). In this scenario, the maximum overall anonymity R is achieved irrespective of weights assigned. Therefore, weights does not matter much when the value of overall anonymity R is considerably high. From the plot, it can be observed that for values of C between 50 and 450, the overall anonymity R is considerably high due to this, the overall anonymity R has similar pattern for different values of weights.

In this particular scenario (Figure 6.9), the probability distribution when $p = 0.4$ and $C = 100$ leads to same probability distribution as the case when $p = 0.6$ and $C = 400$. The probability distribution when $p = 0.4$ and $C = 100$ is given by,

$$p_i = \frac{0.4}{100} \; \forall i \in \{1, 2, ..., 100\} \quad \text{and} \quad p_i = \frac{0.6}{400} \; \forall i \in \{101, ..., 500\}$$

which is same as the distribution when $p = 0.6$ and $C = 400$, given by,

$$p_i = \frac{0.6}{400} \; \forall i \in \{1, 2, ..., 400\} \quad \text{and} \quad p_i = \frac{0.4}{100} \; \forall i \in \{401, ..., 500\}$$

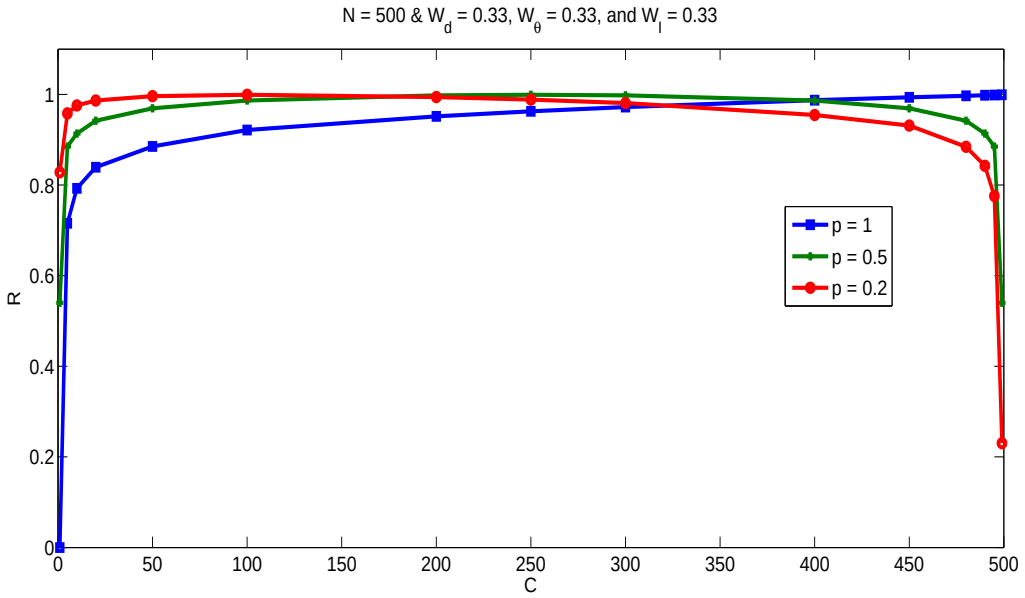

Figure 6.7 – Overall anonymity (R) vs C.

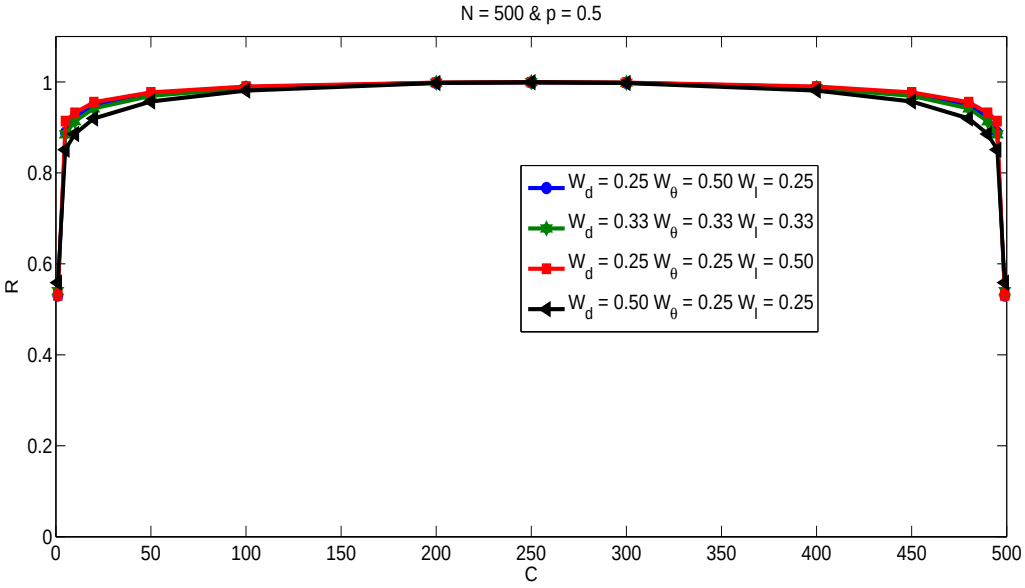

Figure 6.8 – Overall anonymity (R) vs C.

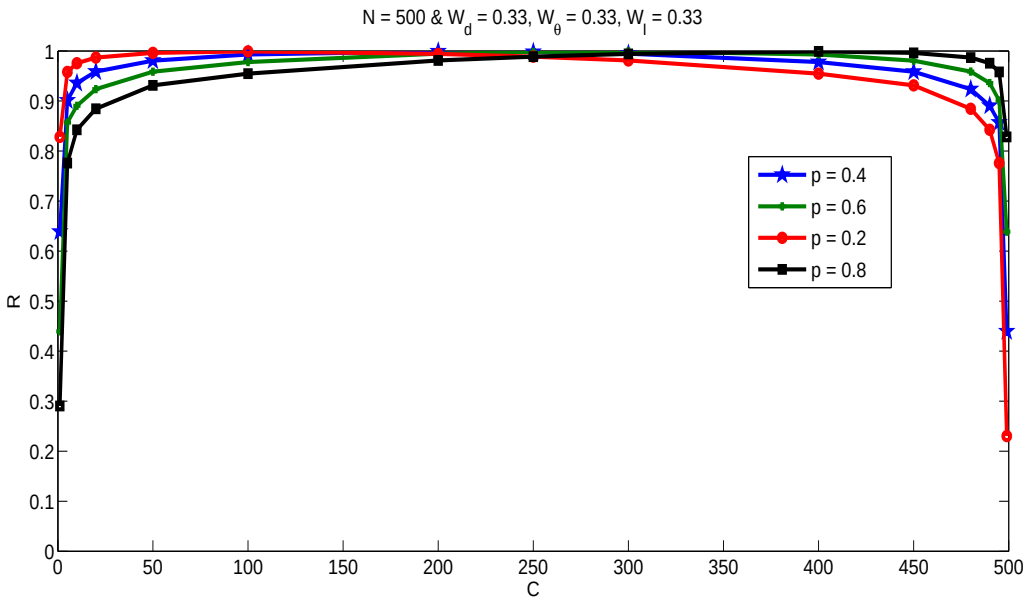

Figure 6.9 – Overall anonymity (R) vs C.

In all the figures 6.7, 6.8 and 6.9, as long as $50 \leq C \leq 450$, overall anonymity R is sufficiently large for all values of P, because for the values of C between [50, 450], the Isolation factor (I) is almost negligible, so do the local anonymity (θ) and the value of normalized entropy (d) is large resulting in higher values of overall anonymity (R).

CHAPTER 7

CONCLUSION

This thesis summarizes the need of Anonymity to secure the user identity from threats included over WWW and outlines different metrics used to measure the level of anonymity provided by an anonymous system. It also provides the drawbacks of existing metrics in measuring the overall anonymity provided by an anonymous system. This thesis proposes a new isolation measure, based upon presence of outliers in a distribution, which is critical towards quantifying the overall anonymity of a system. It also proposes a three-dimensional approach towards measuring sender anonymity. The proposed three-dimensional metric adresses the drawback of existing metrics in capturing the extent of isolation in a system. The proposed metric is applied to existing anonymous systems - crowds, mixes and onion routing. The justification for three distinct aspects of the proposed 3-tuple metric is provided. This thesis also provides interpretation of the metric in terms of attributes desired in the system.

In this thesis, the attack model cosiders a scenario where the attacker tries to identify the initiator of a single data message among a group of potential senders. Future work includes extending the attack model to multiple message scenario. Future work also includes work that shows two systems characterized by same 3-tuple metric provide exactly the same degree of anonymity, which would help to further justify the proposed approach. It would suffice to show that if $d_1 = d_2$, $\theta_1 = \theta_2$, and $I_1 = I_2$ then $D_1 = D_2$. This would also help close the search for examples where the metric's behavior seems to be counterintuitive. Another direction of future research includes using the knowledge of attacker's location (or information gained) to redesign the anonymous system, so as to maximize the system's degree of anonymity.

REFERENCES

REFERENCES

[1] S. Garfinkel and G. Spafford, *Web Security, Privacy & Commerce*, Second Ed. O'Reilly Media, 2001.

[2] R. Geambasu, T. Kohno, A. Levy and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," in *Proc. 18th USENIX Security Symposium*, Montreal, Canada, Aug. 2009, pp. 299-315.

[3] L. A. Gordon, M. P. Loeb and T. Sohail, "A framework for using insurance for cyber-risk management," *Communications of the ACM*, vol. 46, pp. 81-85, Mar. 2003.

[4] M. Shao, Y. Yang, S. Zhu and G. Cao, "Towards statistically strong source anonymity for sensor networks, in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, May 2008, pp. 51-55.

[5] J. Kong, X. Hong and M. Gerla, "An identity-free and on-demand routing scheme against anonymity threats in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 888-902, Aug. 2007.

[6] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *Proc. ACM Workshop on Privacy in the electronic society*, Alexandria, VA, USA, 2005, pp. 71-80.

[7] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proc. 30th IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2009, pp. 173-187.

[8] C. Diaz, C. Troncoso and A. Serjantov, "On the impact of social network profiling on anonymity," in *Proc. 8th International Symposium on Privacy Enhancing Technologies (PETS)*, Leuven, Belgium, Jul. 2008, pp. 44-62.

[9] B. Krishnamurthy and C. E. Wills, "On the leakage of personally identifiable information via online social networks," *ACM SIGCOMM Computer Communication Review*, vol. 40, pp. 112-117, Jan. 2010.

[10] D. L. Chaum, "Untraceable electronic mail, return addresses and digital pseudonyms," *Communications of the ACM*, vol. 24, pp. 84-90, Feb. 1981.

[11] T. Primepq. *Mix Network*, URL: http://en.wikipedia.org/wiki/Mix_network [cited 13 December 2008].

[12] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proc. 2nd International Conference on Privacy Enhancing Technologies (PETS)*, San Francisco, CA, USA, 2002, pp. 41-53.

[13] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, pp. 66-92, Nov. 1998.

[14] C. Diaz, S. Seys, J. Claessens and B. Preneel, "Towards measuring anonymity," in *Proc. 2nd International Conference on Privacy Enhancing Technologies (PETS)*, San Francisco, CA, USA, 2002, pp. 54-68.

[15] M. G. Reed, P. F. Syverson and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 482-494, May 1998.

[16] R. Dingledine, N. Mathewson and P. Syverson, "Tor: The secondgeneration onion router," in *Proc. 13th USENIX Security Symposium*, Aug. 2004.

[17] N. Jaggi, U. M. Reddy and R. Bagai,"A Three-Dimensional Approach Towards Measuring Sender Anonymity," *Submitted for publication*, July 2010.

[18] A. Pfitzmann and M. Hansen, "Anonymity, unobservability and pseudonymity: A consolidated proposal for terminology," Draft, Jul. 2000.

[19] J. Feigenbaum, A. Johnson and P. Syverson, "Probabilistic analysis of onion routing in a black-box model," in *Proc. ACM workshop on Privacy in electronic society*, Alexandria, VA, USA, 2007, pp. 1-10.

[20] G. T'oth, Z. Horn'ak and F. Vajda, "Measuring anonymity revisited," in *Proc. 9th Nordic Workshop on Secure IT Systems*, Espoo, Finland, Nov. 2004, pp. 85-90.

[21] S. Claub and S. Schiffner, "Structuring anonymity metrics," in *Proc. 2nd ACM Workshop on Digital identity management*, Alexandria, VA, USA, 2006, pp. 55-62.

[22] M. Edman, F. Sivrikaya and B. Yener, "A combinatorial approach to measuring anonymity," in *Proc. IEEE International Conference on Intelligence and Security Informatics (ISI 2007)*, New Brunswick, NJ, USA, May 2007, pp. 356-363.

[23] B. Gierlichs, C. Troncoso, C. Diaz, B. Preneel and I. Verbauwhede, "Revisiting a combinatorial approach toward measuring anonymity," in *Proc. 7th ACM workshop on Privacy in the electronic society*, Alexandria, VA, USA, 2008, pp. 111-116.

[24] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. New York, NY, USA: John Wiley & Sons, Inc., 1987.

[25] B. Peirce, "Criterion for the rejection of doubtful observations," *Astronomical Journal*, vol. II, pp. 161-163, Jul. 1852.

[26] B. A. Gould, "On Peirce's criterion for the rejection of doubtful observations, with tables for facilitating its application," *Astronomical Journal*, vol. IV, pp. 81-87, Apr. 1855.

[27] S. Ross, "Peirce's criterion for the elimination of suspect experimental data," *Journal of Engineering Technology*, vol. 20, 2003.

APPENDIX

APPENDIX

CODE IN 'C' TO COMPUTE THE 3-TUPLE METRIC GIVEN A PROBABILITY DISTRIBUTION. THIS INCLUDES COMPUTING THE NUMBER OF OUTLIERS IN THE DISTRIBUTION.

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
int main()
{
    int n=0,count,k=0,i=0,vl,z,l,flag,flag2=0,dummyn,count2;
    float temp=0.0,frac=0.0,total=0.0,phi=0.0,deg=0.0,dev=0.0,mean=0.0;
    float sdev=0.00,logr,logr2=9.3,logqn,logrn,loglr,loglml2,x,x2;
    float logx2m1,sum1,sum2,qn,omean,logl,vlog[n],finalx,threshold;
    float dsum=0.00,outlier[n],sum,*d,*non_outliers,a[600];

    /*These values are from the table in [26] that relates log R to x.
      a[100] gives the value of log R when x=1.00*/

    a[100]=9.501485,a[101]=9.499209,a[102]=9.496941,a[103]=9.494682;
    a[104]=9.492431,a[105]=9.490188,a[106]=9.487954,a[107]=9.485729;
    a[108]=9.483512,a[109]=9.481303,a[110]=9.479102,a[111]=9.476910;
    a[112]=9.474725,a[113]=9.472549,a[114]=9.470380,a[115]=9.468220;
    a[116]=9.466067,a[117]=9.463923,a[118]=9.461786,a[119]=9.459657;
    a[120]=9.457536,a[121]=9.455422,a[122]=9.453316,a[123]=9.451218;
    a[124]=9.449127,a[125]=9.447044,a[126]=9.444968,a[127]=9.442900;
    a[128]=9.440839,a[129]=9.438785,a[130]=9.436739,a[131]=9.434700;
    a[132]=9.432669,a[133]=9.430644,a[134]=9.428627,a[135]=9.426616;
```

```
a[136]=9.424613,a[137]=9.422617,a[138]=9.420627,a[139]=9.418645;
a[140]=9.416670,a[141]=9.414701,a[142]=9.412739,a[143]=9.419785;
a[144]=9.408836,a[145]=9.406895,a[146]=9.404960,a[147]=9.403032;
a[148]=9.401111,a[149]=9.399196,a[150]=9.397287,a[151]=9.395385;
a[152]=9.393490,a[153]=9.391601,a[154]=9.389718,a[155]=9.387842;
a[156]=9.385972,a[157]=9.384108,a[158]=9.382251,a[159]=9.380400;
a[160]=9.378555,a[161]=9.376716,a[162]=9.374883,a[163]=9.373056;
a[164]=9.371236,a[165]=9.369421,a[166]=9.367612,a[167]=9.365809;
a[168]=9.364013,a[169]=9.362222,a[170]=9.360437,a[171]=9.358657;
a[172]=9.356884,a[173]=9.355116,a[174]=9.353354,a[175]=9.351598;
a[176]=9.349847,a[177]=9.348102,a[178]=9.346363,a[179]=9.344629;
a[180]=9.342901,a[181]=9.341178,a[182]=9.339461,a[183]=9.337749;
a[184]=9.336042,a[185]=9.334341,a[186]=9.332646,a[187]=9.330956;
a[188]=9.329271,a[189]=9.327591,a[190]=9.325917,a[191]=9.324247;
a[192]=9.322583,a[193]=9.320925,a[194]=9.319271,a[195]=9.317622;
a[196]=9.315979,a[197]=9.314340,a[198]=9.312707,a[199]=9.311078;
a[200]=9.309455,a[201]=9.307837,a[202]=9.306223,a[203]=9.304615;
a[204]=9.303011,a[205]=9.301413,a[206]=9.299819,a[207]=9.298229;
a[208]=9.296645,a[209]=9.295065,a[210]=9.293491,a[211]=9.291920;
a[212]=9.290355,a[213]=9.288794,a[214]=9.287238,a[215]=9.285686;
a[216]=9.284139,a[217]=9.282597,a[218]=9.281059,a[219]=9.279526;
a[220]=9.277998,a[221]=9.276474,a[222]=9.274954,a[223]=9.273438;
a[224]=9.271927,a[225]=9.270420,a[226]=9.268917,a[227]=9.267419;
a[228]=9.265925,a[229]=9.264437,a[230]=9.262952,a[231]=9.261472;
a[232]=9.359996,a[233]=9.258524,a[234]=9.257057,a[235]=9.255593;
a[236]=9.254133,a[237]=9.252678,a[238]=9.251226,a[239]=9.249779;
a[240]=9.248335,a[241]=9.246895,a[242]=9.245459,a[243]=9.244027;
```

```
a[244]=9.242600,a[245]=9.241176,a[246]=9.239757,a[247]=9.238342;
a[248]=9.236931,a[249]=9.235524,a[250]=9.234121,a[251]=9.232721;
a[252]=9.231326,a[253]=9.229935,a[254]=9.228548,a[255]=9.227164;
a[256]=9.225784,a[257]=9.224406,a[258]=9.223032,a[259]=9.221662;
a[260]=9.220296,a[261]=9.218933,a[262]=9.217575,a[263]=9.216219;
a[264]=9.214868,a[265]=9.213520,a[266]=9.212176,a[267]=9.210836;
a[268]=9.209499,a[269]=9.208166,a[270]=9.206837,a[271]=9.205511;
a[272]=9.204188,a[273]=9.202868,a[274]=9.201551,a[275]=9.200239;
a[276]=9.198929,a[277]=9.197625,a[278]=9.196324,a[279]=9.195027;
a[280]=9.193733,a[281]=9.192443,a[282]=9.191157,a[283]=9.189874;
a[284]=9.188595,a[285]=9.187319,a[286]=9.186046,a[287]=9.184777;
a[288]=9.183511,a[289]=9.182253,a[290]=9.180998,a[291]=9.179746;
a[292]=9.178498,a[293]=9.177252,a[294]=9.176009,a[295]=9.174770;
a[296]=9.173533,a[297]=9.172300,a[298]=9.171069,a[299]=9.169842;
a[300]=9.168617;
printf("THIS CODE HELPS YOU TO FIND LOCAL ANONYMITY, GLOBAL
        ANONYMITY AND ISOLATION FACTOR ALONG WITH OUTLIERS FOR
        A GIVEN PROBABILITY DISTRIBUTION");

printf("\n\n\nENTER THE NUMBER OF USERS:");
scanf("%d",&n);
d = malloc(n*sizeof(float));
for(i=0;i<n;i++)
    d[i]=0;
printf("\nENTER THE DISTRIBUTION ENTER '2' IF YOU WANT TO
        DISTRIBUTE REMAINING PROBABiLITY TO REST OF THE USERS:");
```

```
//*******READING DISTRIBUTION*******//

//At any point of reading distribution, If the input is 2, the
  remaining probability is distributed equally to N users.

while(temp!=2.00)
{
    dsum=dsum+temp;
    scanf("%f",&temp);
    if(temp!=2.00)
    {
        d[k] = temp;
        k = k+1;
    }
    if(k == n)
        break;
}

//Checking for correctness of distribution

if (dsum>1.0)
{
    printf("\nError! Sum of entered distribution is greater than 1");
    exit(0);
}
if (k==n && dsum<1.0)
{
```

```
    printf("\nError! Sum of entered distribution is less than 1");
    exit(0);
}
frac=(1-dsum)/(n-k);
while(temp == 2)
{
    d[k]=frac;
    k=k+1;
    if(k == n)
        break;
}

//****PRINTING DISTRIBUTION****//

for(i=0;i<n;i++)
{
    if (d[i]<=0.0)    //This will assign 0.0 to -0.0 to avoid errors.
        d[i]=0.0;
    printf("%f\t",d[i]);
}

//*******FINDING LOCAL ANONYMITY*******//

temp = 0;
for(i=0;i<n;i++)
{
    if(d[i]>=temp)
```

```
        temp=d[i];    //Maximum of P_i
}
phi = temp;
printf("\n\nLOCAL ANONYMITY = %f",phi);

//*******FINDING GLOBAL ANONYMITY*******//

temp = 0;
total = 0;
for(i=0;i<n;i++)
{
    if(d[i]!=0.0)
        total = total + (d[i]*log(d[i]));    //Pi*logPi
}
total = total * (-1);    //-(pi*log pi)
deg=total/log(n);
printf("\n\nGLOBAL AMONYMITY = %f",deg);

//*******DETERMINING SAMPLE DEVIATION OF SYSTEM*******//

sdev = 0.00;
mean = (1.0/n);
total = 0;
for(i=0;i<n;i++)
    total = total + pow((d[i]-mean),2);

//sdev is first part of numerator in Eq:4.4.6
```

```
sdev = pow(total,0.5);    //sdev=sum of (pi - mean)^2

total = total/(n-1);
dev = pow(total,0.5);    //dev = sample deviation of distribution

//printf("\n\nSAMPLE DEVIATION = %f",dev);

//*******IDENTIFYING OUTLIERS*******//

dummyn=n;
if(n>120)
    n=120;
if(dev!=0)    //dev=0 means P_i is 1/N for all users (deviation = 0).
{
    //Loop continues till outliers detected when
                                    i=K and i=K-1 are same.
    for(i=1;i<n;i++)
    {
        sum=0.0;
        vl=0,l=0;
        flag=0;
        while(1)
        {
            //Outlier Detection Algorithm considering
              natural logarithms

            //Initial assumption of log R = 9.3
```

```
logr=logr2;
x2=x;

//Q^N of Equation 4.3.2 is computed here
qn = pow(i,i)*pow((n-i),(n-i))/pow(n,n);

//log Q^N is calculated here
logqn = log10(qn)+10.0;

//log R^N is calculated here
logrn = (i*(logr-10.0))+10.0;

//Lambda^(N-n) is calculated as per Equation 4.3.3
loglr = logqn - logrn + 10.0;

//log(Lambda) is calculated here
logl = ((loglr-10)/(n-i))+10.0;

//log(1-Lambda^2) is calculated here to use in Eq:4.3.4
loglml2 = log10(1.00-pow(pow(10,(logl-10)),2))+10.0;

//log(x^2-1) of Equation 4.3.4 is calculated here
logx2m1 = log10(((float)n-(float)i)/((float)i))
                                    +loglml2-10.00;

//x of Equation 4.3.4 is calculated here
x = pow(pow(10,logx2m1)+1,0.5);
```

```
    /*new log R is obtained here referring to the table
      (Eq:4.3.5). From array a[] in the code*/
    logr2=a[(int)(100*x)];

    //Keeping track of log R's in the array vlog[]
    vlog[vl]=logr2;

    // Checking current log R with log R of
                        previous iteration.

    if(vlog[vl]==vlog[vl-1])
        break;
    vl++;
}
finalx=x2;

//Computing allowable error value
threshold=finalx*dev;

for(z=0;z<n;z++)
{
    //Checking the error value of each user
    if((d[z]-mean)>=threshold)
    {
        outlier[l]=d[z];
        l++;
```

```
            flag++;
        }
    }

    //Checking # Outliers when i=K and i=K-1
    if(flag==flag2)
        break;
    flag2=flag;
}
printf("\n");
if(flag==0)
{
    printf("NO OUTLIERS, ISOLATION FACTOR = 0\n\n");
    exit(0);
}

//****PRINTING OUTLIERS****//

for(i=0;i<flag;i++)
    printf("\nOUTLIER%d = %f",i+1,outlier[i]);
printf("\n\n");

//****DETERMINING ISOLATION FACTOR****//
non_outliers = malloc((dummyn-flag)*sizeof(float));
sum2 = 0;
for(i=0;i<dummyn-flag;i++)
    non_outliers[i]=0;
```

```
for(i=0;i<flag;i++)
    sum1=sum1+outlier[i];    //sum1 is sum of outliers
omean=(1-sum1)/(dummyn-flag);    //omean is mean of non-outliers

//non_outliers[i]=distribution w out outliers

count2=0;

//with this for loop, each element of distribution is selected.
for(i=0;i<dummyn;i++)
{
    count=0;

    //This loop checks wether the element is an outlier
    for(z=0;z<flag;z++)
    {
        if(d[i]==outlier[z])
            count++;
    }

    //If the element is not outlier, it is placed in non_outlier[]
    if(count==0)
    {
        non_outliers[count2]=d[i];
        count2++;
    }
}
```

```
        //Calculating Isolation Factor

        //Second part of numerator in Eq:4.4.6 is calculated here
        for(i=0;i<(dummyn-flag);i++)
            sum2 = sum2 + pow((non_outliers[i]-omean),2);
        sum2=pow(sum2,0.5);

        //Isolation factor is calculated as per Eq:4.4.6
        sum=(sdev-sum2)/(float)(flag);

        printf("ISOLATION FACTOR = %f\n\n",sum);
        free(d);
        free(non_outliers);
    }
    else
        printf("\n\nNO OUTLIERS, ISOLATION FACTOR = 0\n\n");
return 0;
}
```