Effects of Shared Bandwidth on Anonymity of the I2P Network Users

Khalid Shahbar A. Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Canada
{Shahbar, Zincir}@ cs.dal.ca

***Abstract*—In this paper, we studied the effects of sharing the bandwidth on the I2P network on the anonymity level it provides to the users. To this end, we explored what could be achieved by a potential attacker on the I2P Network in terms of application and user profiling. In both cases, the effect of bandwidth participation has been been analyzed. To achieve this, we used a machine learning based approach to analyze the flows extracted from the traffic generated by the applications and the users. Our results show that profiling the users and applications on the I2P network is possible. The amount of shared bandwidth has an effect on the accuracy of profiling the users and the applications. Furthermore, applications that do not use the shared clients tunnels increases the possibility to profile the behavior of the flows for these applications.**

Keywords— I2P network; Traffic Flow; Anonymity; Data Analytics

I. INTRODUCTION

The available anonymity systems on the Internet work on the concept of separating the users' identity and his/her final destination to provide anonymity. This separation is achieved by indirectly connecting the user to the final destination through multiple stations. The number of stations varies based on the anonymity system used. For each station the user connects, another layer of encryption is added to the user's information. Therefore, the information that could potentially link the user to the final destination (e.g. the website that user browses) is not known by any of the intermediate stations (nodes) that carry the user's data. The stations on the path to the final destination can only see the necessary part (network header information) to carry the data to the next station. Tor [1], JonDonym [2], and I2P [3] are examples of such networks that use this mechanism to separate the user from his/her final destination. For example, if the user is browsing a website, then the server of that website does not know the identity or the location of the user. Also if anyone is observing the connection of the user to the anonymity network, it could be shown that the user is connected to an anonymity network but without revealing the website or the activity the user performs.

There are many differences between the anonymity networks on the design and the applications they support. For example, I2P network is different than Tor in its structure as a private network. The websites on the I2P network "called Eepsites" [27] are hosted within the network itself and have the .i2p names based on the naming and addressing on the I2P network [28]. Even though I2P supports and enables the browsing to websites by using outproxy, the I2P network is designed to work better and more anonymously when accessing the resources within the I2P network. The services (applications) that I2P supports are not limited to browsing [10] [23], it supports multiple other services such as file sharing, Internet Relay Chat, E-mail etc.

I2P network is a decentralized network, there is no central server managing the network. The network database is stored in "netDb" [11]. The netDb contains "routerInfo" and "leaseSet". The rounterInfo contains the required information to contact a router. The leaseSet contains the required information to reach to a destination. The user builds his/her knowledge about the network by using the information from the netDb. Sending and receiving data on the I2P network and building the knowledge about the network is done by building "Inbound and Outbound Tunnels" [12]. The tunnels are unidirectional [25], the inbound tunnels are used by the users to receive messages and the outbound tunnels are used to send messages. The default configuration of the users' agents (clients) enables the bandwidth participation, that means in addition to the user building his/her tunnels, the user can also participate on building other users' tunnels. The tunnels consist of two or more routers based on the client configuration and the tunnel type. Therefore, when the user participates in building tunnels, his/her role could be the first or the last or one in the middle in forming the tunnel. At the same time, the user could continue to send/receive his/her messages (if any). This aims to enhance the anonymity because it makes it harder to separate a specific user's tunnels from the other participating tunnels.

I2P network uses separate tunnels for the outgoing and incoming traffic. That means it is impossible to link between the sent and received data of a user while observing a tunnel. In addition, the inbound tunnels of the users are used to receive any messages from any source. This way it is not possible for an attacker who is observing the connection to detect the source of a message sent to a user.

The tunnels are used to send and receive messages, to communicate with the netDb, and to manage the tunnels. Therefore, the messages that travel through the user's tunnels, do not always represent only the messages traveling between the users. So, if the tunnels contain this type of control and user messages mixed together, and the incoming / outgoing tunnels are separated, then we aim to study the following research questions: What is the effect of such a design in terms of anonymizing the netflow behavior of a user’s activities? Can a user’s activities completely anonymized by this design? Or do they rely on the amount of other users' traffic that shares the bandwidth?

The tunnels in the I2P network are short-lived, this hardens the possibility to profile the user’s activity based on monitoring the tunnels. To overcome that the tunnels on the I2P network are short-lived, is it possible to collect information about multiple short-lived tunnels (related to the same user) to profile the user’s activities? Does this give an indication about the level of the overhead (influence of the overhead due to routing other's traffic) when the user participates on the netDb or when carries other users’ traffic to hide his/her activities? To find answers to these questions, in this paper we studied: (i) the ability to identify the type of an application the user is using; (ii) the effect of the bandwidth participation on the ability to identify the type of an application; (iii) the effect of bandwidth participation on the ability to profile the users; and (iv) regardless of the application used, the ability to profile the user and to distinguish between different users by observing the tunnels.

The rest of paper is organized as follows. The related work is summarized in Section II. The system set up and data collection are introduced in Section III. Section IV presents the experiments and results, while Section V presents our observations on the I2P network. Finally, conclusions are drawn and the future work is discussed in Section VI.

II. Related Work

Timpanaro et al. [4] proposed a monitoring architecture for the I2P network to describe how it is used. The proposed system analyzed what type of applications are used on the I2P network. The applications that the monitoring architecture can identify are limited to web browsing and I2PSnark. The results showed that the proposed monitoring architecture could identify 32% of all running applications. The experiments performed depended on using a router on the I2P network to work as floodfill router. After collecting numbers of leaseSet of the networks, the leaseSet was tested to determine if it belongs to a web server or I2PSnark. Their results showed that the classification of the leaseSet does not relate the type of application with the user.

Egger et al. [5] presented several attacks that could be implemented against I2P network. The authors claimed that their attacks against the I2P network could reveal the services that the I2P user accesses, the time of access, and the time spent using the service. The attacks first control most of the nodes that host the decentralized database (netDB) on the I2P network. Then, they monitor the network activities to link the related ones. Denial of Service (DoS) attacks could be used to disable the nodes hosting the netDB and speed the takeover process.

Liu et al. [6] presented four methods to discover the I2P routers. They discovered around 95% of all the I2P routers in their two weeks long experiment. One of their methods to discover the I2P router was to run an I2P router and monitor the communications with other I2P routers to collect information about them. Another method was to run an I2P FloodFill (the method used to distribute the netDb) [11] router to monitor and collect information about routers that make communication with their FloodFill router. The third method to discover the I2P router was the “crawling reseed URL”. This method used the reseed option (Initial set of I2P nodes needed for Bootstrap) in the I2P network [21] to collect the I2P routers information. The fourth method was “exploiting NetDB”, where the I2P mechanism of a router query and a response were used to collect routers’ information.

Herrmann and Grothoff [7] presented an attack that determines the identity of the HTTP hosting peers (routers) on the I2P network. The attack required using three types of routers. The first type is used to provide information about the tunnel operations to the attacker. The second type is used to direct the user to select the attacker’s routers by performing a DOS attack. The third type is used to perform requests to the Eepsite. The combination of using the three types of routers was then used to identify the hosing router on the I2P network.

On the other hand, AlSabah et al [8] employed machine learning algorithms to study the type of application Tor user runs in the Tor network. The applications studied were web browsing, video streaming, and BitTorrent. They used the circuit and cell level information to extract features that could be used to classify the type of application the user running. The result showed 91% accuracy for offline classification and 97.8% accuracy for online classification.

Shahbar el al [9] built and evaluated two approaches to classify the type of application used by the user on the Tor network: Flow level and Circuit level [8] classification. The circuit level classification employed different set of features related to the circuits that the user creates when using the Tor network. The flow level classification employed the traffic flows between the user and the first node on the Tor network. The results showed up to 100% accuracy in both approaches, demonstrating the strength of flow analysis under such circumstances.

In this research, we investigate the effects of the bandwidth sharing on the I2P network and its potential usage by an attacker to identify both the user and the application on the I2P network.

III. Data Collection and Setup

In this paper, we used three machines (computers) to collect data on the I2P network. The version of the I2P software used on these machines was (0.9.16). The hardware specification of the three machines is shown in table I.

The applications we aim to study in this work (on the I2P network) are browsing, chat, and file downloading. The reason behind choosing these applications is that they are the most used applications. On each machine, we only run one application at a time while collecting the data. This is to ensure the ground truth of the data.

TABLE I. SPECIFICATIONS OF THE MACHINES USED FOR DATA COLLECTION

	OS	OS type	Processor	RAM	I2P Version
Machine 1	Ubuntu 12.04 LTS	64-bit	Intel Core 2 Duo CPU E8135 @ 2.4GHz	1.9 GiB	0.9.16
Machine 2	Ubuntu 12.04 LTS	32-bit	Intel Pentium 4 CPU 2.53 GHz	1.5 GiB	0.9.16
Machine 3	Ubuntu 12.04 LTS	32-bit	Intel Core 2 Duo CPU E4600 @ 2.4GHz	1.9 GiB	0.9.16

All the traffic of the applications and the traffic of the users are our traffic and do not include any other users traffic. For the part where we participate on other users' tunnels, the users' privacy is preserved. The encryption used on the I2P keeps the users' data private. In addition, before analyzing the traffic, all the IP addresses and payloads are removed.

A. *Browsing*

To collect the browsing data, we prepared a list with the available Eepsites on the I2P by default. This list includes the built-in (bookmarked) Eepsites on the I2P software such as (i2p-projekt.i2p). In addition to these web sites, we added some other Eepsites to the list by using Eepsites that provide a "search" service on the I2P network. After the list was ready, we used iMacro [13] to automate the browsing. To this end, we wrote a script that browses the first address on the list. Then it waits for a random period of time before it navigates through the Eepsite by clicking randomly on a link on the Eepsite. After moving (traversing) from one link to another multiple times by using this approach, the script picks the second link in the list and so on. The randomness in picking the link ensures that the visited Eepsites keep changing from one iteration to another. Some of the Eepsites contain links to websites outside of the I2P network. This results in the data collection to also include traffic to websites hosted outside of the I2P network but still accessed through the I2P network. This requires us to use an outproxy (a router on the I2P network works as a proxy to access websites outside of the I2P network). To this end, we used the default outproxies of I2P, namely false.i2p and outproxy-tor.meeh.i2p. To be able to collect real-life data, we set all the tunnel configurations to default. We collected data using this set up for seven days.

B. *Instant Relay Chat (IRC)*

For IRC, each machine in this research was also set up to work independently from the others. Again, only one type of application was working while collecting the data. During this process, we chose jIRCii [22] plugin [14] and installed it on the three machines. Then the machine connected to the Irc2P network (this is the Instant Relay Chat for I2P) by using the Irc2P Tunnel and used one of these servers (irc.dg.i2p) , (irc.postman.i2p) , or (irc.echelon.i2p). The machine stayed connected 24/7 on the Irc2P network and joined multiple channels such as #i2p, #i2pchat, #i2people etc. during this process for five days.

C. *Downloading Files Using Torrent (I2PSnark)*

To download files on the I2P network, we used I2PSnark [24] on all machines. It is one of the built-in applications within the I2P network. The downloaded files included files such as videos, documents, music, movies. etc. The size of the files varies from small to big. We got the torrent files from the Eepsite (diftracker.i2p) and (tracker.postman.i2p). The data of the torrent include both the uplink and the downlink of the files. We collected data using this service for seven days on the I2P network.

IV. EXPERIMENTS AND RESULTS

There are many machine learning algorithms used for the purpose of classification. In our previous work [9], we employed different supervised learning algorithms and approaches to identify applications used on the Tor network. The evaluated algorithms were C4.5, Random Forest, Naïve Bayes, and Bayes Net. Among these algorithm, C4.5 Decision Tree was the best performing algorithm to classify Tor traffic flows. Moreover, we evaluated Tranalyzer [15] and Tcptrace [29] as flow exporters for Tor. In our previous work we showed that Tranalyzer based traffic analysis system performed better than the Tcptrace based traffi analysis system. Therefore, in this research, we used Tranalyzer to export the flows and the C4.5 decision tree classifier (by the open source data mining tool, Weka [17]) to construct our traffic analysis system.

Tranalyzer has 92 features; the features include flow direction, duration, frequencies related to the packets in a flow such as the number of packets sent and the number of packets received, IP header information such as TOS and TTL, TCP header information such as window size and sequence number, packet length statistics such as the mean and the minimum packet length, inter arrival time statistics such as the median and the quartile. Tranalyzer features also include features related to ICMP, VLAN, MAC addresses which we removed from the data because they are not relevant for our experiments. The complete list of Tranalyzer features can be found in [15]. It should be noted here that we did not use the IP addresses and the port numbers in the analysis of the collected data not to bias the learning algorithms. Given that the data set is not big and only three machines are used in the collection of the data, the learning algorithms may easily link the applications to port numbers or IP addresses, if they are used as features in the analysis.

A. *Tunnel based data analysis*

In this case, we focused on differentiating Application tunnels from Exploratory and Participating Tunnels [26]. Exploratory Tunnels are used for the management (administration/control traffic of the I2P network) and also for testing purposes. The Participating Tunnels are the tunnels that the users use to relay other users' traffic. In the training phase of our classifier, to train a decision tree model in order to

TABLE II. BINARY CLASSIFIER ON THE TUNNELS

	TP Rate	FP Rate	TN Rate	FN Rate
Applications Tunnels	0.875	0.288	0.712	0.125
Others (Exploratory & Participating Tunnels)	0.712	0.125	0.875	0.288
Accuracy	82.04%			

differentiate the application tunnels from Exploratory and Participating Tunnels, we labelled the I2Psnark, Irc2p, and the shared clients tunnels as Applications tunnels class. We also labeled the Exploratory tunnels and the Participating tunnels (when the bandwidth setting is set to default 80% participating) as one class, called 'others'. The reason behind this is to investigate the ability to distinguish the application traffic from the management or other users' traffic. This way we have a binary classification problem, one represents the "applications" and the other represents "others" shared traffic. In this case, our analysis shows that we can differentiate these two groups of traffic in I2P tunnels up to 82% accuracy. Table II. shows the performance of our classifier on the test data, which was unseen by the classifier during the training, for this analysis.

The results are calculated using the following performance measurements: The metric "Accuracy" is defined as the summation of True Positive (TP) and True Negative (TN) values divided by the total number of instances (N). For example, when measuring the accuracy of the classification for the "applications tunnels" traffic, TP is the total number of correctly classified instances as "applications tunnels". TN is the total number of correctly classified instances as "others".

If an "Others" instance is classified as "applications tunnels" instance, then this is considered as FP. The opposite is when the classifier classifies an instance as an "others" instance while it is an "applications tunnels" instance, then this is a False Negative (FN). The TPR, FPR, TNR, and FNR are calculated using the following equations:

$$\text{TPR} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)+False Negative (FN)}} \quad (1)$$

$$\text{FPR} = \frac{\text{False Positive (FP)}}{\text{False Positive(FP)+True Negative (TN)}} \quad (2)$$

$$\text{TNR} = \frac{\text{True Negative (TN)}}{\text{False Positive(FP)+True Negative (TN)}} \quad (3)$$

$$\text{FNR} = \frac{\text{False Negative (FN)}}{\text{False Negative (FN)+ True Positive (TP)}} \quad (4)$$

TABLE III. CLASSIFICATION RESULTS FOR THE TUNNEL BASED TRAFFIC ANALYSIS

	TP Rate	FP Rate	TN Rate	FN Rate
I2Psnark	0.661	0.033	0.967	0.339
jIRCii	0.778	0.084	0.916	0.222
Eepsites	0.531	0.143	0.857	0.469
Exploratory & Participating Tunnels	0.755	0.152	0.848	0.245
Accuracy	70.3%			

We also aimed to analyze for what purpose a tunnel might be used. In this case, if we were running an application, for example I2Psnark, then we extracted the tunnels related to the I2Psnark and labeled them as I2Psnark. We did the same for jIRCii and Eepsites. The Eepsites tunnels, which are the client tunnels [27], might be used for another application on the I2P network. They also stay alive all the time that the user is online. On the other hand, the I2Psnark (Irc2P) tunnels stay alive as long as the user uses the application. The shared client tunnels could be used for I2Psnark, if the user changes the setting, but the default setting is to use the Irc2P tunnels. The Exploratory and Participating tunnels stay alive as they are. Aiming to shed light into for what purpose a tunnel might be used, is a very challenging problem. However, we could still achieve 70% accuracy (on the unseen test data) in predicting the potential purpose of a tunnel on the I2P network by just analyzing the flow features. Table III. presents the results for this analysis.

B. Applications and User based data analysis

In this part of our experiments, we examined the effect of the bandwidth participation on the I2P network based on two scenarios: the first one is the identification of the type of the application the user is running (Traffic Profiling); and the second one is the ability to profile the users under the effect of the amount of shared bandwidth (User Profiling). In both scenarios, we also included the investigation of the protocol used (TCP or UDP) on improving the detection rate.

In the traffic identification scenario (Traffic Profiling), we labeled our data as Eepsites, I2PSnark, and jIRCii. This way the traffic of one application includes the behavior of the traffic of multiple users using the same application. The important difference in this part is that when we run an application, for example I2PSnark, we intentionally label all the tunnels (exploratory, shared client, and participant if any) as I2PSnark. This way we can test if the overhead of the exploratory tunnels and the participant tunnels will affect the ability to distinguish the application type.

In the user identification scenario (User Profiling), we labeled our data as Machine 1, Machine 2, and Machine 3, since each machine was used by only one user. In this case, the Machine 1 traffic will include the I2PSnark, jIRCii, and Eepsites generated from Machine 1. The same applies on Machine 2 and Machine 3. The purpose of combining different traffic from each machine into one class is to mimic the user behavior on using multiple applications. Subsequently, measuring the ability to analyze the I2P users' behaviors.

On the I2P network, the traffic could be in the form of TCP or UDP Traffic. Therefore, we also include the separation of the traffic based on the protocol in both scenarios (the traffic and the user profiling) and on both bandwidth cases.

The following summarizes the results of both scenarios in addition to the effect of the protocol separation on the test data:

1) With Bandwidth participation

Table IV. shows the accuracy per class for the Traffic and User profiling when the amount of shared bandwidth is 80%, this is the default case on the I2P network. The accuracy measures the percentage of correctly classified instances out of all instances. It should be noted here that even though we do not use any IP addresses and port numbers in our analysis, we can achieve 80% - 86% accuracy for differentiating one user from another. However, it seems like differentiating traffic behavior in terms of protocols is much more challenging. We hypothesize that this may be due to two main reasons: (i) many different application behaviors are bundled up together in each of TCP and UDP traffic tunnels; and (ii) I2P garlic routing approach is better in anonymizing the protocol behaviors. In this case, further analysis is necessary to study the effect of each component.

TABLE IV. SUMMARY OF TRAFFIC AND USER PROFILING PERFORMANCE

		Number of Instances (flows)	**Accuracy (%)**
80 % Bandwidth Participation	***Traffic Profiling***	190,000	47.4
	Traffic Profiling – TCP Only	61,453	61.7
	Traffic Profiling – UDP Only	128,547	56.3
	User Profiling	189,906	81.8
	User Profiling – TCP Only	62,882	86
	User Profiling – UDP Only	127,024	79.8

TABLE V. SUMMARY OF TRAFFIC AND USERS PROFILING PERFORMANCE WITHOUT BANDWIDTH SHARING.

		Number of Instances	**Accuracy (%)**
0 % Bandwidth Participation	***Traffic Profiling***	195,081	73.7
	Traffic Profiling – TCP Only	40,075	65.6
	Traffic Profiling – UDP Only	155,006	75.7
	User Profiling	195,081	66.7
	User Profiling – TCP Only	40,075	81.7
	User Profiling – UDP Only	155,006	63.2

2) Without Bandwidth participation:

The configuration we used in our experiments in sections III (A, B, and C) was by activating the default bandwidth configuration (300 KBps In, 60 KBps Out) of an I2P client. Under this setting, the bandwidth participation is 80% which equals to 48KBps. To observe and study the effect of this amount of participation on the anonymity, we configured this bandwidth participation parameter on the I2P client to 0%. In both cases, the floodfill was disabled. Table V. presents the results of our analysis for the traffic and user profiling when the bandwidth participation is set to 0% and effectively not allowing any bandwidth sharing. In this case, while the user profiling drops by ~15%, traffic profiling increases by ~20%. Intuitively, this was expected because under no traffic sharing finding patterns in the tunnels is more likely to happen. However, under the same conditions differentiating users / machines without using IP addresses and port numbers is more challenging.

C. Clustering Tunnels Using SOM

Based on our analysis in section IV part A and B, the classification of tunnels seems to be more challenging than the classification of users. Also the confusion matrixes of our classifiers show that there is an overlap between the tunnels classes. Therefore, we employed an artificial neural network based an unsupervised learning algorithm, namely Self-Organization Map (SOM) [18] to cluster and visualize the different patterns (if any) that may exist in the data of the tunnels captured in this research. For this purpose, we used the Matlab [19] SOM toolbox [20]. Fig. 1 presents the visualization of SOM Clusters (groupings) on our data consisting of four classes: I2PSnark, jIRCii, Eepsites, and Exploratory & Participating Tunnels. In this figure, you can see the four clusters in four different colors. SOM is an unsupervised learning technique, therefore no labelled data is used during the training phase. However, we used the labels post training to analyze the performance of this clustering algorithm on our data sets. Fig. 2 shows the hits of the four classes, post training, on the SOM Map introduced in Fig. 1. This means, we projected the instances of the labeled data on Fig. 1 to obtain Fig. 2. The ideal case is when each class is represented by a separate cluster on the map which means that the map has good representation of the data. In Fig. 2, we have one cluster, the yellow hexagons, representing I2PSnark tunnels. We have another cluster, magenta hexagons, representing the Exploratory & Participating Tunnels. The third cluster shown in red represents the hits of the jIRCii tunnels on the Map. The green ones represent the Eepsites hits on the map. Based on how these clusters are distributed on the SOM, the Eepsites data flows seem to overlap with the Exploratory & Participating Tunnels data flows, namely, magenta ones. Thus, based on the SOM output, the Eepsite and the Exploratory & Participating tunnels (green and magenta) seem to be grouped together.

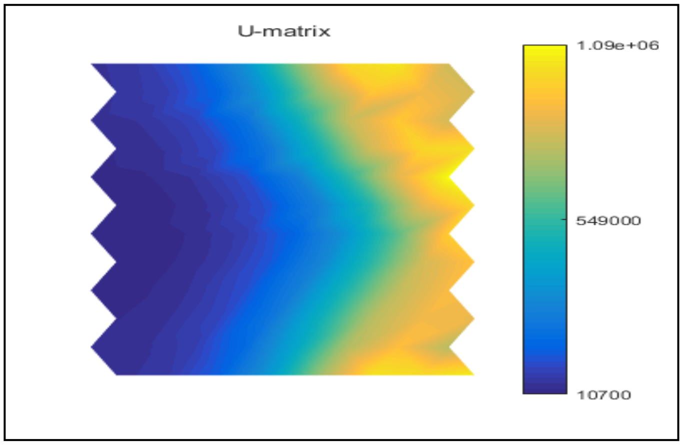

Fig. 1. Tunnels on the SOM Map – “sheet” shape

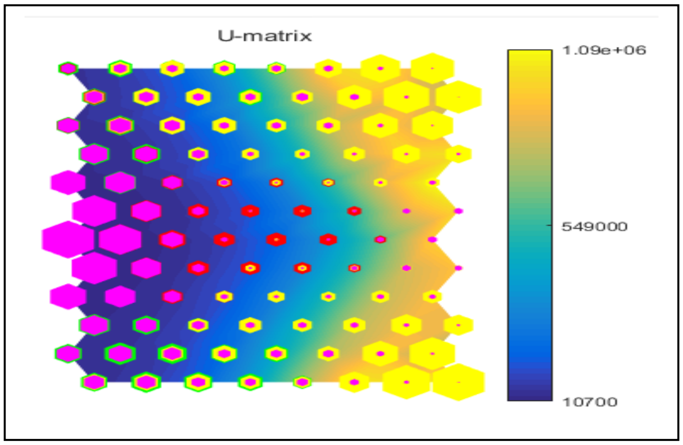

Fig. 2. Hits on the SOM Map for all Classes.

Actually, this matches with how the I2P tunnels are used. The I2PSnark and the jIRCii both use separate tunnels (I2PSnark & Irc2P Tunnels). The client Tunnels are the tunnels that are used for the Eepsites. Therefore, in Fig. 3 we grouped the Eepsites with the Exploratory & Participating tunnels to form one class.

V. Discussion

When we collected the data, we used the information of the client tunnels to label the data for a better level of accuracy. For example, if we are running jIRCii and we are connecting to a participant in one of our inbound or outbound tunnels and we label that participant for IRC traffic, that does not mean that participant will not be part of any other tunnels, for example one of our client tunnels for shared clients (DSA). Indeed, this adds a challenge to the data analysis problem we undertake in this research.

In our analysis, we do not use the IP addresses, port numbers and the protocol features. Therefore, when we combine both transport layer protocols (TCP and UDP) in the data set, the accuracy of our analysis drops. This is expected, because in network traffic analysis, transport layer protocol filters are shown to be useful [31]. So this means that when in real life, the protocol feature is used in the analysis, the accuracy will increase. However, our data collection network is small so the data set only involves three machines / users. In short, any classifier using the protocol, port number and IP address features will be 100% accuracy but will be very specific to our network. Therefore, it will not generalize well to larger networks where more machines and protocols exist.

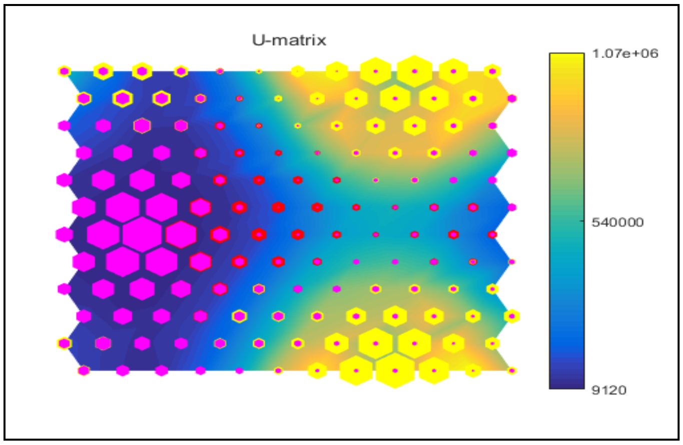

Fig. 3. Hits for the Merged Eepsites and Exploratory & Participating tunnels- “cyl” shape

The resource sharing (bandwidth participation) increased the anonymity level when profiling the applications. The shared client tunnels are used for Eepsites application. They could be also configured to be used for other applications. The default is to use client tunnels for Eepsites, while separate tunnels are used for I2PSnark and Irc2P. Furthermore, when application tunnels are grouped as one class and the Exploratory tunnels in another class, this increased the accuracy of profiling the applications. Therefore, we think that forcing all the applications to use the client tunnels will improve the users’ anonymity on the I2P network. On the other hand, based on our experiments, increasing the bandwidth participation improves the ability to profile the users. When the user allocates more resources to participate on the network that means more traffic flows on the network belong to the user. This seems to enhance the profiling of the users. Therefore, we think that decreasing the bandwidth participation (but still keeping it more than 50%) will improve the users’ anonymity on the I2P network. However, more analysis on bigger data sets is necessary to get better understanding of such user behaviors.

Moreover, the unsupervised learning algorithm SOM shows that the Eepsites tunnels tend to have similar behaviors with the exploratory and participating tunnels. Therefore, when the Eepsites tunnels are merged with the exploratory and participating tunnels to find the hits on the SOM, it showed more consistent behavior. This reinforces that the separation between the tunnels for different applications seems to enhance the applications profiling. Therefore, changing the default setting on the I2P client to force applications such as IRC to use the “shared clients” tunnels hardens the applications profiling. Consequently, improving the anonymity level.

VI. Conclusion and Future Work

The I2P network works differently from other anonymity networks such as Tor [9] [30] and JonDoNym [2] in terms of its design which is based on the private network approach. The connection of the users to the I2P network is not hidden, but the users’ activities within the network (type of applications) are supposed to be anonymous. Based on our analysis on the I2P data, the resource sharing (bandwidth participation) of the users on the I2P network improves the anonymity level of the users. On the other hand, using the default setting for not using the shared client tunnels for all applications seems to reduce the anonymity level and enables the applications profiling ability of a potential attacker. For future work, we will expand our research to study effects of the bandwidth on the I2P network on a larger scale. This will include more types of applications (or plug-in) and more users.

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: the second-generation onion router,” in *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*. USENIX Association, 2004, pp. 21–21.

[2] Project: AN.ON – Anonymity [Online]. Available: http://anon.inf.tu-dresden.de/index_en.html.

[3] The Invisible Internet Project (I2P) [Online]. Available: https://geti2p.net/en/

[4] J. Timpanaro, I. Chrisment, and O. Festor, “Monitoring the I2P Network,” 2011.

[5] C. Egger, J. Schlumberger, C. Kruegel, and G. Vigna, “Practical Attacks Against the I2P Network,” In the Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2013), October 2013.

[6] P. LIU, L. WANG, Q. TAN, Q. LI, ,X. WANG, and J. SHI, “Empirical Measurement and Analysis of I2P Routers”. Journal of Networks, North America, 9, sep. 2014.

[7] M. Herrmann and C. Grothoff, “Privacy Implications of Performance-Based Peer Selection by Onion Routers: A Real-World Case Study using I2P” In the Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 2011), Waterloo, Canada, July 2011.

[8] M. AlSabah, K. Bauer, and I. Goldberg, “Enhancing Tor’s performance using real-time traffic classification,” in Proceedings of the 2012 ACM conference on Computer and communications security, Raleigh, USA, 2012, pp. 73-84.

[9] K. Shahbar, and A. N. Zincir-Heywood, “Benchmarking two techniques for Tor classification: Flow level and Circuit level classification,”. in IEEE Symposium on Computational Intelligence in Cyber Security, 2014.

[10] *I2P: A Scalable Framework for Anonymous Communication* [Online]. Available: https://geti2p.net/en/docs/how/tech-intro#intro

[11] *I2P: The Network Database* [Online]. Available: https://geti2p.net/en/docs/how/network-database

[12] *I2P: Tunnel Implementation* [Online]. Available: https://geti2p.net/en/docs/tunnels/implementation

[13] *iMacros* [Online]. Available: http://imacros.net/overview

[14] *I2P: Plugins* [Online]. Available: https://geti2p.net/en/docs/plugins

[15] *TRANALYZER2* [Online]. Available: http://tranalyzer.com/

[16] J. Ross Quinlan, C4.5: Program for Machine Learning, Morgan Kaufmann Inc., San Francisco, CA, 1993.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten,”The WEKA data mining software: an update,” SIGKDD Explorations, vol. 11, no. 1, pp. 10-18, 2009.

[18] T. Kohonen, (2001). Self-organizing maps (3rd ed.). Berlin, Germany: Springer-Verlag.

[19] *Matlab* [Online]. Available: https://www.mathworks.com/products/matlab/

[20] *SOM Toolbox* [Online]. Available: http://www.cis.hut.fi/somtoolbox/

[21] *I2P Reseed Hosts* [Online]. Available: https://geti2p.net/en/docs/reseed

[22] *jIRCii: The Ultimate IRC Client* [Online]. Available: http://www.oldschoolirc.com/

[23] *I2P: Supported Applications* [Online]. Available: https://geti2p.net/en/docs/applications/supported

[24] *I2P: I2PSnark* [Online]. Available: https://geti2p.net/en/docs/how/tech-intro#app.i2psnark

[25] *I2P: Unidirectional Tunnels* [Online]. Available: https://geti2p.net/en/docs/tunnels/unidirectional

[26] *I2P: Peer Profiling and Selection* [Online]. Available: https://geti2p.net/en/docs/how/peer-selection

[27] *I2P: Frequently Asked Questions* [Online]. Available: https://geti2p.net/en/faq#eepsite

[28] *I2P: Naming and Addressbook* [Online]. Available: https://geti2p.net/en/docs/naming

[29] *TCPTRACE* [Online]. Available: http://www.tcptrace.org/

[30] K. Shahbar and A. N. Zincir-Heywood, "Traffic flow analysis of tor pluggable transports,“ Network and Service Management (CNSM), 2015 11th International Conference on, Barcelona, 2015, pp. 178-181.

[31] F. Haddadi and A. N. Zincir-Heywood "Benchmarking the Effect of Flow Exporters and Protocol Filters on Botnet Traffic Classification," In IEEE Systems Journal, 2014.