Enhancing Internet Traffic Privacy using l-Diversity and Tor

Pramod Kumar Maurya
School of Computer Science and Engineering
VIT, Vellore, Tamil Nadu, India

Rimjhim Singh
School of Computer Science and Engineering
VIT, Vellore, Tamil Nadu, India

Damayanti Chattopadhyay
School of Computer Science and Engineering
VIT, Vellore, Tamil Nadu, India

Apoorva Jain
School of Computer Science and Engineering
VIT, Vellore, Tamil Nadu, India

Devjyot Singh Siddhu
School of Computer Science and Engineering
VIT, Vellore, Tamil Nadu, India

Siddarth Anoop Nayar
School of Computer Science and Engineering
VIT, Vellore, Tamil Nadu, India

Divyang Arora
School of Computer Science and Engineering
VIT, Vellore, Tamil Nadu, India

Abstract- In the past few years, there has been a considerable rise in the implementation of computer systems and technology across different sectors. In turn, it has led to the significant storage of both personal and organizational sensitive data on unprotected network channels. The universality of PCs and keen gadgets on the Internet has prompted tremendous measures of information being gathered from clients, shared, and investigated. This prompts a sharp consciousness of the need to shield information and assets from divulgence, ensure the realness of information and messages, and shield frameworks from network-based assaults. Most people operating their systems, are unaware of the required protocols for the maintenance of confidentiality, integrity and availability; which can seriously expose them to online assailants. Neglecting updates, defaults in product design and uncertain development issues leave the clients open to PC security weaknesses. Therefore, we will be researching the techniques to anonymize the IP Addresses and mask the data packets.

Keywords –L-Diversity, Tor Browser, Anonymity, Internet Privacy

I. INTRODUCTION

The growing network insecurities have prompted us, users, to learn that in the modern world of technology, it is important to know how to protect your organization's data from any kind of threats. Furthermore, the biggest threat does not come from any rogue hackers or other nations, but from within. We have learnt that around 70% of the world's database administrators have complete access to any data in a system they support, including private and sensitive data. Also, only 41% of organizations control access to data based on roles. Thus, we follow data masking.

Data Masking is the process of shielding or hiding original confidential data from any unintended exposure to reduce the risk of data breaches. It reduces the risk of data breaches by masking the test and development environments created from production data regardless of the database, platform, or location. One of the key reasons to follow this approach is in its freedom from requiring any direct changes to the database or application source code. This implies that masking can be applied rapidly and with no unnecessary cluttering to secure information in an association.Moreover, data masking is granular and helps in selective masking of rows, columns, or cell level. The technique we use for data masking is l-diversity. L-diversity is a method used to publish data in data sets while limiting and anonymizing the amount of sensitive information which is being disclosed. We follow the l-diversity approach over the k-anonymity model because it handles some of its weaknesses where protected identities to the level of k-individuals are not equivalent to protecting the sensitive values that were masked, especially when the sensor values within a group exhibit uniformity.

Another method we use in masking is IP address masking. Here, a subnet mask is used to divide the IP address into two parts; wherein the first part identifies the host (host address) while the second part identifies the network (network address) to which it belongs. For this, we use the Tor network. Tor is an internet browser that provides the user with the ability to surf the web anonymously. Some of the reasons why we select Tor are, firstly, Tor directs the internet traffic through a network of thousands of relays, secondly, the messages sent across this network are encapsulated in layers of encryption much like the layers of an onion. Moreover, the Tor network consists of .onion sites or "hidden services". Thirdly, Tor facilitates anonymized browsing by allowing traffic to pass onto or through the network through nodes that only know the immediately preceding and following node in a relay. Lastly, the source and destination of messages are hidden by encryption.

II.LITERATURE SURVEY

P. Winters mentions in their work [1], The Onion services look somewhat similar to the Internet from the 1990s, the pages load slowly, the UI is clumsy, and the search engines are inadequate but still, people choose to use the onion services for extra security, privacy, and NAT punching properties. The following biases were noticed during the survey:

Not all the people who were aware of the survey participated which influenced the results. Their non-participation could be accredited to a lot of reasons like value for privacy, insufficient time, the belief that their opinion doesn't matter among others.

Most of the people who thought that Tor Browser wasn't made for them, didn't participate in the survey and the authors never got to know what drove these people away from Tor. The problem with this is that after the compilation of results, the user experience for the subset of people whose tolerance for inconvenience is higher than the rest was optimized. Each paragraph should start with an indentation of 4 spaces or 0.20".

According to K. Gallagher's work [2], the usage of anonymity-preserving tools such as Tor has increased greatly, because of the surveillance of online activities by corporate. The study was taken to understand the Tor browser UX in a normal setting. The sample considered was small and homogenous in terms of age, education, and cultural background. Furthermore, the research was carried out in the USA, so the nature of threats to civil services is different for the rest of the world. The browser monitoring script relied on various heuristics to determine state transitions in the browser, which made it prone to false positives.

A. Machanavajjhala uses [3] implementation of Incognito for generating k-anonymous tables and it was modified so that it can produce l-diverse tables as well. A k-anonymized dataset can't stop strong attacks because of the lack of diversity in the sensitive attributes. The authors showed that l-diversity and k-anonymity have structural similarity and the k-anonymity algorithm can be modified to work with l-diversity.

Experimentation devices encourage investigation of Tor execution and security research issues and permit analysts to securely and secretly lead Tor tests without gambling mischief to genuine Tor clients [4]. Notwithstanding, scientists utilizing these devices design them to produce network traffic dependent on working on suspicions and obsolete estimations and without understanding the adequacy of their arrangement decisions. In this work, the authors planned a novel method for powerfully learning the Tor network traffic models utilizing covered up Markov displaying and protection saving estimation strategies. They lead a safe however point by point estimation investigation of Tor utilizing 17 transfers (about 2% of Tor data transmission) throughout a half year, estimating general measurements and models that can be utilized to create a grouping of streams and bundles. They then show how the estimation results and traffic models can be utilized to produce traffic streams in private Tor organizations and how their models are more practical than standard and elective organization traffic age strategies.

B.K. Tripathy's work explains [5] utilization of distributed hierarchical information for an assortment of purposes gets the opportunity of infringement of spillage of individual privileged data. Primer endeavours toward this path are

powerless to spillage of important data through quasi-identifiers. In recent years, a few algorithms dependent on the idea of k-anonymity have been created to deal with such issues. A superior security model, called l-diversity was proposed to deal with a portion of the issues in k-anonymity. The researchers' principal commitment in this paper is to improve the grouping period of the OKA algorithm so it deals with k-anonymity and l-diversity to an impressive degree and in a mix with the improved second and third periods of the algorithm in prompts a productive l-diversity algorithm. They likewise show that every one of the three phases of the algorithm is essential to cover various circumstances.

The authors of the paper [6] specifically discussed the security analysis of the tor browser. They attempted to analyse or quantify the security provided by the tor browser against specific attack in detail. They first discussed the initial prototype of the tor browser and how it secured data. After analysing it in detail they pointed out the shortcomings in the first method. According to the author's research, the initial design lacked no. of features to make the system robust and scalable, also it lacked security against insiders attack or more extensive eavesdropping. This is where they start analysing the second-generation security model proposed for the tor browser. They start by describing the architecture and design of the second-generation system, followed by security goals and adversary models. At last, they present security assessment based on earlier defined assumptions and compared the system with similar goals, most specifically Crowds.

Most associations are managing enormous measures of data assortment and are put away in huge information bases [7]. Personal Health Record (PHR) is an arising patient-focused model of trade of wellbeing data, frequently reevaluated for third-party storage, for example, cloud providers. There have been wide-running issues concerning security, in any case, as close-to-home wellbeing information might be uncovered to those outsider workers and unapproved parties. This work expects to feature three of the mainstream methodologies for clinical anonymization, specifically k-anonymity, l-diversity, and t-closeness. There is likewise an outline of the advantages and shortcomings of these methodologies. Broad scientific and test discoveries are introduced showing our proposed plan's security, adaptability, and execution.

The vulnerabilities that presently exist about the adequacy of organization data anonymization, from both technological and strategic points of view, leave the research community in a weak position. Indeed, even as the field walks forward, it does as such with minimal comprehension of the ramifications of distributing anonymized network information on the security of the organizations being checked and the utility to scientists. Without that understanding, information distributors are left to think about what fields should be anonymized to dodge legal aftermath, while analysts question the certainty of results acquired from the information. Be that as it may, the broad work done on miniature information obscurity furnishes the organization research local area with a few valuable experiences about how to successfully apply anonymization to distributed information. Simultaneously, earlier shrewdness can't be applied straightforwardly without first conquering a few difficulties, including the improvement of fitting security and utility definitions for the more perplexing instance of organization information. Tending to these difficulties is fundamental, in our view, to guarantee the proceeded, yet mindful, accessibility of organization follow information to help security research. [8]

The recent growth of many prominent social networks, as well as the release of social network data, has increased the risk of individuals' sensitive information being exposed. This necessitated the protection of privacy before the release of such information. Several algorithms have been developed to protect privacy in microdata. However, since nodes in social networks have structural properties in addition to labels, these algorithms cannot be used directly. To anonymize microdata, k-anonymity and l-diversity are effective tools. As a result, similar algorithms for handling social network anonymization have been sought. The paper suggests an algorithm for achieving k-anonymity and l-diversity in social network anonymization. This algorithm is based on some previously established algorithms in this area. [9]

Conventionally, anonymisation techniques have only addressed static datasets, which remain unchanged even after processing. However, real-world data sets are dynamic. Most of these data sets are large and it is impractical to re-anonymize them. The authors have identified the limitations of such conventional techniques in anonymizing the dynamic datasets and have proposed a new probabilistic data structure, called ' A Cuckoo Filter', for approximating set-membership tests and for improving the efficiency of dataprocessing. [10]

Albeit k-anonymity has acquired prominence for information protection as a result of its straightforwardness, and various calculations have been created and applied to genuine world datasets, sensitive data anonymized utilizing k-anonymity isn't altogether secure and is powerless against certain assaults. Hence, l-Diversity was developed to address the limitations of k-anonymity and also, to strengthen the protection measures of privacy.

Test results exhibited that the proposed information anonymization calculation handled information more proficiently than other customary calculations, requiring substantially less running time than regular re-anonymization

of whole datasets. The Cuckoo-sifted calculation was particularly productive, significantly diminishing activity execution times while keeping up the protection of progressively developing datasets.

III. VULNERABILITIES IN NETWORK CHANNEL

A network vulnerability is a flaw or weakness in software, hardware, or organizational processes that can lead to a security breach if it is exploited by a threat. Software or data are the most common nonphysical network vulnerabilities. For instance, if an operating system (OS) is not modified with the latest security updates, it can be vulnerable to network attacks. A virus could infect the OS, the host it's on, and possibly the entire network if it's not patched. One of the network vulnerabilities that we will be addressing here is that of IP addresses.

Every IP address is special, almost like a virtual fingerprint, so any online activity you take part in can be easily traced back to you. While you might believe that this isn't a big deal because you don't visit illegal websites or partake in illegal activity online, the truth is that if anyone knows your IP address, they also know your Internet service provider and can easily locate your physical location. Since having your IP address revealed can put you in a lot of danger, it's critical to hide your IP address for your privacy and protection.

Someone may want to use your IP address for malicious purposes because it contains information about you. People can obtain your IP address in a variety of ways.

Any member of the swarm (total seeders and leechers) will see your IP address when you download material from torrent sites. All they have to do now is look at the list of peers. If you give someone an email, they will look at the message's header, which might contain your IP address. IP addresses are known to be used in the email headers of Yahoo! and Microsoft Outlook. Any connection you click would need your IP address for the server on the other end to deliver the content requested. Your IP address would be visible to whoever owns the server.

A common form of internet surveillance technique called 'traffic analysis' is used to infer who is talking over the web and to whom. By knowing the IP addresses of the source and the destination of such a communication, one can easily tap into the conversation.

Internet data packets have two parts; a data payload and a header used to route the data. The payload refers to the information or the data that is being sent while the header consists of the information about the source, the destination, the size and the timing of the data. Traffic analysis focuses on the header part of the packet and hence can be used effectively even against encrypted data.

IV. TOR-BROWSER

The Tor (The Onion Router) Project, Inc is a USA non-profit entity since 2006 as a 501(c)(3) organization. However, the concept of Tor or ‘onion routing’ in particular has been around for more than 3 decades now; since the 1990s. The basic idea behind the development of the browser as mentioned on Tor's official website is to let every surfer of the internet have private access to an uncensored web.

The goal of onion routing, as mentioned above, was for all users to access the internet without any concern for their privacy. The idea was simple, to route the traffic through multiple servers and encrypt it every time. This is fundamental that still forms the basis of Tor in today's date.

4.1. Working of Tor Browser

Tor is an anonymity network that, as the name suggests, helps in letting the user stay anonymous while surfing the internet. Tor is a preventive method for the users to dodge internet surveillance using traffic analysis.

Tor tackles the above-mentioned problem of traffic analysis through a distributed and anonymous network. What this means is that when a user sends some information, it has to go through a series of anonymous network nodes before finally reaching the receiver. Moreover, on every node, the data is encrypted and re-encrypted except the last node which communicates with the receiver. The route which was chosen and the number of nodes the data goes through is not the same for every data packet and hence makes it even more difficult for trackers to track it. According to the official Tor website, the idea is somewhat analogous to the use of a twisty and long path to throw someone off your trail and then periodically erasing your footprints as well. Tor network ensures that no single point link can be formed between the source and the destination.

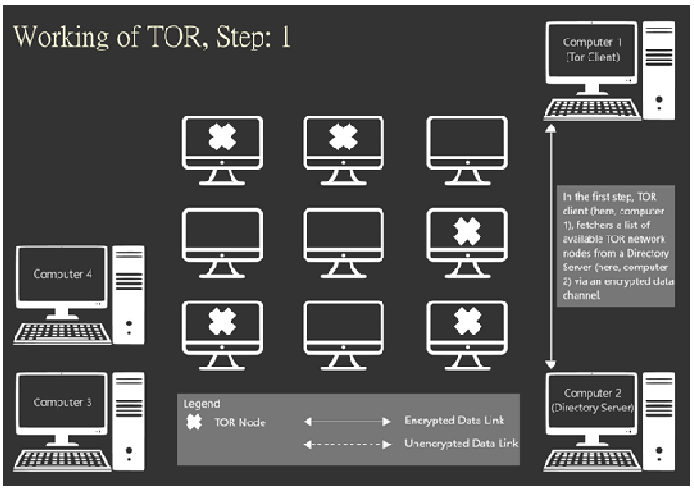

Figure 1. Step 1 of working of TOR

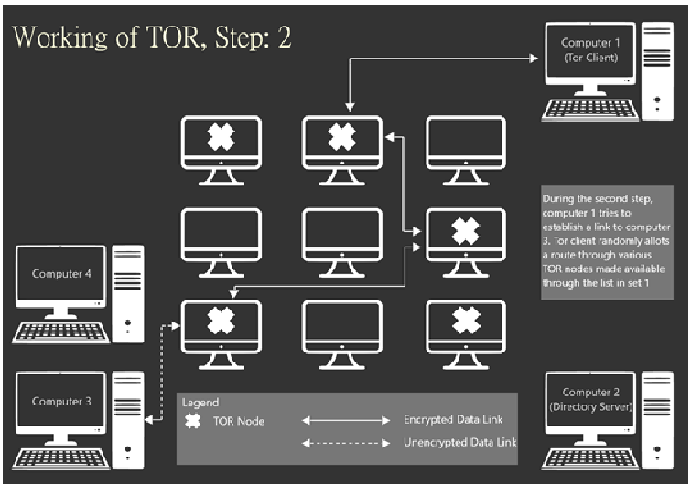

Figure 2. Step 2 of Working of TOR

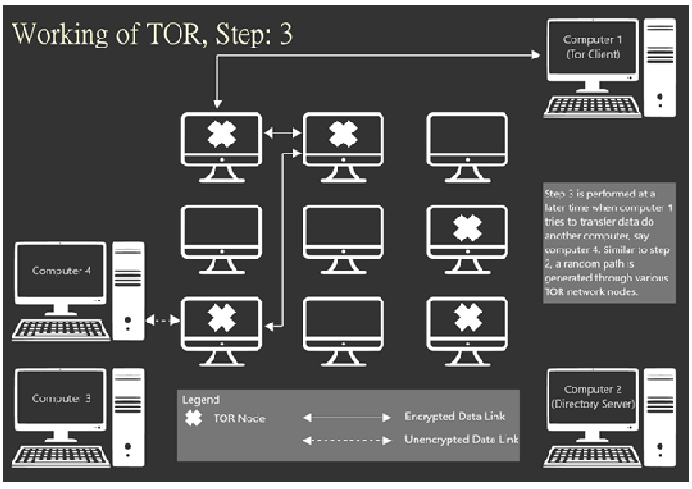

Figure 3. Step 3 of Working of TOR

V. VULNERABILITIES IN A NETWORK PACKET

In every web-based application, communication between the client-side and the server-side takes place, which is 2 programs running simultaneously. The program on the browser responds to user inputs and sends requests to the server, and the program on the server responds to this request. The web application's status changes as per the user inputs and the data goes from the client to the server following the request-response cycle. The packet bursts that are generated are unique in themselves, which makes it easy for an attacker to snoop on the exchange of data, this is also known as a side-channel attack. These attacks are based on side-channel information, which is the data that can be retrieved which is neither in the form of encrypted data nor in the form of simple text.

An attacker can try to backtrack the user's inputs by tracking the packet burst behaviour and observe them determine the content of the packet bursts. The attacker can monitor the network traffic and guess the actual data or get closer to obtaining the data which is being sent by the client to the server.

VI. STUDY ON L-DIVERSITY

6.1. Limitations of k-Anonymity–

Table -1 Depiction of lowest scores obtained by students in a class

	Age	Gender	Test Score
1	20	M	10
2	20	F	20
3	21	M	33
4	21	M	40
5	20	F	30

The above table, Table 1 depicts the lowest scores obtained by five students in a class. The Teacher of the class didn't reveal these students' test scores to the whole class as she didn't want to embarrass them in front of everyone and give them a second chance. However, the naughtiest student in the class got hold of this table.

1. Homogeneity Attack

Even though the records have been anonymized, the esteem of the sensitive property for that group of k records can be anticipated precisely. In the given example, the teacher had announced the marks of each student in the class except the ones who scored the lowest. The naughtiest student would have identified those five students since being a member of his class, he knows all his classmates. Then he will have a 20% accuracy on matching the data to the correct individual. This circumstance can also occur when large datasets are involved. Therefore, it is important to diversify the sensitive attributes that share the same QI values.

2. Background Knowledge Attack

For the above scenario, the naughtiest student may have more background information on one of the students. He might know the age of the three males present in the table and could easily deduce the marks of the youngest male since the other two are of the same age. Similarly, he might know other behavioural patterns of the two female students, like, a student might treat herself to a brownie from the school canteen every time she scored well. So, if in this case, the person didn't buy the brownie on receiving her marks that imply that she didn't score well. Thus, it will become easier for the naughtiest student to match her marks to her identity.

According to the l-Diversity principle, a q*-block is said to be l-diverse only if it contains at least l "well-represented" values for the sensitive attribute S and once, every q*-block is diversified, does a table become l-diverse.

6.2.l–Diversity for Internet Packets

We have previously discussed the demerits of k-anonymity and have defined how l-diversity overcomes the issues faced in k- anonymity. We will be implementing l-diversity on internet packet burst instead of k-anonymity. Unlike k-anonymity, l-diversity gives equal preference to inputs, but each input has some weight associated with it. This weight simply denotes the frequency f occurrence. There are various scales on which the weight associated can be calculated, for instance, millions of actions per input or number of mouse clicks per second, etc. This enables us to use l-l-diversity on the weights and form a quasi-identifier group.

VII. APPLYING L-DIVERSITY TO INTERNET PACKETS

L-diversity is an algorithm that is considered an enhancement to k-anonymity by reducing the overheads and padding cost. The algorithm follows the same steps as k-anonymity with one extra rendition with each keystroke

assigned to weight. This makes it more efficient than k-anonymity. There are three types of applicable l-diversity: distinct l-diversity, entropy l-diversity, and recursive (c,l)- diversity.

Distinct l-Diversity: Distinct l-diversity is one of the most commonly applied diversifying techniques. Applying this technique ensures that for each equivalence class, the sensitive field has at least 'l' distinct values. However, this technique cannot prevent probabilistic interference attack that employs a data mining procedure to deduce accurate conclusions.

Entropy l-Diversity: According to the Entropy l-diversity each QI value in a group should be greater than or equal to the logarithm (l).

To present it mathematically:

$Entropy\ (QI^*) \geqslant log\ (l)$

$Entropy\ (QI^*) = -\sum ((QI^*, sa)\ log\ (p\ (QI^* sa \in \mathcal{S}, sa)))$.

Here,

sa= sensitive attributes

(p(QI, sa))*= fraction of records in the equivalence class E.

Recursive l-Diversity: Recursive l-Density tackles an important issue of diversifying tables, i.e., a sensitive attribute that is common in the table, will not appear very frequently, and at the same time, a sensitive value whose occurrence is less in the table, should not appear very rarely.

Based on the definition, in any quasi-identifier group, the sensitive attribute should always be l well represented. For the table to be l-diverse, all the groups in a table should satisfy this condition of l-diversity. Here we proposed distinct l-diversity and checked the probability of occurrence of keystrokes being less than or equal to 1/l to satisfy the condition for l-diversity. If all the groups of quasi-identifier satisfy this condition then the whole table is considered l-diverse.

Following are the steps to implement l-diversity on raw data:-

Step 1: Consider all packet bursts for each keystroke. Each keystroke in the table is represented as Key_i, and its respective packet burst for the i^{th} key is represented as such that $1 \leq i \leq n$.

Step 2: Denote weight as *W*. *W* is the probability of occurrence for each keystroke. Now sort the data w.r.t. *W* in decreasing order.

Step 3: Consider the sorted data, and form groups based on privacy parameter *l* such that the keystroke in each group is $\leq 1/l$.

Step 4: Groups are formed based on the composition of formula $A^2 - 1$ (i.e. If the no. of letters are 6 then possible groups are $6^2 - 1$. For simplicity in this example, we will only take 2 options.

Step 5: Then, pad the packets to the value of the highest size in that group. After padding chooses the optimum cost of partition from multiple group partitions.

Step 6: Once the best possible partition is chosen for keystroke 1st, then choose the candidate to form a group for 2nd keystroke. This time it should have a prefix relationship with groups formed in 1st keystroke. Choose the most optimum option out of all and continue the process for i^{th} keystroke.

Step 7: Finally, we will now have a scheme for l-diverse packet burst for a valid set of user interactions such as keystrokes, voice recognition or touch screen, among others. For example, we have chosen these associated weights

For first keystroke {a, b, c, d, e, f} = {31, 30, 21, 19, 16, 15}, and for second keystroke {ap, bu, ct, dh, en, fm} = {36, 31, 19, 18, 16, 15}

Table 1 represents l-diverse partition options. This table is formed by the steps explained earlier. Here padding cost is the extra bytes required for forming groups.

Table -2 l-diverse partition options

FirstKeystroke		**ActualPacket Burstin Bytes**	***l*-diversifiedPacketBursts in Bytes forFirst Keystroke**		**SecondKeystroke**		**ActualPacket Burstin Bytes**	***l*-diversified PacketBursts in Bytes forSecondKeystroke**	
	Weight		***Group Options***			***Weight***		***Group Options***	
			Option 1	***Option 2***				***Option 1***	***Option 2***
a	31	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{3},\overrightarrow{3}$	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{8},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{8},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{8},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{8},\overrightarrow{3}$	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{5},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{5},\overrightarrow{3}$	**ap**	36	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{3},\overrightarrow{1}$	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{9},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{9},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{9},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{9},\overrightarrow{1}$	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{5},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{5},\overrightarrow{1}$
b	30	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{5},\overrightarrow{3}$			**bu**	31	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{5},\overrightarrow{1}$		
c	21	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{8},\overrightarrow{3}$		$\overrightarrow{3},\overrightarrow{4},\overrightarrow{17},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{17},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{17},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{17},\overrightarrow{3}$	**ct**	19	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{9},\overrightarrow{1}$		$\overrightarrow{6},\overrightarrow{5},\overrightarrow{14},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{14},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{14},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{14},\overrightarrow{1}$
d	19	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{6},\overrightarrow{3}$			**dh**	18	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{7},\overrightarrow{1}$		
e	16	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{17},\overrightarrow{3}$	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{17},\overrightarrow{3}$ $\overrightarrow{3},\overrightarrow{4},\overrightarrow{17},\overrightarrow{3}$		**en**	16	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{14},\overrightarrow{1}$	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{14},\overrightarrow{1}$ $\overrightarrow{6},\overrightarrow{5},\overrightarrow{14},\overrightarrow{1}$	
f	15	$\overrightarrow{3},\overrightarrow{4},\overrightarrow{9},\overrightarrow{3}$			**fm**	15	$\overrightarrow{6},\overrightarrow{5},\overrightarrow{12},\overrightarrow{1}$		
TotalPadding Costs			**18bytes**	**30bytes**				**14bytes**	**16bytes**
SA		**QI**	**Generalization**		**SA**		**QI**	**Generalization**	

Table -3 Padding cost for Table 1

Padding Cost forFirst KeystrokeOption1:18 bytes	**Group1** {a,b,c,d} **Group2** {e,f}	**Padding Cost forSecond KeystrokeOption1:**14 bytes	**Group1** { *ap,bu, ct,dh}* **Group2** { *en,fm* }
Padding Cost forFirst KeystrokeOption2:30 bytes	**Group1** {a,b} **Group2** {c,d,e,f}	**Padding Cost forSecond KeystrokeOption2:**16 bytes	**Group1** {ap, bu} **Group2** {ct, dh,en,fm}

VIII. LIMITATIONS

3 major drawbacks were identified in this approach:

- Identification of optimum padding cost groups – Only the cost of parent nodes are being considered whereas the cost of both the parent nodes and child nodes should be considered. The tree should be traversed from the first keystroke to the nth keystroke and then the minimum cost amongst them should be checked.

- The huge number of feasible partitions while establishing groups – Initially, raw information was sorted in non-increasing order for grouping based on integer composition. But now, the candidates which are in the order of sorted raw information are considered for grouping. This is inefficient as it skipped feasible partitions.

- A large number of probable ways while computing the padding cost – The main focus of l-diversity is to find the probability of the instance of each user action. Let us consider A={a,b,c} to be a group with {31,30,21} as weights and 'p' be the probability of instance.

  Then,

  $p(a|A) = \frac{31}{31+30+21} = 0.37$

  $p(b|A) = \frac{30}{31+30+21} = 0.36$

  $p(c|A) = \frac{21}{31+30+21} = 0.25$

  This means, 'a' occurs 37% of the time, 'b' occurs 36% of the time and 'c' occurs 25% of the time.

  Padding cost for 'a' will be padding cost * p(a|A), which is 1.85 bytes (5 bytes * 0.37)

  Padding cost for 'b' will be padding cost * p(b|A), which is 2.16 bytes (6 bytes * 0.36)

  Padding cost for 'c' will be padding cost * p(c|A), which is 1 byte (4 bytes * 0.25)

  The total padding cost for A will become 5.01 bytes. So, the padding cost for each group should be calculated similarly and the optimum padding cost should be chosen.

IX. FUTURE ENHANCEMENTS

We have discussed the application of TOR in masking IP addresses. Then we included the steps to implement L-diversity to mask internet packets. We dug deep and found a way to implement l-diversity for internet packets on tor browser. This proposed architecture diagram helps in masking both internet packets as well as IP addresses. Making the data and information more secure from side-channel attacks. TOR browser is 33rd party software so we cannot change architecture to accommodate o diversity. This is just a proposed diagram for future implementation.

In this, suppose a packet jumps from NODE 1 to NODE 2 in a TOR connection, we want NODE 1 to implement l- diversity on internet packet. Making it almost impossible to regain information from outside. Again when the packets jump from NODE 2 to NODE 3 the packets are masked for more security. This not only secures IP Addresses but also secures the data packets. The cost of implementation and time to reach the destination will depend on the number of nodes in between but this tends to increase. Due to the addition of l-diversification of packets the results might reach late to the target. The impact of increased cost can be compensated with the privacy this method provides to the user. This work is only proposed theoretically and yet to be implemented.

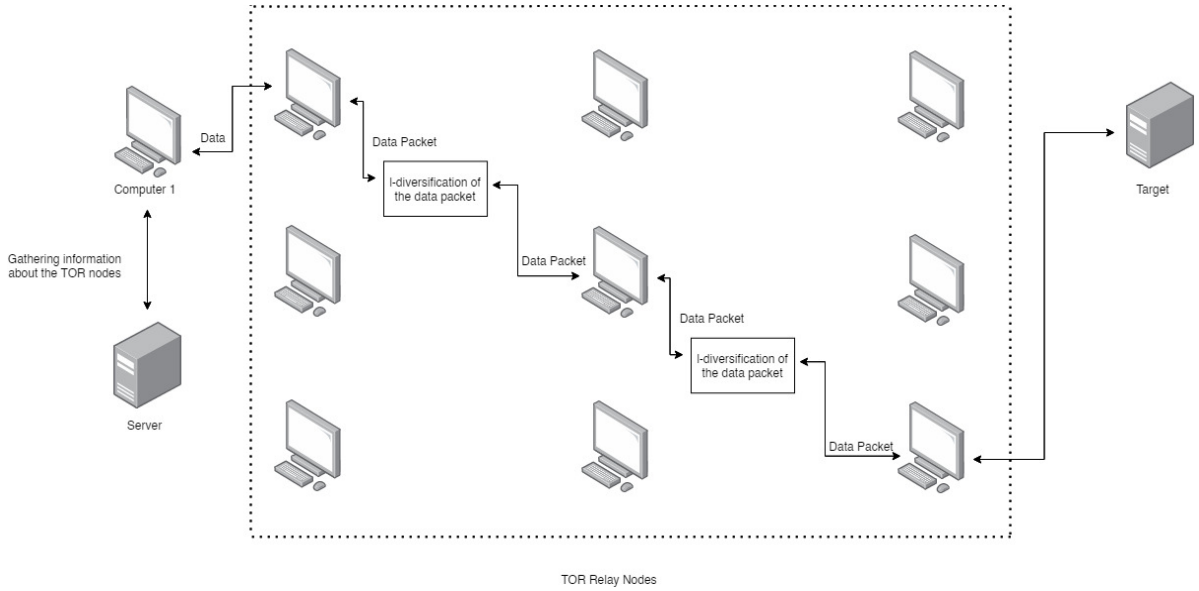

Figure 4. Architecture diagram of the proposed method

X.CONCLUSION

Internet is now present globally. It has now become an inevitable part of our lives. The need for privacy of data on the internet is creasing day by day. With the very passing hour users across the globe upload/enter their sensitive data on the internet. We first discussed the usage of TOR for masking the IP addresses. We dive deep into its usage and working. But the problem with the TOR browser is it only provides masking for IP address leaving the data packets vulnerable tothe side-channel type of attack. In the second part of the paper, we discussed various types of l-diversity and their implementation on a simple database.

We took forward the same concept and implemented l-diversity on internet packets. There is a constant flow of internet packets between the client and server. These packets carry sensitive information. An attacker can intercept these packets and forma link among them making the sensitive information of the user vulnerable. We came up with the solution by implementing l-diversity on these packets'values. This masks the sensitive information of the packet by grouping them. Inthis way, the attacker won't be able to intercept the message that is being transferred between client and server.

In the end, we also proposed a theoretical way of implementing l-diversity on a TOR network. When the packet jumps from one node to another in a network, data packets can be anonymised using l-diversity making it more secure and impossible to intercept or interpret.

REFERENCES

[1] P. Winter, L. Roberts, M. Chetty, and N. Feamster, "How Do Tor Users Interact With Onion Services?", in SEC'18: Proc. of the 27th USENIX Conference on Security Symposium, USA pp. 411–428 2018.

[2] K. Gallagher, S. Patil, B. Dolan-Gavitt, D. McCoy, and N. Memon, "Peeling the Onion's User Experience Layer", in Proc of ACM SIGSAC Conference on Computer and Communications Security, pp. 1290–1305 2018

[3] A. Machanavajjhala, J. Gehrke, D. Kifer and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," 22nd International Conference on Data Engineering (ICDE'06), pp. 24-24, 2006.

[4] R. Jansen, M. Traudt, and N. Hopper, "Privacy-preserving dynamic learning of Tor network traffic." in Proc. of the ACM SIGSAC Conference on Computer and Communications Security, pp. 1944-1966, 2018.

[5] B.K. Tripathy, K. Kumaran, G.K. Panda "An Improved l-Diversity Anonymisation Algorithm" Communications in Computer and Information Science, Springer, Berlin, vol 157, 2011.

[6] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, "Towards an Analysis of Onion Routing Security," International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability, pp 96-114, 2001.

[7] R. Keerthana, M. Jayabalan, and M. E. Rana. "A study on k-anonymity, l-diversity, and t-closeness techniques." IJCSNS vol. 17, issue 12, 2012.

[8] S. E. Coull, F. Monrose, M. K. Reiter and M. Bailey, "The Challenges of Effectively Anonymizing Network Data," 2009 Cybersecurity Applications & Technology Conference for Homeland Security, pp. 230-236, 2009

[9] B. K. Tripathy and A. Mitra, "An algorithm to achieve k-anonymity and l-diversity anonymization in social networks," 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), pp. 126-131, 2012.

[10] O. Temuujin, J. Ahn and D. Im, "Efficient L-Diversity Algorithm for Preserving Privacy of Dynamically Published Datasets," in IEEE Access, vol. 7, pp. 122878-122888, 2019