

5-2014

Privacy Preserving Statistics

Oluwakemi Hambolu

Clemson University, yemhal@yahoo.co.uk

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses



Part of the [Computer Engineering Commons](#)

Recommended Citation

Hambolu, Oluwakemi, "Privacy Preserving Statistics" (2014). *All Theses*. 1985.

https://tigerprints.clemson.edu/all_theses/1985

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

PRIVACY PRESERVING STATISTICS

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

by
Oluwakemi Hambolu
May 2014

Accepted by:
Dr. Richard R. Brooks, Committee Chair
Dr. Kuang-Ching Wang
Dr. Adam Hoover

Abstract

Over the past few years, there have been an increase in the development and improvement of circumvention tools like Tor and Psiphon. These tools provide an environment for citizens of oppressive regimes to access websites freely without fear of identification, these tools aid democracy activists and journalists in West Africa in using the Internet securely. A similar circumvention tool was developed by us. This tool circumvents DNS and IP address blocking/filtering, by leveraging technologies developed by criminal botnet enterprises.

To improve and maintain the circumvention tool we developed, it is important to quantify the number and country of origin of users. System statistics are used to give feedback to the US State Department, who funded this project. We need to show them that target users are taking advantage of the developed system. Considering that the system helps provide anonymity to users as well as bypassing DNS and IP filtering, and system users have a high demand for privacy, we must not collect sensitive user information. We therefore develop statistics that aim to not compromise user anonymity.

Two probabilistic data structures are introduced, evaluated, improved upon and used, to keep system statistics without compromising user privacy. The first data structure is the negative survey. Using negative survey we can keep an aggregate count of user countries of origin without knowing the country of origin of any individual session by asking the user to report a country that they do not belong to. Negative survey allows

us to calculate how many accesses there have been from each country, while keeping insensitive user information. The second data structure is a probabilistic counting algorithm which, without keeping a list of already encountered data, like IP addresses, estimates the number of distinct elements in a large collection of data. We use hash values to obtain the number of unique users of the system. This algorithm is based on statistical observations made on bits of hashed values of records. Our records contain the hash values of the users' SSL certificates. We store the least significant bit that was set in the SSL certificate hash. From the bit position of the lowest bit that is not set, we get a good estimate of the number of system users. We contribute to this technique by considering when the number of collisions of the hash values will affect the estimate and use this amount to give a better estimate. This also allows us to decide on-line the proper register size to maintain.

Dedication

This thesis is dedicated to God, who has brought me thus far, my family and my fiancée Femi. I love you all.

Acknowledgments

First of all, I want to thank my advisor, Dr. Richard R. Brooks. This dissertation could not have been done without your insightful directions, helpful suggestions, incredible patience and a lot of your valuable time. Secondly, I would like to thank Dr. Adam Hoover and Dr. Kuang-Ching Wang for being my committee members Thirdly, I want to appreciate the help and cooperation of the students in my research group. It has been a pleasure to work with these talented and helpful students. I also want to thank Femi, my Bible study group and my family for your prayers and support. No matter what happens, you are always there to give me endless support. Your understanding and support gives me the courage and wisdom to overcome any difficulties in the past and the future. Last but not the least, this material is based upon work supported by, or in part by, US Department of State award number S-LMAQM-12-GR-1033.

Table of Contents

Title Page	i
Abstract	ii
Dedication	iv
Acknowledgments	v
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Problem Domain	2
1.2 Application	3
1.3 Negative Survey	5
1.4 Probabilistic Counting	5
1.5 Organization	6
2 Related Work	7
2.1 Tor Case Study Discussion	7
2.2 Negative Survey	9
2.3 Probabilistic Counting	9
2.4 Summary	10
3 Research Design and Methods	11
3.1 Negative Survey	11
3.2 Probabilistic Counting Algorithm	13
3.3 Summary	24
4 Results	25
4.1 Anonymity Analysis	25
4.2 Performance Analysis	26
4.3 Conclusion of Results	28

5	Conclusions and Discussion	31
5.1	Summary	31
5.2	Conclusion	32
5.3	Future Work	32
	Appendices	34
A	Cirriculum Vitae	35
	Bibliography	38

List of Tables

- 3.1 The number of inputs that gives at least 50% probability of having a collision 20

List of Figures

1.1	Censored network and secure USB	3
3.1	Pattern hash mapping	14
3.2	Random hash mapping	15
3.3	The mapping of 20000 users to the bit position in a bitmap	17
3.4	Cardinality calculated without implementing the birthday paradox	23
3.5	Cardinality calculated after implementing the birthday paradox	23
4.1	Cardinality calculated with the number of collisions that causes an over-counting	27
4.2	Estimate proposed by Flajolet and Martin compared with the estimate+collision and over counting for register length 16 bits	29
4.3	Estimate proposed by Flajolet and Martin compared with the estimate+collision and over counting for register length 17 bits	29
4.4	Estimate proposed by Flajolet and Martin compared with the estimate+collision and over counting for register length 18 bits	30

Chapter 1

Introduction

West Africa is known to be held back by poverty, political instability and poor human rights records. With censorship technologies becoming widely and readily available, and with the fact that it has been observed that Israel and France are providing censorship technologies to West African governments [5, 16], it becomes apparent that the suppression of a free press in the region includes Internet censorship. Many democracy activists and journalists in West Africa are technically unsophisticated. When repressive local governments start to take full advantage these Internet surveillance technologies, the results are likely to be catastrophic for reform movements and journalists in the region. Hence, the activist community is concerned about their digital security. They are aware that the computers and Internet which they rely on are becoming insecure. They are also aware that technical threats will increase in number and sophistication.

Circumvention and anonymization tools allow the activist community of a country to speak freely without fear of persecution. With the increasing need for these tools, research has gone into the development and improvement of more sophisticated and user friendly tools. Maintaining a system statistics without compromising user anonymity is a difficult problem. The Tor project, a group committed to a free and open Internet,

currently stores lists of unique IP addresses on their node bridges for at least 24 hours uptime so that they can collect more accurate statistical data [18]. Even though these IP addresses are never transmitted, the very storage of unique IP addresses poses a security risk for system users. Solutions are therefore needed to offer privacy guarantees during the collection of the system's statistics.

1.1 Problem Domain

1.1.1 Political Threat

As West African governments become aware of the impact of social media tools, they will increase network surveillance and track IP addresses. Surveillance tools are in place in some West African countries like Cote d'Ivoire to help the police track Internet criminals. In December 20th, 2010, in Cote d'Ivoire, there was unrest following Laurent Gbagbo's refusal to step down after losing Presidential Elections, Globalvoicesonline.org translated French tweets [20], one of which read, @Sanders225 To those who tweet from their computers I hope you have installed software to hide your IP address #civ2010. This tweet is evidence of the growth in the level of sophistication on the part of West African Internet users and their desire for anonymity.

There have been other examples of Internet communication being affected, SMS messaging blocked on election day until results were officially announced, to hinder local activists from reporting what they had seen in polling booths. It is important to provide a system for activists to ensure that threats to Internet freedom, such as those seen in Iran, will not occur in West Africa, as Internet penetration and network surveillance reaches the same levels as other parts of the world.

Many countries have laws restricting access to pornography. Ghana and Nigeria



Figure 1.1: Censored network and secure USB

have laws in place that allow government authorities to demand network use information with little or no due process. In Sub-Saharan Africa, Internet censorship currently consists primarily of the sporadic blocking of access to IP addresses [15]. Rather than using technical measures, most on-line censorship currently involves the arrest and incarceration of bloggers, activists, and citizen journalists [15].

1.1.2 Technical Threat

Technically literate Internet users have realized that anonymity is difficult, if not impossible, to achieve. As Scott McNealy of Sun Microsystems said, “You already have zero privacy, Get over it” [3]. With current events, like uprisings in the Middle East, the importance of on-line media in promoting democracy has become apparent. Unfortunately, repressive governments, like Iran, Tunisia, and China, have found on-line social media useful for identifying activists [19]

1.2 Application

To counter on-line censorship and media suppression, we developed and packaged a distributed proxy system, which is used to bypass network censorship technologies by democracy advocates and citizen journalists of West Africa. See figure 1.1 We created

secure USB drives, whose contents are encrypted to avoid incriminating their owners. The drives can be booted directly, or run as plug and play with encrypted archives, as long as the user has the passphrase. A number of useful tools for anonymous communications allow users to access Internet resources with reasonable anonymity guarantees. Censorship systems in countries like China keep users from accessing anonymous communications tools by both filtering the Domain Name System (DNS) records that map names to IP addresses and blocking access to IP addresses associated with those services. The USB drives include technologies that effectively circumvent these mechanisms. Automated access to anonymization systems will be provided, and users will also be able to access other Internet services directly. The tool created circumvents DNS filtering and IP address blocking by leveraging technologies developed by criminal botnet enterprises. Botnets are distributed systems built from criminally compromised PCs. They are central to spam, phishing, pharming, and identity theft scams. The criminal gangs running botnets use fast-flux technologies to avoid detection and remain anonymous. Connections change their DNS names weekly. Fast-flux technologies change the IP addresses these DNS names map to every few minutes. This ruse has made it impossible for western law enforcement agencies to find botnet operators. We used the same approach for creating a distributed proxy system. Running the USB drives, other nodes where access to anonymization tools is available will relay messages to users whose access is blocked, to aid them in getting around DNS filtering and IP address blocking. Advocates and journalists use this to circumvent censorship without putting themselves at risk.

The US State Department sponsored this project there is a need to provide reliable statistics of system use, to show that people are taking advantage of the technology developed, and money spent on this project is being put to good use. The method of collecting the system statistics has to be secure and anonymous to prevent identifying users. In the event that a repressive government may gain access to the system database,

we do not want them to find identifiable information linking a user to the system. To accomplish this, we use two data structures with strong guarantees of privacy to track the number of unique users and their country of origin.

1.3 Negative Survey

To track users, service providers query location information directly. The problem with this method is that it reduces user anonymity. To solve this problem will be to use reverse auctions also known as negative survey, by creating an aggregate frequency distribution of category membership without collecting specific information from the users. In a negative survey, the service provider offers the user a list of mutually exclusive groups. The user is asked to select one or more categories that they do not belong to. In this manner, negative survey stores a negative representation of the data. In [14], a sensor network base station uses negative samples to reconstruct a histogram of the original sensor reading. We use this method to track aggregate statistics of the country of origin of system users.

1.4 Probabilistic Counting

We use a probabilistic counting algorithm originally introduced by Flajolet and Martin [10], to estimate the number of unique system users, given a good hash function [10]. We can take an arbitrary set of data, hash it and use hash values to form a collection of binary strings storing the position of the least significant bit in the binary representation of the hashed values. The pattern observed indicates the number of distinct values in data [10] by using the position of the leftmost zero in the string as an indicator of $\log_2 n$ where n is the total number of data in the set.

This result is not a great estimator [10], because it gives at best a power of 2 in estimating the number of distinct elements. To improve the estimate, multiple independent hash functions are averaged. This method is expensive because hashing is expensive. Stochastic averaging can use a single hash function instead of multiple independent ones, and split output into many buckets [10]. This thesis, looks at our contribution to the method in [10]. We considered the effect the birthday paradox has on the hashed values and how collisions affect the final estimate. This results in the estimate undercounting the number of users. We account for this effect and propose a solution.

1.5 Organization

The outline of this thesis is as follows: Chapter 1 describes the motivation behind this thesis, exploring the issues and challenges with anonymous. It discusses the application area, the distributed proxy system we developed, and how we will incorporate anonymous statistics. This chapter also introduced the data structures used to implement anonymous statistic.

Chapter 2 discusses previous work on anonymous statistics, and how it applies to our work, it also discusses how the Tor network reports the number of users. Chapter 3 gives the background. It introduces information needed to understand our work, by discussing the data structures used in greater detail and the logic used to provide anonymity. It includes verification done to obtain a good hash function and our contributions to probabilistic counting procedure. Chapter 4 presents the analysis of these data structures. This analysis looked at the anonymity the overall system presents as well as its performance. Chapter 5 summarizes our technique with conclusions based on what we have learned, future work is presented in Section 5.3.

Chapter 2

Related Work

The need to provide privacy in statistics collection is a growing research area. There are various methods implemented in systems especially for surveys, to collect statistics without leaking compromising information. For example, **Differential privacy** a new research area that uses a mathematical rigor in privacy preserving analysis of data, helps in releasing statistics without compromising the privacy of the individuals [4], it ensures that a limited amount of risk is involved in participating in a database. This is because Differential privacy does not guarantee absolute privacy [4].

To contribute to this growing research area, we use a mixed method approach by using two probabilistic data structures (reverse auction and hash functions). With this, we hope to achieve a method of obtaining the statistic metrics of our developed system while preserving privacy. The following sections discuss related work on the two data structures.

2.1 Tor Case Study Discussion

The Tor network is widely used by hundreds of thousands of users, and it makes it possible for these users to hide their location when they access various online services.

Therefore, it is expected that sensitive information are not used in obtaining the usage statistics of the Tor network. [13] describes the approaches used by the Tor network to collect aggregate usage data and how the number of the system users were estimated from this data. Tor uses two approaches to estimate the number of users [13]; the first approach was used to get the number of new and returning users while the second approach was used to estimated the number recurring users. The difference between these two is based on the design they used in their estimates which is based on Tor's directory protocol. As of 2010, there were eight of such directory authorities [13]. Seven out of the eight report the number of directory request they are answering every day. The method proposed in [13] was to weight the request seen by a single directory authority with the expected fraction of the request that the directory authority should see. The assumption was that a directory authority sees 1 out of 6 requests. [13] multiplied the reported requests by 6 to obtain an estimate count of the new or returning daily users. There were a few difficulties with this approach which was based on some uncertainties.

The second approach was used to estimate the number of recurring daily users, and they based the approach on the fact that each client will be required to refresh its network information every few hours to make indistinguishable path selection decisions. To estimated the number of users, they count the number of network status requests on a fast directory mirror. They estimate how many network status requests there are in a network by dividing the local request number by the share of requests that a directory mirror thinks it should see. This estimate is flawed because of outliers and missing values [13].

An alternate approach proposed was to count the number of daily unique IP addresses on a relay that sees most of the clients at least once a day [13]. Directory mirror keeps the observed client IP addresses in memory for at most 24 hours and report the absolute number of distinct addresses and then discard the IP addresses [13].

The current approach used by the Tor network [13], is to have the Tor bridge

report the number of unique IP addresses they see everyday. They sum up the unique IP addresses per day and interpret the result as the estimated number of users with at least 24 hours uptime.

We do not want to use the approach used by the Tor network and store the IP addresses that access our system because, we do not want any identifiable information stored in our database even if it is for 24 hours. We believe that our system user can be identified if the IP address is stored.

2.2 Negative Survey

Negative surveys are closely related to randomized response techniques (RRTs), because both aim at conducting private surveys rely on a randomizing device to answer questions in the survey [6]. RRTs uses this to choose among questions while in negative survey respondents choose among answers [6]. In [7], several formulas needed to generate a negative database were compared. Negative representations of data have also been expanded to negative surveys. The application of a negative survey to sensor networks where the negative survey stores membership information is in [14], the base station wants to know how many nodes belong to different categories. Each node is aware of the possible categories and transmits some category to which it does not belong back to the base station. This is similar to the work described in this thesis, in that each user will be aware of the different countries and they report a country they do not belong to.

2.3 Probabilistic Counting

Probabilistic counting is similar to a bloom filter [1]. Bloom filters are probabilistic data structures that give a compact approximation on set membership [1], like the

approach used in this thesis, bloom filters are also performance/memory efficient, which makes them appealing in a variety of scenarios, from web caching [8] to key reuse in sensor networks [2]. The hash map data structure is similar to the bloom filter because both store membership information in a bit vector, the difference is in their functionality. The bloom filter tries to determine whether an item is in the set or not while probabilistic counting is used to determine the number of members in the set. Both structures have no false negatives but some false positives.

2.4 Summary

In this Chapter we discuss related work in privacy preserving statistics. We incorporate the method in [14] to obtain the country of origin of a user who gained access to the proxy system is from and with the probabilistic counting, we can estimate the total number of unique users. We propose combining these two methods to provide anonymous statistics.

Chapter 3

Research Design and Methods

3.1 Negative Survey

Negative database is a database that contains counterfeit data along with actual data [21]. Negative databases have been expanded to negative survey. Negative survey is used in this thesis to determine the aggregate number of accesses from a country of origin while keeping no incriminating information.

Negative survey creates an aggregate frequency distribution of category membership without collecting specific information from users. It is used for conducting surveys that is mindful of participant's privacy [6]. It allows users to keep the name of their country undisclosed by asking them, instead, questions about their country name. An example a question asked is as follows:

Choose 1, I **am not** from:

☐ Chad

☐ Ghana

☐ Ivory Coast

☐ Senegal

If a user is from Chad, the user will randomly select one of the last three options. With this method, aggregate user country statistics can be calculated without users directly disclosing this information [6]. A user will belong to only one these countries and we are interested in estimating the proportions of the users that positively belong to each of the countries. We estimate this value. Let $p_{i,j}(1 \leq i, j \leq t)$ be the probability that option X_i is chosen given that the user belongs to X_j and $\sum_i p_{i,j} = 1$. Let π_i denote the proportion of the total users that positively belong to category i with $\sum_i \pi_i = 1$. The probability of selecting X_i is given by:

$$\lambda_i = \sum_j p_{i,j} \pi_j \quad (3.1)$$

[6]

From Equation 3.1 and the fact that $\sum_i \pi_i = 1$, an unbiased estimator of π_i is given by:

$$\hat{\pi}_i = 1 - (t - 1) \hat{\lambda}_i$$

where $\hat{\lambda}_i = n_i/n$ is an unbiased estimator for λ_i . With Equation 3.1, a user provided with a fair, $t-1$ sided, die can select a country by privately obtaining a value m , and choosing the m^{th} true option from the top, skipping over the false category.

The relationship between anonymity and accuracy will be quantified with respect to the probability that a user can be correctly identified. This relationship is developed from the number of countries (C), the number of users (S), and the number of samples each user provides (\bar{X}_s). The probability that the user is a selected country is denoted by $Pr(X_i = T | \bar{X}_s = F)$ where X_i is the country in question and \bar{X}_s is the set of negative responses related to a user's latest entry.

Proof. Using Bayes' rule,

$$\begin{aligned}
Pr(X_c = T | \bar{X}_i = F) &= \frac{Pr(X_i = F | \bar{X}_c = T)}{Pr(\bar{X}_i = F)} \\
Pr(X_c = T) &= \frac{1}{C} \\
Pr(\bar{X}_i = F) &= \left(\frac{1}{C}\right)^S \\
Pr(X_i = F | \bar{X}_c = T) &= \left(\frac{1}{C-1}\right)^S \\
\bullet \bullet \frac{Pr(X_i = F | \bar{X}_c = T)}{Pr(\bar{X}_i = F)} &= \frac{\left(\frac{1}{C-1}\right)^S * \left(\frac{1}{C}\right)}{\left(\frac{1}{C}\right)^S} \\
&= \frac{T^{S-1}}{(C-1)^S} \quad \square
\end{aligned}$$

We automate this process, having the system automatically report a country other than the one access is from.

3.2 Probabilistic Counting Algorithm

Probabilistic counting first introduced by Flajolet and Martin is used to give a calculated estimate of the number of unique users that access our system. The algorithm is probabilistic in nature because it is based on a hash function of the data it operates on.

3.2.1 Hash Function

Hash functions appeared in 1950 as tools for hash tables. In 1969 Bloom filters were used to test set membership. In 1977 Carter & Wegan published a paper called *Universal Hashing*, where they considered hash function as probabilistic objects. People became inspired to use hash functions to transform inserted data to random variables or sequences of random bits if considered from a discrete point of view. Hash functions take

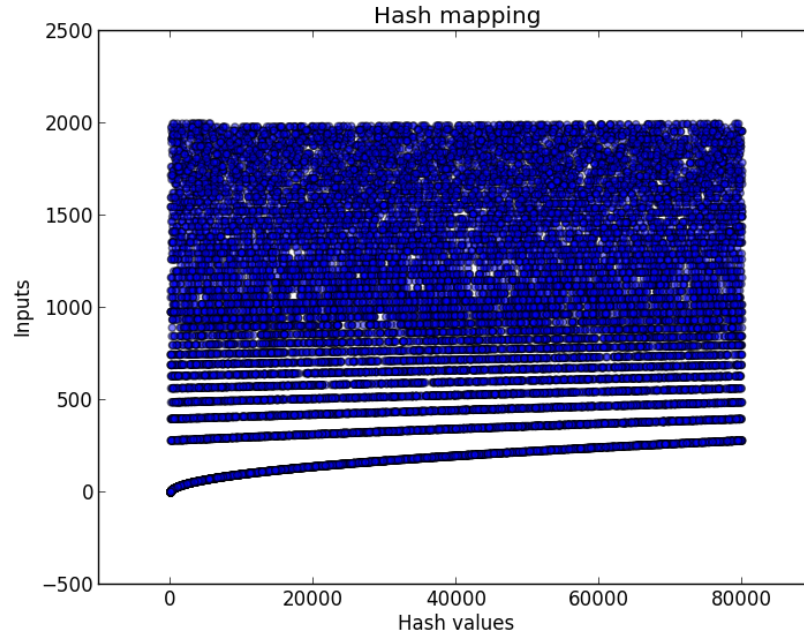


Figure 3.1: Pattern hash mapping

elements from the domain of interest and map them into strings of 0's and 1's such that each bit has the same probability to be 0 and the same probability to be 1. They transform data into independent and identically distributed (i.i.d) uniform random variables. Hash functions also map data of arbitrary length to data of a fixed length. To make our system anonymous, we require a hash function that does a random mapping, where no pattern is observed to make it easily detectable.

While looking for a suitable hash function that meet our requirement of randomness, we looked at different hash functions. The main consideration was the pattern in the hash mapping, for example, Figure 3.1 gives an observable pattern in the mapping of a hash function $h(k) = k \bmod m$. We want our hash function to give a random mapping to the space with no pattern as shown in Figure 3.2.

A simple multiplication hash function [11] was chosen to be used in the algorithm

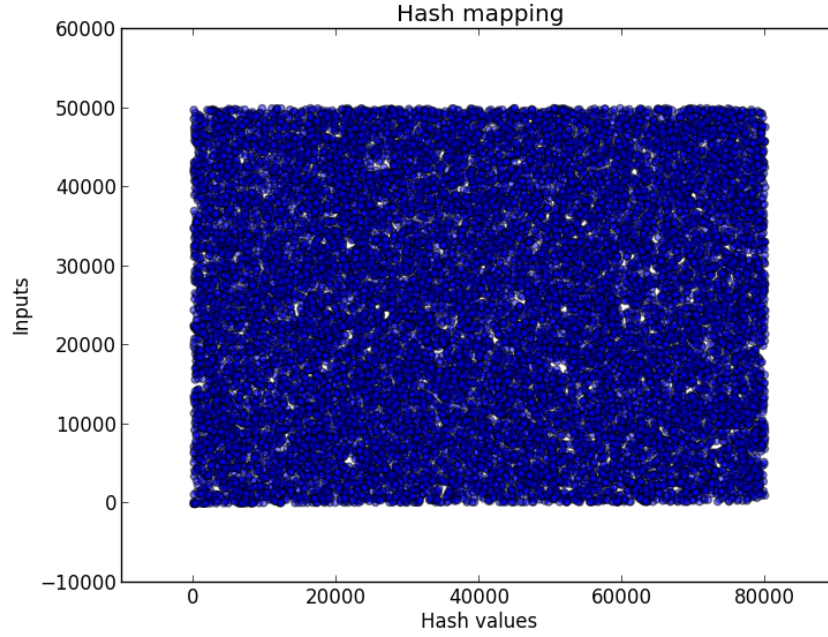


Figure 3.2: Random hash mapping

and it defined as follows:

Let k be the input, A = a random real number, r = range of the hash function and $M = 0.5 \times (\sqrt{5} - 1)$

$$s = k * A$$

$$x = r * \text{fractional part of } s$$

$$h(k) = \text{floor}(m * x)$$

[11]

This hash function gives the random mapping of Figure 3.2

3.2.2 Probabilistic Counting Procedure

Probabilistic counting [10], works as follows. Given a function $h(x)$ of range $[0 \dots 2^L - 1]$ that transforms records $x \in [0, \dots, N - 1]$ into integers uniformly distributed

over the range L , where L is a set of *binary strings*. Let $lsb(y)$ represent the position of the least significant 1 – *bit* value of the binary representation of y . So an element in the record x , will be mapped to $lsb(h(x))$. As discussed in 3.2.1, the hash function will transform the records into a string of random bits. It is observed that if the hash values $h(x)$ are uniformly distribute, it is expected to see a sequence of 10xxx... with probability $1/2$, a sequence of 110xxx... with probability $1/4$, a sequence of 1110xxx... with probability $1/8$. With the strings being uniform, it can be generalized that a sequence $1^k0...$, appears with probability of $1/2^{(k+1)}$.

The idea of the algorithm is to record this occurrences in a vector **bitmap**[0...L-1], that is, maintain a **bitmap** vector of L bit initialized to *zero*, and for each element that we hash, find the $lsb(h(x))$ set the corresponding position on the **bitmap**. We keep track of the prefixes $1^k0...$. The **bitmap** depends on the set of hashed values and not on the frequency of the x . If an element occurs more than once, it will be counted once. We expect that if d is the number of distinct elements in M , then **bitmap**[0] is accessed about $n/2$ times approximately, **bitmap**[1] is accessed approximately $n/4$ times [10]. For example, let us assume that there are $n = 20000$ distinct elements, Figure 3.3 shows that exactly 10000 elements are mapped to bit position 0 that is $\frac{d}{2}$ maps to **bitmap**[0], the figure also shows that 5000 elements are mapped to bit position 1, that it $\frac{d}{4}$ maps to **bitmap**[1]. It follows therefore that $\frac{d}{k}$ maps to **bitmap**[k-1] [12].

It was proposed in [10] to use the position of the leftmost zero in the **bitmap** with rank starting at 0 as an indicator of $\log_2 n$. Let P be this value. Based on the assumption of hash values being uniformly distributed, the expected value of P will be close to :

$$E(P) \approx \log_2 n$$

We therefore estimate the cardinality with 2^p , where p is the size of the largest prefix in the bitmap. If the **bitmap** is $1^k0...$, then $p = k + 1$.

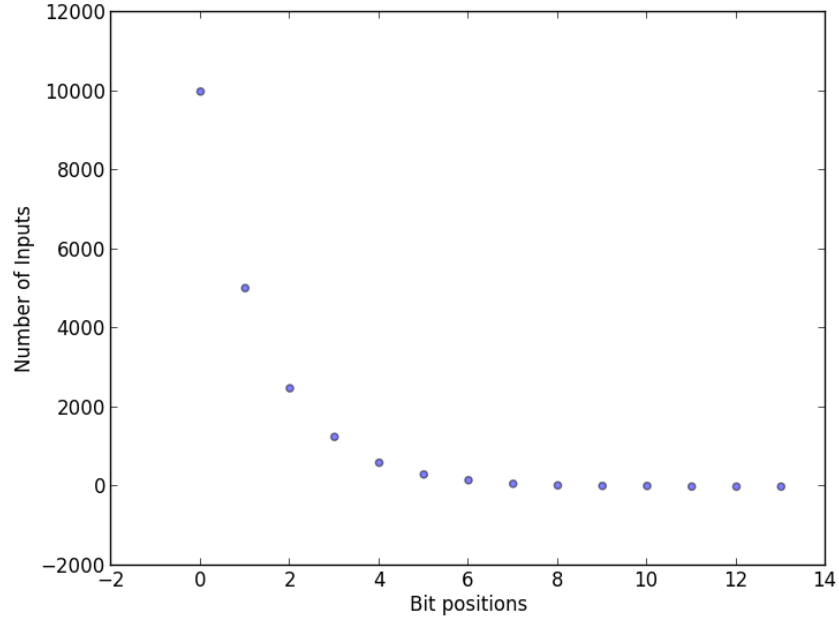


Figure 3.3: The mapping of 20000 users to the bit position in a **bitmap**

This works but there is a small bias, the values estimated are slightly off ($\mathbb{E}[2^p] \neq n$). In [9]Philippe showed that, with mathematical analysis, we can find the exact bias estimator known as the *magic constant*. [10] proposed to apply this estimator as a simple correction and have an unbiased estimator. That is $\mathbb{E}[2^p] \approx n\phi$. Where $\phi \approx 0.77351$ and is defined by:

$$\phi = \frac{(e^\gamma)\sqrt{2}}{3} \prod_{p=1}^{\infty} \left[\frac{(4p+1)(4p+2)}{(4p)(4p+3)} \right]^{\varepsilon(p)}$$

The cardinality can be calculated by:

$$\begin{aligned} \mathbf{Z} &= \frac{1}{\phi} 2^p \\ \mathbf{E}[Z] &= n \end{aligned} \tag{3.2}$$

Typically, estimates were found to be one binary order of magnitude off the exact result. The idea proposed to improve the accuracy of the algorithm was to use m different hash

functions and a different **bitmap** table for each hash function and then take the average. The accuracy should be increased by $1/\sqrt{m}$. This unfortunately is very costly, because hashing is expensive. They found an alternative, to use a technique called *stochastic averaging*. The idea consists of using the hashing function in order to distribute each element into one of m substreams randomly computed as $\alpha = h(x) \bmod m$. Only the corresponding **bitmap** vector of address α will be updated with the information contained in $h(x)$, ie $h(x) \bmod m \equiv \lfloor h(x)/m \rfloor$. To compute the cardinality, the position of the leftmost zero of each of the bitmap p_i will be calculated and then calculate the average of all the bitmaps $A = \frac{p_1 + p_2 + \dots + p_m}{m}$. The Cardinality will be equal $(1 - \phi)^A$. This method in [10] was called the algorithm Probabilistic Counting with Stochastic Averaging. As a future work we hope to further investigate if this algorithm will have a better performance if we consider storing both the least significant bit and the most significant bit of the hashed values while keeping track of the m bitmaps.

3.2.3 Birthday Paradox in anonymous statistics

We look into the effect that the birthday paradox will have on probabilistic counting, because the algorithm uses hash functions and there are likely to be collisions occurring during the mapping. There will be error introduced into the calculation when we have more than one element mapping to the same bit position in the **bitmap**. The accuracy of our method depends on how often this happens. We quantified the number of collisions that occur and add this value to the estimated cardinality. We investigated the length of the **bitmap** L and when to increase the length. We hope to improve the accuracy of the system to make it robust and flexible.

3.2.3.1 Birthday Paradox

When considering the birthday paradox, we are interested in the event, A that among n people, each having a particular day of the year as a birthday, at least two of them will have the same birthday. We assumed that each day is equally likely for each person. In other words, the probability that a person born on day r will be $\frac{1}{k}$, where k is 365.

Proof. This probability is one minus the probability that the n birthdays are distinct [22]. Given as:

$$P(A) = 1 - P(\bar{A}) \quad (3.3)$$

$$= 1 - \frac{k}{k} \cdot \frac{k-1}{k} \cdots \frac{k-n+1}{k} \quad (3.4)$$

$$= 1 - \frac{kP_n}{k^n} \quad (3.5)$$

□

The probability of A passes 50% when n exceeds 23.

We need to calculate the minimum number of inputs that will cause a collision. A collision happens when an element is mapped to a slot that already stores an element. If $k = 2^p$, and n is number of elements in the sequence, given a certain sample space, what will be the minimum number of input that will have a 50% probability of having a collision.

Using the formula of birthday paradox on our system, and with different bit lengths, we calculated the minimum number of inputs to our system that will have a 50% probability of having a collision Table 3.1 shows the findings

From Table 3.1, we can see that with a bit length of 16 we only need 320 people to access our system to cause at least 50% probability of having a collision. It becomes

Bitmap	Input	Probability
16	320	0.54
17	450	0.53
18	650	0.55
19	900	0.53
32	80000	0.53

Table 3.1: The number of inputs that gives at least 50% probability of having a collision

necessary to know when the number of collisions in the system will start affecting our result and cause us to undercount the estimated cardinality.

3.2.3.2 Impact of the Birthday Problem

Having one collision will not affect the efficiency of the result. It is important to find out when the number of collisions become large enough that it begins to affect the result causing a leveling of. To determine the number of collision in hashing [17] proposed a way of finding the expected number of collision in hashing.

3.2.3.3 Expected Number of Collisions

If we have n elements mapped to any one of k slots, to determine the expected number of collisions, we determine the expected number of elements that will be mapped to the same slot and the number of empty slots. This will lead to the expected number of collisions. Note: all slots have equal probability.

Proof. To calculate the expected number of elements mapped to the same slot, let us consider the random variable,

$$\mathbf{x}_i = \begin{cases} 1 & \text{if element } i \text{ is mapped to slot } 1; \\ 0 & \text{otherwise.} \end{cases}$$

The number of items that will be mapped to slot 1 is:

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2 + \dots + \mathbf{X}_n$$

[17]

the expected value of $\mathbf{X}_i = \frac{1}{k}$

••• the expected number mapped to slot 1 is:

$$\begin{aligned} \mathbf{E}(X) &= \sum_{i=1}^n \mathbf{E}(X)_i \\ &= \frac{n}{k} \end{aligned}$$

[17]

□

To calculate the expected number of empty slots, lets consider the probability that slot j remains empty after all the n elements have been mapped as $(1 - \frac{1}{k})^n$. The expected number is as follows:

Proof. Define a random variable as follows:

$$\mathbf{X}_j = \begin{cases} 1 & \text{if slot } j \text{ remains empty;} \\ 0 & \text{otherwise.} \end{cases}$$

The number of empty slots: $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2 + \dots + \mathbf{X}_k$ [17]

••• the expected number of empty slot is:

$$\begin{aligned}\mathbf{E}(X) &= \sum_{i=1}^k \mathbf{E}(X)_i \\ &= k(1 - \frac{1}{k})^n \quad [17]\end{aligned}$$

□

We now calculate the number of collisions from the number of empty slots. Taking the number of empty slots as X , we know that $K - X$ elements are hashed without collision and $N - K + X$ are mapped with collisions.

Proof. If Z is the expected number of collisions, we have:

$$\begin{aligned}Z &= n - k + \mathbf{E}(X) \\ &= n - k + k(1 - \frac{1}{k})^n \quad ??\end{aligned}$$

[17]

□

3.2.3.4 Application of the Number of Collisions

We applied the expected number of collisions found in Section 3.2.3.3 to our estimated cardinality and we found out that it resulted in a better estimate of the cardinality. Figure 3.4 and Figure 3.5 shows this result. The bit length used in the figures is 16 and we used different number of inputs ranging from 10000 to 100000.

Figure 3.4 shows that at some point (30000), the estimated number begins to level off because the register size is too small to accommodate all the inputs, when we consider the effect of the birthday paradox and compensated for the collisions, we found according to Figure 3.5, that we get a better estimate.

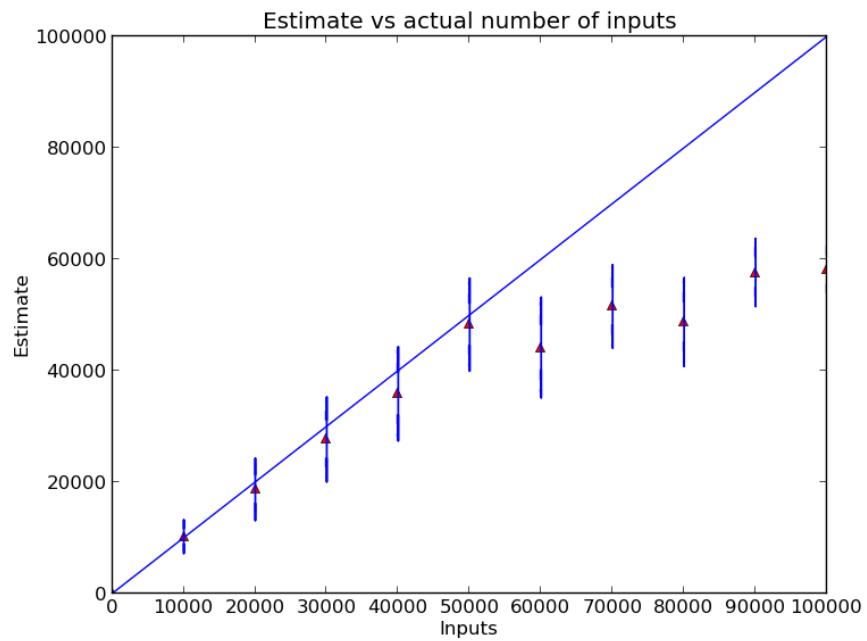


Figure 3.4: Cardinality calculated without implementing the birthday paradox

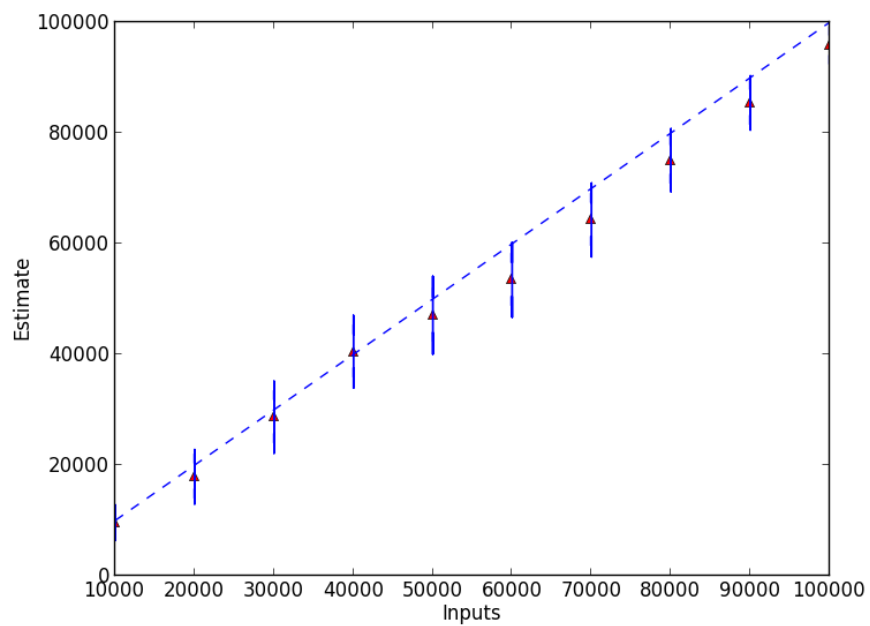


Figure 3.5: Cardinality calculated after implementing the birthday paradox

We therefore conclude that the estimated number of users can be calculated by:

$$E = \frac{1}{\phi} 2^p + (n - k + k(1 - \frac{1}{k})^n)$$

where E = number of users,

K = length of the bitmap and

N = number of actual users

3.3 Summary

This chapter looked at the essential background information of this thesis. We identified two data structures used to develop a statistics metric of our distributed proxy system. The data structures are used to collect statistics of the system while preserving privacy. Section 3.1 discussed the reverse auctions approach method. While Section 3.2 talked about the probabilistic counting structure was used to estimate the number of unique users of the system. Lastly, we looked into the effect the number of collisions of the hashed values will make on the final estimated cardinality and we compensated for that amount.

Chapter 4

Results

The two data structures used in this thesis increase the privacy of the distributed proxy system we developed, by preserving the privacy of the system users while collecting statistics. This chapter analyzes the anonymity and performance provided by the statistic metrics developed in Chapter 3.

4.1 Anonymity Analysis

4.1.1 Probabilistic Counting

4.1.1.1 False Positive

The question we want to answer in this section is, What happens to the user anonymity if a repressive government gains access to our database, the hash function used, and a user's certificate. Will the government be able to associate the user with our system? It is important to calculate the probability of detecting that the user is infact associated to the system. It is obvious that we will never have false negatives, because if the user has used the system, the least significant bit position of the record hashed value

will be set in the corresponding position of the **bitmap**. It means that if the position in the bitmap is not set, then it is acceptable to conclude that the user did not use the system. However, if the position is set, it is possible that it was set by a different user, hence it indicates false positive. To calculate the probability of a false positive : From Chapter 3 Section 3.2, it was established that the probability that bit n is set is $\frac{1}{2^{(n+1)}}$, the probability that the bit was not set is therefore: $1 - \frac{1}{2^{(n+1)}}$.

Therefore the probability of false positive is given by:

$$1 - \left[e^{-\frac{1}{2^{(n+1)}}} \right] \quad (4.1)$$

The higher this probability is, the higher the anonymity of the system will be.

4.2 Performance Analysis

4.2.1 Probabilistic Counting

4.2.1.1 Effect of Over and Under Counting

In Chapter 3 Section 3.2.2 we showed that we get a good estimate if we add the number of collisions. It is beneficial to find out if we have the best estimate. If Chapter 3 Section 3.2.2 gives us the correct estimated number of users of our system, it is possible that there will be some collisions that will occur. We will like to quantify this amount. To solve this, we took the calculated estimate from Chapter 3 Section 3.2.2 as the number of input and used that to calculate the number of collisions that will occur. Figure 4.1 shows the result.

Figure 4.1 shows that there will be an over counting in the final result of our estimate if we implement this.

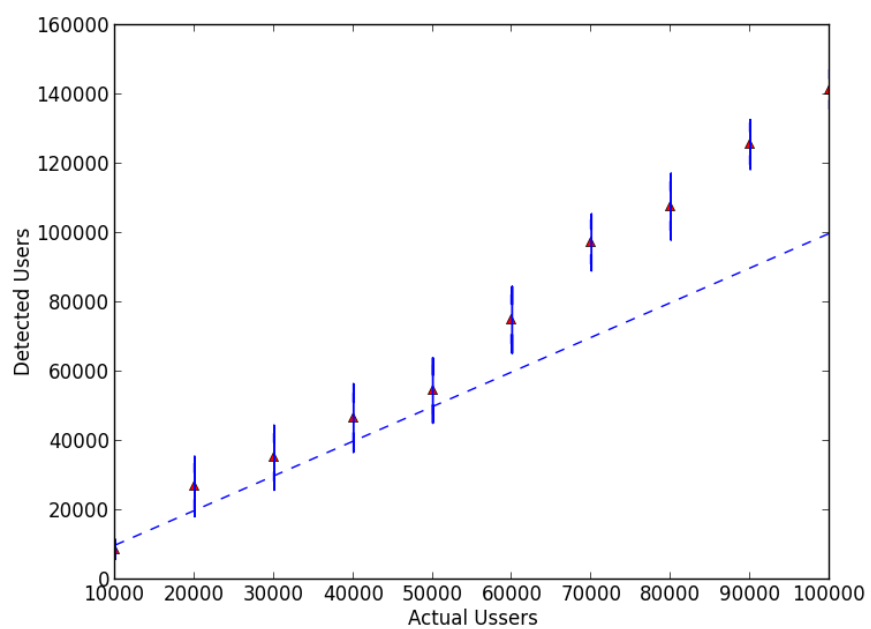


Figure 4.1: Cardinality calculated with the number of collisions that causes an overcounting

4.2.1.2 Register change

For our statistics metrics to give a good result, it needs to be flexible and it should accommodate as many users as possible. To achieve flexibility, we need to change the register length at a certain point and know when to implement the change. It is useful to know what cardinality can be successfully calculated with each register length. Figure 4.2 shows the mapping for register bit length 16, the figure shows that when the better estimate is taken as the actual number of inputs, and the number of collisions brought by this estimate is added to it to produce an even better estimate, this will result in an over counting of the cardinality and the over counting increases as the number of actual inputs increases. We suggest changing the size of the **bitmap** by 1 bit, when the range between the original probabilistic counting procedure and the overestimated result grows really large. Figure 4.3 and Figure 4.4 show the result of bit length 17 and 18 respectively. For length 17 according to Figure 4.3, we see at about 80000, the range between the over estimate and the probabilistic counting procedure grows larger, so it is advised to change the bit length to 18 bits. For length 18 according to Figure 4.4, we see at about 160000, the range between the over estimate and the probabilistic counting procedure grows larger, so it is advised to change the bit length to 19 bits. We propose that this will make our system more robust.

4.3 Conclusion of Results

This chapter looked at the performance and anonymity analysis that our probabilistic counting procedure in privacy preserving statistics. If a repressive government or attacker gets a hold of the whole database, there will be no exact incriminating information in the database that can be used to successfully identify individual users.

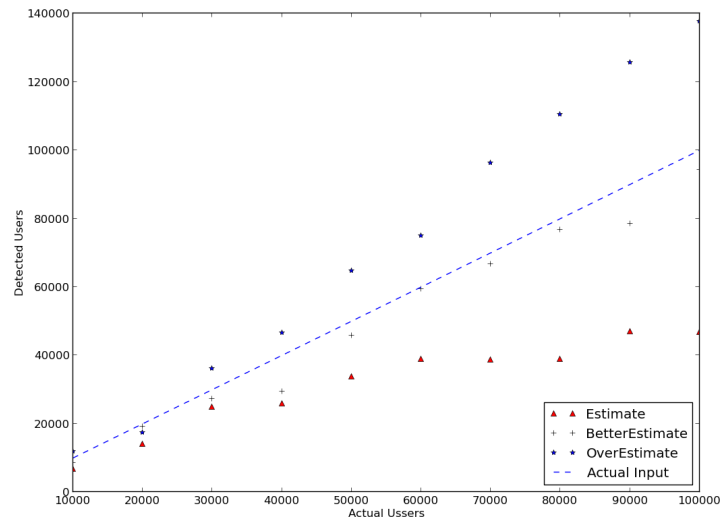


Figure 4.2: Estimate proposed by Flajolet and Martin compared with the estimate+collision and over counting for register length 16 bits

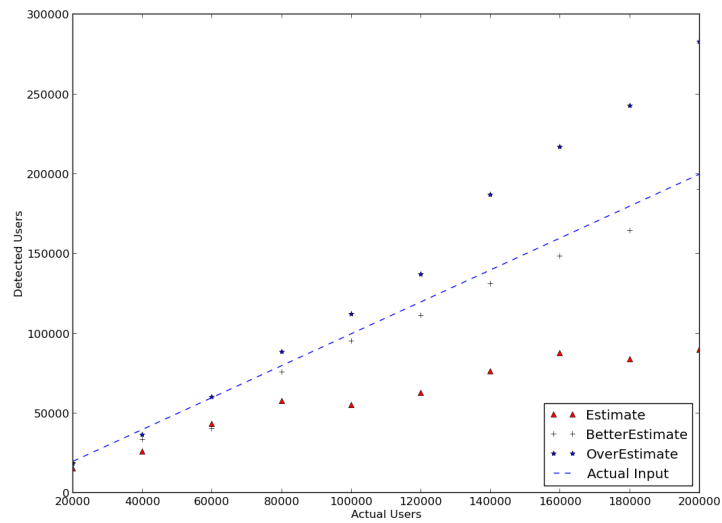


Figure 4.3: Estimate proposed by Flajolet and Martin compared with the estimate+collision and over counting for register length 17 bits

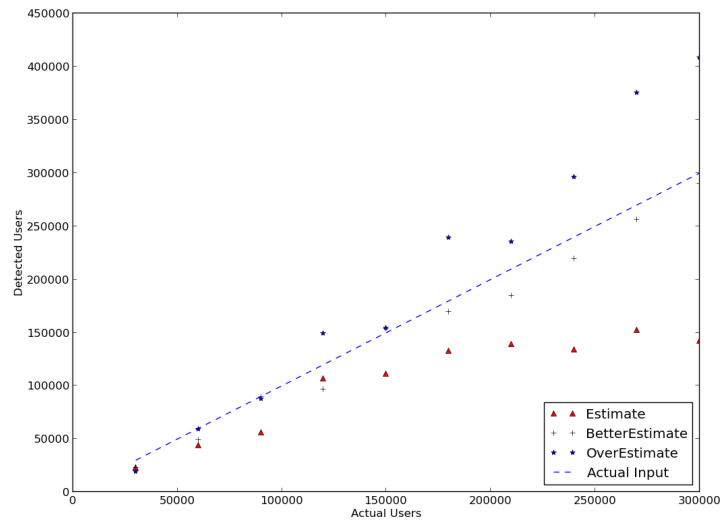


Figure 4.4: Estimate proposed by Flajolet and Martin compared with the estimate+collision and over counting for register length 18 bits

Chapter 5

Conclusions and Discussion

5.1 Summary

The work done in privacy preserving statistics looks at a way of getting the statistic of the circumvention system developed by us, without compromising the privacy of the users. Two probabilistic data structures were used to develop system statistics while preserving privacy. Chapter 3 looks into the design and implementation of these data structures. Chapter 3 Section 3.1 discussed reverse auctions. With each access a user gains to our system, the system will automatically report a country other than the one the user is from. These results can be used to estimate how many users came from each country. In Chapter 3 Section 3.2, a probabilistic counting structure was used to estimate the number of unique users of the system. Each user will need a certificate to get authenticated with the system. A hash will be taken of each certificate and the least significant bit that was set will be stored in a list known as **bitmap**. A register does a logical OR of those bits. The position of the leftmost zero on the **bitmap** is used to calculate a good estimate of the number of unique users of the system. Chapter 3.2.3 looked at the addition we made to improved the probabilistic counting procedure to give

a better estimate in the cardinality by considering the effect that the number of collisions will make on the calculated cardinality, which results in undercounting of the distinct number of users.

5.2 Conclusion

We propose combining the two data structures used in this thesis, we will be able to get necessary information needed for statistics collection while preserving the identity of the user.

5.3 Future Work

For the future work, we propose investigating and quantifying in greater detail, how much information about the system users, can be leaked from one bit on the **bitmap** in the probabilistic counting structure and develop a method to prevent or minimize this amount. We also hope to further investigate if the stochastic averaging algorithm will give a better performance if we use both the least significant bit and the most significant bit to account for the cardinality.

With the negative survey, we get anonymity for individual users by keeping a record of the negative responses provided by these users. Given the total database, this anonymity might decrease. For instance, if we have 3 countries of reference; Chad, Ghana and Benin and it is found that 100 people accessed the system. If the histogram of the responses provided show us that 65 people claim not to be from Chad and 35 people claim not to be long to Ghana. It can be logical to conclude that most of the people that used the system are most likely to be from Benin. We will like to further study how much information can be leaked by the negative survey. Finally after verifying the anonymity

provided by the two data structures, we will be able to quantify in greater detail how good our statistics metrics is.

Appendices

Appendix A Cirriculum Vitae

Oluwakemi Hambolu

Phone: 404-643-5146

Email: ohambol@clemson.edu

310 Riggs Hall, Clemson University

Clemson, SC 29634

Expertise: Network and computer security, Probability and Statistics **Academic**

M.S. Computer Engineering May 2014 Clemson University, Clemson, SC GPA: 3.85/4.00

B.Eng. Electrical Engineering January 2010 Ahmadu Bello University, Zaria GPA: 3.4/4.00

Proficiencies

- Programming languages: Python MATLAB
- Operating Systems, Platforms: Windows, Apple, and Linux OS
- Other Tools: Latex

CAREER HISTORY

Clemson University, ECE Department Clemson, SC August 2012 to Present

Research Assistant

- Conducting research under Dr. Richard R. Brooks in computer and network security.
- Developed a distributed proxy system using Python and batch scripts that adopts DNS technologies to over come censorship based on DNS and IP filtering
- Developing a secure boot solution (software environment) for people to access the internet securely from a USB on insecure host using the distributed proxy

- Improving network security and network performance evaluation skills
- Using knowledge discovery from data streams to create a privacy preserving usage analysis statistics. Implemented this method to get an anonymous statistics of the number of users of the distributed proxy system we have developed
- Conducted research independently, analyzed data, and documented and published the research methods and results.

Guaranty Trust Bank, Kano, Nigeria August 2010 to July 2011

Account Officer

- Advertised the products of the bank to customers
- Opened and maintained bank accounts for customers
- Developed a good customer relation skill

Quanteq Technology Services, Abuja, Nigeria August 2008 to December

2008

Engineering Trainee

- Improved IT skills in carrying out computer systems maintenance and repairs
- Provided ICT network set-up/troubleshooting and repairs
- Developed good networking skills

Other Experiences

- Clemson LeaderShape 2012 (leadership training) May 2012
- Public Relation Officer for International Student Association. Clemson University
January 2013 to August 2013

- Senator for Computer Engineering in Clemson University Graduate Student Government August 2013 to Present

Publications and Presentations

- R.R.Brooks, O. Hambolu, Y. Fu, P. Marusich, S. Balachandran, "Creating a Tailored Trustworthy Space for Democracy Advocates using Hostile Host" accepted as an extended abstract and to be published in the proceedings of the 8th annual workshop on CSIIR(2013)

Bibliography

- [1] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, July 1970.
- [2] R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir. On the detection of clones in sensor networks using random key predistribution. *Trans. Sys. Man Cyber Part C*, 37(6):1246–1258, November 2007.
- [3] R. R. Brooks. *Disruptive Security Technologies with Mobile Code and Peer-to-Peer Networks*. CRC Press, Boca Raton, 2005.
- [4] Cynthia Dwork. Differential privacy: A survey of results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, TAMC’08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.
- [5] Ogala Emmanue. Exclusive: Jonathan awards £40million contract to israeli company to monitor computer, internet communication by nigerians,. [urlhttp://www.premiumtimesng.com/news/131249-exclusive-jonathan-awards-40million-contract-to-israeli-company-to-monitor-computer-internet-communication-by-nigerians.html](http://www.premiumtimesng.com/news/131249-exclusive-jonathan-awards-40million-contract-to-israeli-company-to-monitor-computer-internet-communication-by-nigerians.html), 2011.
- [6] F. Esponda. Negative Surveys. *ArXiv Mathematics e-prints*, August 2006.
- [7] F. Esponda. Everything that is not important: Negative databases [research frontier]. *Comp. Intell. Mag.*, 3(2):60–63, May 2008.
- [8] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: A scalable wide-area web cache sharing protocol. *IEEE/ACM Trans. Netw.*, 8(3):281–293, June 2000.
- [9] S. Finch. *Mathematical Constants*. Cambridge University Press, 2003.
- [10] Philippe Flajolet, G. N. Martin, and G. Nigel Martin. Probabilistic counting algorithms for data base applications, 1985.
- [11] Margaret Fleck. Hash functions. <http://www.cs.hmc.edu/~geoff/classes/hmc.cs070.200101/homework10/hashfuncs.html>, 2000.

- [12] Joao Gama. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 1st edition, 2010.
- [13] S. Hahn and K. Loesing. Privacy-preserving ways to estimate the number of tor users. Technical Report 2010-11-30, The Tor Project, November 2010.
- [14] James Horey, Michael M. Groat, Stephanie Forrest, and Fernando Esponda. Anonymous data collection in sensor networks. *Mobile and Ubiquitous Systems, Annual International Conference on*, 0:1–8, 2007.
- [15] Open Net Initiative. Internet filtering in sub-saharan africa, 2009.
- [16] Internet. <http://www.africaintelligence.com/LCE/business/2012/07/12/cofrexport-equips-benin-s-president,104311032-BRE>.
- [17] Internet. Probability in hashing. <https://www.cs.duke.edu/courses/cps102/spring09/Lectures/L-18.pdf>.
- [18] Karsten Loesing. Counting daily bridge users. Technical Report 2012-10-001, The Tor Project, October 2012.
- [19] E. Morozov. *The Net Delusion: The Dark Side of Internet Freedom*. Foreign Affairs Press, 2011.
- [20] Julie Owono. Ivorian internet users fear a new era of terror in abidjan. <http://globalvoicesonline.org/2010/12/20/cote-divoire-ivorian-internet-users-fear-a-new-era-of-terror-in-abidjan>, 2011.
- [21] Anup Patel, Niveeta Sharma, and Magdalini Eirinaki. Negative database for data security. In *Proceedings of the 2009 International Conference on Computing, Engineering and Information*, ICC '09, pages 67–70, Washington, DC, USA, 2009. IEEE Computer Society.
- [22] Paul S. Generalized birthday paradox. <http://gdtr.wordpress.com/2013/01/13/generalized-birthday-paradox-keygenme3-by-dcoder/>.