Weighted Factors for Measuring Anonymity Services: A Case Study on Tor, JonDonym, and I2P

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

***Abstract*— There are many systems that provide anonymity service for the users. Most of these systems work on the separation between the users' identity and the final destination. The level of anonymity these services provide affected by several factors. Some of these factors are related to the design of the anonymity service itself. Others are related to how the system is used or what is the application/purpose the user wants to run over the anonymity service. In this paper we: (i) propose five factors that aim to measure the anonymity level from the user's perspective; (ii) evaluate these factors on three anonymity services, namely Tor, JonDonym, and I2P as case studies; and (iii) present a mechanism to evaluate anonymity services based on our factors and measure the level of anonymity.**

Keywords— Anonymity Factors; Metrics; Tor; JonDonym; I2P

I. INTRODUCTION

There are many tools, applications, and websites on the Internet claiming to provide services to protect the privacy of the Internet users. The level of provided privacy protection for these services is different based on the way they work. For example, VPN (Virtual Private Network) which can be provided as a free or a paid service, hides the user identity to surf the Internet anonymously. At the same time, the service provider has access to the user identity and his/her activity on the Internet. Some of these service providers also keep the log records of the users. This is also the case on the free proxy websites, which claim that they protect the user's identity.

In fact, a user's privacy in such services depends on the amount of trust the user has for these service providers. On the other hand, there are other systems that claim to provide anonymity service without the possibility to log or know the user activity. They relay the user connection to the final destination (such as web server) via multiple stations. The design of such systems aims to prevent the stations from making a link between the user request and the final destination.

Tor, JonDonym, and I2P are popular anonymity services. They provide anonymity to their users to hide their identity from Internet web servers and hide the websites they accessed. These systems prevent not only the webservers from revealing the user identity but also the operators of the systems themselves from identifying the users. However, there are lots of details behind this kind of anonymity that might not be clear or obvious to the user. For example, changing the default setting in some of these systems' browsers could lead to a breach concerning the user anonymity. These systems provide the anonymity and at the same time give the user the ability to customize the settings of the anonymity system to control the level of anonymity. For example, JavaScript by default could be enabled or disabled in these systems based on which system is used. Many websites require that JavaScript to be active to show the website contents properly. The user has the ability to enable JavaScript which might conflict with how these anonymity services work. Tor and JonDonym have their own browsers which are modified to ensure the users' privacy. However, the user has the ability to use any other available browser of his/her choice and the ability to configure it to work with these anonymity services. In this case, it is the user's responsibility to ensure the proper configuration that guarantees not to break the anonymity. Even with a proper configuration or using the default browser, the privacy and anonymity of the users is not only about the setting but it includes also the user behavior and the tools employed.

Browser fingerprinting is one of the examples of how the anonymity of the user could be broken. The browser itself could provide considerable information about the user environment and consequently his/her identity. This type of information is obtained from the HyperText Transfer Protocol (HTTP) that is used for the communication between the web browser and the web server. The HTTP Header of this protocol contains information about the browser name, version, operating system, language, and other information. For example, enabling cookies, could lead to the storage of third-party cookies, which then could provide the ability to track the user by the web sites he/she visited. Browser fingerprinting is not limited to this only, there are many studies about the application of different methods to implement browser fingerprinting [1] [2] [3].

In [4], Eckersley collected a sample of 470,161 browsers that visited the website: http://panopticlick.eff.org. A fingerprinting algorithm was then applied to the sample based on the information available in the http request field (stored in the web server access log files), and the JavaScript running in the browser to test the ability to define how unique the browsers were. The results showed that the browser fingerprinting was possible with a promising performance especially when the browser supports Flash or Java utilities.

Therefore, the anonymity level of the users is not the same even when using an anonymity tool. The reason/goal behind using an anonymity service varies from one user to another. This variation of the goal could affect the anonymity level and the choice of the right anonymity service to achieve this goal. The design of the anonymity tools varies based on: (i) Which services such a tool could offer to the users, (ii) How could the user decide/measure the anonymity level given all the different anonymity services? In this paper, we aim to answer these questions. To this end, we present a method to calculate/compare the user's anonymity level that takes into consideration the different needs for anonymity of different users. Therefore, we aim to assist the user to choose a suitable anonymity service for his/her needs. The proposed method depends on evaluating the anonymity systems based on five factors. To measure the anonymity level using this method, the factors are converted to numeric values to be able to assign weights and scores. In addition, each factor is compared with the other factors according to the anonymity case (the goal/purpose). Therefore, the weights (importance) of the factors are determined based on the case (who is using the anonymity service and how / which). In doing so, our objective is to provide a comprehensive measurement technique that could be used to evaluate the level of anonymity based on what environment the anonymity service is used.

The rest of this paper is organized as follows. Related work is reviewed in section II. The Tor network, the JonDonym network, and the I2P network are discussed in section III. Section IV presents and discusses the five factors regarding the level off privacy in anonymity services studied in this work. Section V shows the evaluation of these anonymity factors. Finally, conclusions are drawn and future work is discussed in section VI.

II. Related Work

Researches on studying the anonymity networks cover different subjects related to these networks. This includes the design of the anonymity networks, the threat models and possible attacks, the performance, and the analysis of the networks usage. Measuring the level of anonymity that the anonymity network may provide is also one of the concerns on the research field. Measuring the anonymity level is a challenge due to multiple reasons. One of them is the difference in the design and the goal of the anonymity network. On the other hand, there is not one way to measure the anonymity level on the anonymity networks. In addition, the anonymity level is not directly quantifiable compared to other network traffic measurements such as delay, bandwidth, volume etc.

In [5], Ries et al. evaluated five anonymization tools with regards to the performance, the usability, the anonymity, the network reliability, and the cost. The evaluated tools were: Tor, I2P, JonDonym, Perfect Privacy and Free proxies. They defined performance factors to evaluate and rank these tools. Performance factors used to evaluate these tools were the Round Trip Time (RTT), the Inter-Packet Delay Variation (IPDV), and the throughput. Moreover, the authors used the installation, the configuration, and the verification of the anonymization connection as factors to define the usability of these tools. The anonymization of the tools was evaluated by using a ranking for the ability of the adversary to perform de-anonymization attacks against the tools. It should be noted here that these evaluations were limited to the specific scenarios. The network reliability was measured using the failure rate. The Mean Time Between Failures (MTBF) and the Mean Time to Recovery (MTTR) were used to measure the failure rate.

In [6], Abou-Tair et al. examined the usability for four anonymity tools (Tor, JonDo, I2P, and Quicksilver) during the installation phase. They detailed the installation process of these tools. They applied four tasks to test the installation phase: the success of installation, the success of configuration, the confirmation of the anonymization, and the ability to disable the anonymization. To test the usability of these tools, the authors used eight guidelines taken from [7]. The guidelines focused on the user's ability to perform the four tasks mentioned above.

In [8], Wendolsky et al. compared Tor and AN.ON (JonDonym) from the user's perspective based on the performance and the number of users. Latency and bandwidth were used to measure the performance. Their results showed that Tor has unpredictable performance based on the time of the day. In contrast, AN.ON (JonDonym) had more consistent performance.

The above studies focused mainly on evaluating the anonymity services based on their performance or usability where anonymity was not the focus in the evaluation. On the other hand, there are studies where measuring anonymity was the main goal. The raise of the idea of measuring the anonymity is synchronized with the proposed ideas to develop the anonymity by passing the message between the sender and the receiver through multiple stations until it reaches the final destination (the Mix concept) [38]. This concept aims to separate the ability for the attacker to link the sender and the receiver even if they communicate over an observed channel by the attacker. To anonymize against such a threat model, Chaum [39] presented the concept of "anonymity set" where the set is the total number of participants in the anonymity service, which the sender might be one of them. When the size of the set is increased, the anonymity level is considered to be increased, too. Consequently, if the size of the anonymity set is one then there is no anonymity and the sender is identified by the attacker.

Serjantov and Danezis [40] developed the concept of anonymity set by using the information-theoretic metric based on the anonymity probability distribution.

Diaz et al. [35] also used information theoretic model to evaluate the anonymity level of an anonymity system in a particular attack scenario. The model aimed to evaluate the anonymity level of a system by finding the level of information the attacker can statistically gain to relate a user in the anonymity system to his messages. Shannon's definition of entropy is used to calculate this gain.

Murdoch [36] surveyed studies performed on measuring anonymity for low-latency anonymous networks and high-latency email anonymous networks and discussed the development in the techniques used for measuring anonymity.

There are other studies such as [37] [41] that also evaluate the anonymity level of the anonymity service from the perspective of the possibility to link the message within the anonymity service with the sender.

Even though, the above studies are important and significant in measuring anonymity level, measuring anonymity could be analyzed from a different perspective other than looking into the possibility to link the message with the sender among the anonymized users. There are other factors that affect the anonymity level. For example, the user behavior and the browser setting that the user is using to browse websites on the Internet. Even the link between the user and the final destination varies in the theoretical ability to achieve based on the design of the anonymity service itself which is different form one system to another. Therefore, in this paper, we aim to measure the level of anonymity by aiming to analyze the anonymity service from different perspectives and propose measurable metrics (factors) that enables the quantization of the anonymity of such services.

III. Anonymity Systems Studied

Multilayered encrypted anonymous networks share the goal of providing anonymous services to their users. The provided anonymity services vary from one service to another in terms of design, performance, delay, and provided services. The following introduces the most popular anonymity networks: Tor, JonDoNym, and I2P.

The Tor network is based on volunteers to run their machines as Tor relays (also called routers or nodes) [9]. Tor provides anonymity to its users by hiding the IP addresses of the users and by hiding the content of the users' traffic as long as the traffic is still on the Tor network. The IP addresses of the users are hidden by relaying all the users' requests through the Tor network. The users' traffic are hidden by dividing the packets into smaller fixed size encrypted Cells. Tor also provides a service called Hidden Services that hides the IP address of a web server for the users who run web servers and like to keep their identity hidden.

There are three types of Nodes on the Tor network. These are: Entry node, middle node and exit node. The entry node is the first node that the user communicates with, when trying to establish a circuit that carries his/her traffic. The middle node is an intermediate node that lies between the entry node and the exit node. The exit node is the node used to relay the user's request to the web server. Therefore, the IP address of the exit node is the IP address that appears in the web server log. Since all three types of nodes are run by volunteers, running an exit node is an optional choice available while configuring the node to run into the Tor network. The exit node has the option to be configured for allowing certain types of traffic based on the port number. This configuration gives the volunteered user, who runs the exit node, the choice to allow the type of traffic to go through the exit node and block everything else.

Whenever the user sends his/her traffic through Tor, a virtual circuit is used to relay the user's traffic. The virtual circuit consists of a connection of the three types of nodes (entry, middle and exit nodes). The user starts by establishing a TLS (Transport Layer Security) connection with the first node. After the connection is made to the first node, the user requests the entry node to extend the connection to the middle node. Finally, the connection is extended again to the exit node. The virtual circuit is ready when the connection reaches to the exit node. Tor browser is responsible of translating all the user's requests to the virtual circuit. This includes hiding the IP address of the user, dividing the packets into smaller size Cell(s), encrypting the traffic with three layers of encryption, receiving the data from the web server that comes in encrypted Cells and Decrypting the received Cells. The Cell is the building block in Tor. The user's data are divided into small fixed size Cells. The Cell size is 512 bytes. It consists of two parts; header and payload. The header contains information about the circuit that the Cell belongs to and the type of Cell. There are two types of Cells. Control Cells and Relay Cells. The Control Cells direct the relays what to do with the Cell. The Relay Cells contain the user's data. The structure of the Cell's header varies based on the type of the Cell. The header of the Cell consists of the circuit Identity ("CircID") and a command. The command type distinguishes between the control Cell and the relay Cell. Commands like "CREATE", "CREATE_FAST", "CREATED", "CREATED_FAST", "PADDING", "DESTROY", "NETINFO", "CREATE2" and "CREATED2" are control commands. They used to manage the connection inside Tor. The payload of the control Cell contains different information based on the type of the command. The path that is used to establish a circuit is controlled by the path selection protocol [25]. The circuit path is not the same each time the user makes a connection. The bandwidth of the nodes and the policy of the exit node play a major role in selecting the best circuit path.

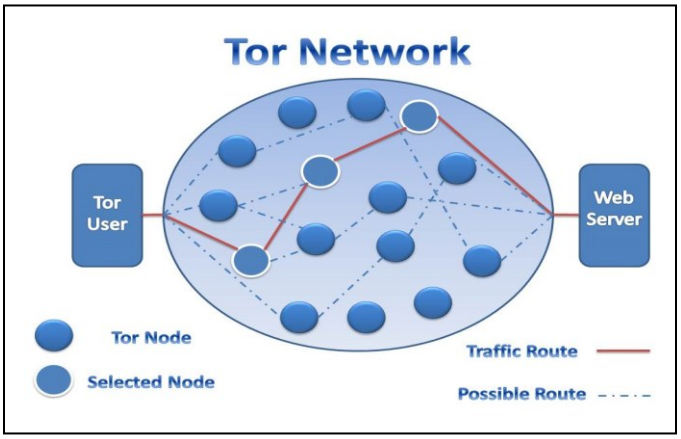

Fig. 1. Path selection on the Tor Network.

Figure 1 shows how the circuit path could take different paths according to the path selection protocol. The circuit will be used for a short period then another circuit is created. The user has the option to fix the entry node and/or the exit node.

JonDonym/AN.ON is a network of mix cascades to provide anonymity to the users [10]. It provides anonymity based on multilayer encryption. The cascade consists of two (free) or three (paid) mix servers. The user starts the connection to the JonDonym network by selecting the mix cascade. Currently there are five free cascades and eleven paid cascades the user can choose from. JonDo previously known as JAP is the client software which connects the user to the JonDonym network.

Only one active connection to one cascade is possible during the user's connection to the JonDonym network. Each HTTP request will create a connection from the browser (JonDoFox) to the client software JonDo. The JonDoFox browser could create multiple connections to the JonDo. All these connections are multiplexed into one connection to the first mix server. The first mix receives connections from multiple users. All the users' connections then multiplexed into one TCP/IP connection to the second mix or to the last in case of only two mixes in the cascade.

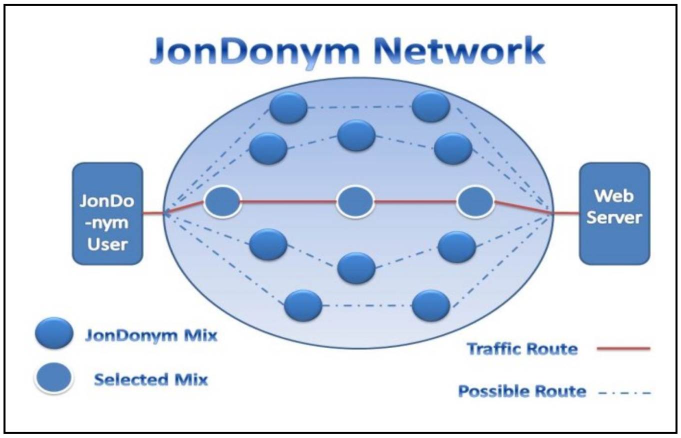

Fig. 2. Cascades on the JonDonmym Network.

The information about the available cascades, the number of users, the loads, and the mix status are stored at the InfoService [31]. The user gets the information about the cascades from the InfoService. The last mix sends the users' requests to cache proxies. Multilayer of encryption is used during the communication between the user and the last mix. The encryption ensures that even the mixes cannot access the user's data. The path that the user's data take is fixed based on the chosen cascade. To choose another path (cascade), the user has to start a new connection to the JonDonym. The user can only have one connection to one cascade at any given time. Figure 2 shows that a user is connecting to one cascade which has fixed path. It also shows that there are other possible paths (cascades) that the user can select to connect to.

When the connection is established, the IP address that is visible to the websites is the IP address of the last mix. JonDo and mixes use fix size packets called MixPacket. The first mix receives multiple packets from multiple users. These packets then multiplexed and send to the second mix. The MixPackets size is 998 bytes. It consists of 4 bytes for the channel ID, 2 bytes for flags, and 992 bytes for the data field. The data field is readable only at the last mix. It contains 2 bytes of information about the length of the data, 1 byte of information about the type, and 989 bytes for the payload.

I2P network is a decentralized anonymous network; there is no central database or server that contains the network database. The network database (netDb) is distributed by using Kademlia algorithm [33]. The algorithm is used in many applications where peer to peer (P2P) communication is needed in a decentralized network. The information that the user gets from the netDb is what enables the user to build tunnels. Communications over I2P require inbound and outbound tunnels. These tunnels are unidirectional where messages only transfer in one direction. The netDb contains the leaseSet of the tunnels and routers. leaseSet shows the routers involved in a tunnel. RouterInfo in the netDb shows how to contact a specific router. The user has the option to modify the number of routers in the outbound tunnel. I2P uses the concept of garlic routing [34] where layered encryption is implemented in addition to binding multiple messages together. The messages within the I2P network are encrypted end-to-end as long as the two communication parties are within the I2P network. However, when the user communicates with an end-system that is outside of I2P network using an outproxy, then the encryption is not an end-to-end encryption.

By default, the user within the network transfers his/her data and the data of other users where the user's machine contributes as a resource to the network. The user can change the amount of bandwidth dedicated to the network from the user's console. The users' contributions in passing the network data are restricted by passing the data within the I2P network only. A different configuration is required in the case where a user wants to pass the I2P traffic to an end-system outside of the I2P network (outproxy). The number of outproxies in the I2P network is limited.

One of the major differences between I2P and other anonymity networks such as Tor and JonDonym is that I2P is designed as a private network. The users mainly communicate within the network. The user builds two tunnels, inbound and outbound tunnels. The inbound tunnels are used to receive messages and the outband tunnels are used to send messages. Figure 3 shows the inbound and outband tunnels on the I2P network.

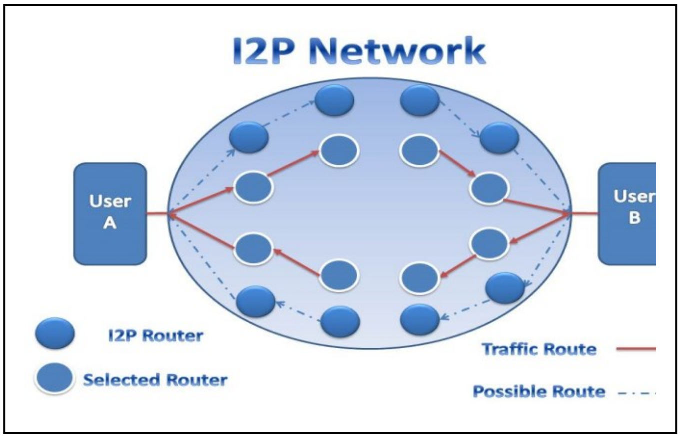

Fig. 3. Inbound and Outband tunnels on the I2P network.

Based on the above, Tor, JonDonym, and I2P networks have similarities and differences. The three anonymity services share the goal to provide anonymity by relaying on passing traffic to multiple stations using multiple layers of encryption. The multiple layers of encryption is used to harden/deny the link between the user and his/her messages. Tor and JonDonym mainly focus on providing anonymity for the users to access websites that are outside of their network. On the other hand, I2P provides anonymity to access websites hosted privately within the I2P network itself. At the same time, Tor also has websites hosted within the Tor network (hidden services). Moreover, I2P supports access to websites hosted on the Internet not on the I2P network via using outproxy. In terms of the path used to relay the traffic, the path on the Tor network and the I2P network changes and not fixed, while the path on the JonDonym network is fixed. The duration that the user will stay connected to one path (circuit – tunnels – cascade) is different based on the anonymity system. The routing technique and the path selection also have differences between the three anonymity services. The differences in the design and the main goal that the anonymity service aims to achieve entail different threat models for each one of them. Therefore, measuring the anonymity for such systems requires a criterion that includes the differences existing among these systems in a way that captures this variation when measuring the anonymity level. The following section introduces the proposed factors that could be used to compare and measure the anonymity level for such services.

IV. PROPOSED FACTORS

In this section, we present the five proposed factors to analyze the anonymity level of the aforementioned anonymity systems.

A. The Level of information available for the service provider

When a Tor user starts to connect to the Tor network, a virtual circuit is created. The circuit consists of three nodes; the first node has the actual IP address of the user, therefore his identity. But it does not have the knowledge about his Internet activity. Tor uses the concept of (Entry Guard) which means assigning the Tor user who wants to connect to the Tor network to a specific node. This node acts as a permeant gate to the Tor Network for this user. The goal of this process is to increase the privacy of the users in the case of the existence of a compromised node by an attacker. By using this process, the probability for the user to be among the users of such a compromised node becomes low. On the other hand, using an Entry Guard provides other information about the users to the operator of the Entry node. For example, the Internet browsing habit of the user such as time of the day, online duration, and the amount of data transferred could be obtained by the operator. This amount of information could be used to perform attacks that depend on the correlation between the duration, data, and the server. The exit nodes through which all the requests of the users pass, have considerable amount of information. They are the link between the Internet and the Tor users. The operator of the exit node has the ability to know and statistically evaluate the user's activities on the Tor network. For example, McCoy et al. provide percentages of the Tor Internet activities based on the exit node observations in [30]. Another important fact about the exit node that might not be clear for non-technical users is that the encryption of the requests through the exit node all are based on the encryption of the original requests and has nothing to do with the three levels of encryption on the Tor Network. Therefore, the exit node alone can totally break the anonymity of the users if they used their login information to access their email or any web server without using an encrypted request.

Furthermore, JonDonym works in a similar way to Tor in terms of connecting the user with the requested destination without revealing the user information to the destination. The first point on the JonDonym network (First Mix) receives the connection request from the user. It has the information about the connection duration and the user's identity. The last point (Last Mix) does not know about the user's identity but it has the activities or the websites that the users request. Even though the user data pass through several mixes, the operators of these mixes do not have the ability to access the contents of the data. The encryption layers used by JonDonym and Tor overlay networks protect the data even from the operators. The exception is when the data sent by the user to the webserver are not encrypted then the last node/mix has the ability to access the data sent by the user. The anonymity mechanism in Tor/JonDonym depends on relaying the user data through multiple points (Node/Mix). Each node/mix only knows part of the connection information not the whole information required to relate the user to the request for the webserver. This way, the assumption is that even a compromised mix/node will not be able to find the whole connection information.

On the other hand, what if all the nodes/mixes on the path between the user and the server are compromised or attacked? On the Tor network, the three nodes in the circuit path are selected by using the path selection protocol [25]. The protocol specifies the three nodes the user will use to relay the data in conjunction with the policy that the exit node operator defines. When the user's request does not match with the exit node policy, the path selection protocol finds another exit node that permits such traffic. For example, an exit node might allow only port 80. In addition, the user has the ability to override the path selection protocol and to choose a specific exit/entry node. In this case, the chosen exit node will not change for any circuit created by the user. This flexibly and randomness in node selection makes it harder for an attacker to target a specific user by trying to compromise the three nodes that the user selects. On the other hand, it might be possible to compromise a node on the Tor network. Potentially, volunteering to run a node on the Tor network does not require information about the operator more than the IP address and the nickname. However, running and compromising three nodes do not mean that these three nodes will be selected by the path selection protocol.

On the JonDonym network, this type of attack is also possible; the difference is the operation of the mixes. The number of mixes on the JonDonym is much less than the number of nodes on the Tor network. On the other hand, the operators of the mixes are registered with their identities. They also sign an agreement with JonDonym that confirms not to exchange information between operators of the mixes and not to save users data. One of the differences between Tor and JonDonym is that JonDonym mixes do not change. The path is always the same. In the case of cooperation between all the mixes, it is possible to break the user anonymity on the JonDonym network.

Last but not the least, the goal of I2P network is different than Tor and JonDonym. I2P is designed to provide anonymity for the users within the I2P network. However, that does not mean that I2P services are limited within the network boundaries. Browsing webpages outside the I2P network requires configuring the user's machine to use an outproxy. In this case, the information available to the outproxy is similar to Tor's exit router or JonDonym's last mix. The outproxy can has access to all traffic passing through. If the traffic is not encrypted then the outproxy can see sensitive information.

The common point between the three anonymity services is that at any point during using the service, there is part of the network that has some kind of information about the user. This information could be the IP address of the user that is available to the first point on the anonymity network that connects the user to the network. Or, it could be the amount of traffic that the last point can see when sending the traffic to the final destination. Thus, the difference in the design of the anonymity service regarding how to relay the traffic is what effects how difficult to link the user with his/her traffic.

B. Blocking Anonymity And Obfuscation options

The anonymity systems could hide the user activity on the Internet but could not always hide that such a system is in use. Sometimes using anonymity systems might raise questions about why such a system is in use. In some countries, the IP addresses of the hosts running such systems are blocked to prevent the access to such networks.

On the Tor network, a bridge [11] is a special node (host – router) connecting the user with the Tor network. The IP address of the bridge is not announced like the Tor nodes. The user sends an email to the Tor network (bridges@torproject.org) to get an IP address for a bridge. The user could also use the bridge database website to get the IP addresses (https://bridges.torproject.org/). The Tor network provides the user with the IP addresses of three bridges during a 24-hour period. This is to prevent the censorship organizations from obtaining all the IP addresses and blocking them.

Furthermore, pluggable transports [12] work as an interface between the Tor user and the Tor network. The user connects to a pluggable transport which sends the connection request to the Tor network on behalf of the user. The purpose of using the pluggable transports is to hide the connection between the user and the Tor network. There is more than one pluggable transport tool available for the Tor users to choose. These tools work differently using different techniques to resist the different blocking methods.

In addition to blocking Tor by blocking the IP addresses of the nodes [13], there are cases where Tor service is blocked by other techniques. The encryption in the Tor network is based on using TLS between the communication parties; the user to the first node – the first node to the second node and so on. Therefore, fingerprinting the Tor TLS is one of the blocking techniques. Another blocking technique is Deep packet inspection (DPI). DPI is used to find a pattern that recognizes Tor. To this end, the handshake phase in establishing a TLS Tor connection could be used to identify Tor. Therefore, changing the content to look like something other than a TLS is used by some of the pluggable transports to hide the connection to Tor. In fact, Obfs3 [14] is one of the pluggable transports that obfuscates the Tor TLS to look like random strings. It uses another layer of encryption to wrap the TLS handshake used by Tor. Even though Obfs3 aims to hide the TLS from an observer, the packet length and the timing of the packets are still the same as normal Tor connections. This is because Obfs3 mainly focuses on preventing the Tor TLS from being fingerprinted.

It is possible to intercept a TLS handshake to extract the destination IP address. In the case of Tor, this IP address is the bridge or the node IP address. After getting the IP address, the censorship can establish a Tor connection to this IP address. When a reply is received, it confirms that this IP address does belong to the Tor network. This active probing method is also used to find bridges and to block them [15] [16]. Scramblesuit [17] is one of the Tor pluggable transports. One of the goals of designing Scramblesuit was to prevent such active probing. To resist against active probing, a password and a ticket are used to connect to the Scramblesuit server. In the Tor network, the

Scramblesuit password is exchanged by requesting the password from the bridge database (email/website).

Blocking the IP address of a bridge prevents the Tor users from connecting to this bridge. Even though the IP addresses of the bridges are not announced, they could be discovered [18]. Flashproxy [19] is another pluggable transport that works around the IP blocking by using the IP addresses of the visitors of a website. These IP addresses change based on the IP address of the website that supports Flashproxy. When a website chooses to provide the Flashproxy service to the Tor users, it includes a JavaScript code which is activated when visitors access the websites. The code uses the websites visitors' browsers to pass the connection between the Tor user and the Tor relay. Therefore, the IP address of the Flashproxy always changes based on the IP addresses of the visitors of the Flashproxy supported websites. Accordingly, blocking these websites that support Flashproxy does not affect the ability of the Flashproxy to connect the Tor user to the Tor relay. Once the visitors leave the websites, their IP addresses are not used anymore. Yet blocking their IP addresses does not prevent the Tor users from connecting to the Tor relays because simply new website visitors will take over the connection task. This makes the blocking of the IP addresses challenging and less efficient. On the other hand, Flashproxy by itself does not work on changing the form or pattern of the connection to Tor. Rather it depends on the Obfsproxy framework to accomplish this.

The user needs to install Flashproxy client transport plugin (included in Tor browser bundle) and defines a specific port in the configuration to receive Flashproxy connections. When the user starts Tor, the client plugin sends an encrypted message to a facilitator containing the IP address of the user. The facilitator keeps track of all users, who need to communicate with the Flashproxies, and send their IP addresses to the Flashproxies. The client communicates with the facilitator indirectly by connecting to Gmail and sending a message. The facilitator then gets the IP addresses of the users from the server. This way blocking the facilitator does not prevent the user from contacting the facilitator. In order to prevent the user from communicating with the facilitator, services such as Gmail are needed to be blocked. This then makes it challenging for the censorship.

The distribution method of Flashproxy is through the websites of the volunteers. They install the FlashProxy and activate it when they get visitors to their websites. When the Flashproxy is activated on a volunteer's browser, it communicates with the facilitator, which provides the volunteer's browser with an IP address of the Tor user. The volunteer then sends a connection to the Tor user via the browser to the port that the Tor user is configured to receive the Flashproxy connection. Also, the volunteer's browser sends a connection to the Tor relay and starts to transfer the data between the Tor user and the Tor relay. Again, blocking the websites of the volunteers that host the Flashproxy will not prevent the Tor user from communicating with the Tor relay. This is because Flashproxy runs on the volunteers' browsers and the IP addresses of the Flashproxies are their own IP addresses.

Whitelisting is another method that could be used to block Tor. In this case, all the allowed traffic is profiled and anything that does not match with this (list) is blocked. Format-transforming encryption (FTE) [20] is a pluggable transport that takes a ciphertext and transforms it to another format that matches a regex. In the Tor case, FTE changes the Tor traffic to look like HTTP traffic. It generates HTTP regex out of Tor traffic that matches what DPI expects from HTTP traffic. Meek [21] is a pluggable transport that uses the concept of "domain fronting" which hides a Tor message inside an HTTPS request. Meek uses Google – Amazon – Azure for domain fronting to send Tor messages on behalf of a Tor user.

Last but not the least, network traffic flow analysis is another technique that could be used to detect Tor [32]. To evade the network flow analysis [1], Scramblesuit forms the traffic in a way not to resemble a specific shape (form). This includes the packet length and the interarrival time for every Scramblesuit server. For example, it changes the packet length distribution to mislead classifiers. This way each server has its own flow characteristics. To this end, the server starts by generating a 256-bit seed randomly. This seed is used in PRNG to generate two random distributions. The packet length is then changed by using a padding (0-1520 bytes) of random sampling from the distribution of all the packet lengths.

JonDonym has two options to bypass the blocking of the service. The first one is using TCP/IP forward method where the user will use encrypted connection to another user who has unblocked access to the JonDonym network. The speed and the stability will suffer when using such method. The connection also depends on the forwarder to stay alive. The second method is using Skype to tunnel the blockage of the JonDonym service. It is more reliable than using the TCP/IP forward method.

On the I2P network, there are not any obfuscation options similar to Tor pluggable transports. It is possible for an observer to collect the routers IP addresses. Harvesting attack [45] is an example of such an attack on the I2P network. Currently, I2P network has not developed any obfuscation option that could provide the users to connect to the I2P network if the network is blocked by using such an attack. However, the I2P network implemented other improvements in the design of the transport layer to harden the identification of the I2P network traffic. I2P employed random port numbers, point to point encryption, DH key exchange, and the use of both TCP and UDP. In addition, several obfuscation options are still considered by the developers of the I2P network. For example, these obfuscation options include using padding techniques at the transport layer to achieve random length, studying the signature of the packet size distribution, studying the technique used to block Tor.

It should be noted that anonymity services do not hide, in general, that the users are connecting to the service. So in a regular situation where the user is directly connected to the anonymity service, anyone who observes the traffic can notice that an anonymity service is in use.

In case of using any obfuscation option, then the observer who wants to de-anonymize the user needs to identify that the

[1] Flow is obtained by using the following five tuple: The source IP address, the destination IP address, the source port, the destination port, and the protocol.

user is connecting to the anonymity service in the first place. Therefore, the existence of such an obfuscation options is a factor we take into consideration to measure the level of anonymity.

C. Application and anonymity

The common way to use anonymity service is to use the default browser of the aforementioned services (Tor, JonDo, I2P etc.) to browse the web. However, these anonymity services could also be used with other applications not just web browsing. This requires the user to configure the application and the anonymity service to work together. For example, JonDonym enables the user's e-mail service to work with JonDonym. It also supports any application that has the ability to configure the proxy. Tor supports any application that has the ability to pass all its traffic thought a proxy. However, using any application other than the default browser on the Internet raises the chance to breach the user's anonymity.

The configuration for these applications is not that simple for non-technical users. When configuring any application to work with an anonymity service, it is important to fully understand how this application works to ensure not to leak the user information. For example, the DNS request which accompanies many applications might leak the user's data and this in return might breach the user's anonymity. Applications might not use the anonymity service to resolve the DNS name even if they are configured to do so [42].

The user can run any application on the I2P network that depends on TCP or UDP. The I2P messages are based on UDP. TCP applications count on using I2PTunnel which passes the TCP stream within the I2P network. For example, Eepsites [43] and IRC (Internet Relay Chat) use I2PTunnel [44] to work within the I2P's UDP based network. Eepsites are websites hosted anonymously on the I2P network. The user accesses these websites without getting any information about the one(s) created the website(s). At the same time, the website cannot detect the real identity of the users. These types of applications work by default only within the I2P network. To use browsing outside of the I2P network, an outproxy is needed to pass/forward the traffic. I2P network supports many applications such as Blogging, File storage, DNS, Email, File sharing, Web hosting and others. These applications differ from working within the I2P network or outside the I2P network. Some of these applications are supported by third parties. Therefore, the anonymity and the security level varies on these applications.

The configuration of the application and how the user sets the application on the anonymity network is an important issue. For example, the web browsing contains many details other than what anonymity system the user is using. The anonymity tools aim to make their browsers undistinguishable to raise the anonymity level. Tor browser is a modified version of Firefox based on Mozilla's Extended Support Release (ESR) Firefox branch [23]. It includes HTTPS-Everywhere [26], NoScript [27], modifying some of the default Firefox settings, and modifying some of the default extension settings. JonDoFox is the browser of JonDonym. It is a modified version of Firefox [28].

Even when using the default browser for the anonymity services, the right setting of the browser is important to ensure the safety of the user against many of the Internet websites which track their visitors. To this end, some of the tools used by web sites could also identify the user or his/her behavior for the purpose of advertisements, collecting data for different types of studies, or building a database about the visitors of the website. Thus, it is crucial to know the policy and the default setting for a browser with such tools. The question to consider here is: How such tools deal with the trade-off between browsing the websites with full offered services and saving the anonymity of the users.

Table I. presents the information how Tor Browser and JonDoFox, i.e. JonDonym browser, deal with these trade-offs. Compared to Tor and JonDonym, I2P network does not have a specific browser preference. After the connection establishment to the I2P network, the user manually configures the proxy setting in any browser to use the I2P network. The network encrypts the traffic between the users within the network regardless of the application used. I2P network is designed to work as a private network on the Internet. The browser could be used to configure the router of the user. For example, configuring the bandwidth up & down, participation on the floodfill, starting or stopping services such as IRC, WebHosting are all possible possible on the I2P network.

The application supported by the anonymity services are not the same. The method used to run applications other than the web browsing also varies from one anonymity service to another. How well the anonymity service is structured to support a number of applications is affecting the level of anonymity. For example, using the default anonymity browser or configuring the user's browser, could make a difference on the anonymity level. Therefore, it is not only the anonymity service that affects the anonymity level; it is also what application is used on that anonymity service.

D. Authority and logs

No doubt that the policy of the anonymity services about the cooperation with the authority (operator, or regulator) and keeping logs affects the level of privacy. For example, JonDonym's agreement with the operators requires not to keep any log and not to exchange information between operators of the mixes. The reason behind this policy is that the identities of the operators are known, they work according to the regulations in their own countries. Therefore, in JonDonym, there are several points that must be taken into consideration when evaluating the anonymity of such a system:

- The mixes that construct the path are fixed. That means knowing that the user employs one of these mixes, e.g. the last mix, implies knowing the first and the second mix.

- The number of mixes on the JonDonym network is very limited compared to the Tor network. On the JonDonym network, there are only nine cascades. Six of them are operated by companies and three of them are operated by individuals.

- The operators of these mixes are known and registered. They work according to the regulations in their countries regarding the cooperation with the authorities.

- These cascades are fixed. This eases being approached and investigated by the authorities.

On the other hand, on the Tor network:

- The nodes that construct the user's path are not fixed. The user connects to three nodes that change periodically. Therefore, knowing that the user connects to a specific exit node does not necessarily imply knowing the first or the middle node.

- The number of Tor's nodes is around 8000, which makes it relatively harder to get information about them.

- The operators of these nodes are not known. Tor does not require their users to identify themselves when offering to run a node. This might help to protect the operators' identities but it also does not guarantee that the operators are trusted.

- The nodes on the Tor network are supposed to be online as much as possible. However, there is no guarantee, because most of these nodes are run by volunteers.

- On the other hand, keeping the log for the created circuits is an available option for the nodes' operators. When the debug option is enabled in the configuration file, then the log file will contain the information about the circuits and cells. The operator of the node has the ability also to modify the source code of Tor to log additional information about the cell. It can be used to extract information and analyze them later [24]. Getting these extra information does not mean that these tools do not provide anonymity, it indicates that there is specific amount of information available to the operators of the nodes that the user should be aware of.

Furthermore, on the I2P network:

- The I2P user has the option to modify the number of routers used when exchanging messages. In addition, end-to-end encryption is used. The concept of garlic routing also used when exchanging messages. This way, messages that pass through the routers are not distinctive. That means the purpose or the content of the messages could not be extracted or inferred easily. For example, information such as whether the messages are to form an extension to the number of routers in the tunnel or if they contain data would not be extracted from the messages.
- The I2P network is decentralized, So there is not one point that is responsible for the network or represents the network.
- The user does not need to know all the routers in the network to be able to use the network resources.
- I2P network's design is different from Tor and JonDonym; it is basically designed to provide a private network within the Internet. The number of outproxy is very limited. This makes the browsing outside the network also low compared to Tor and JonDonym. Therefore, the possibility that the user will use the same exit point frequently is high. This does not mean that it is a threat, but increases the probability to correlate the user with its traffic based on factors such as access time, duration, and the amount of data used.

Based on the above, the harder the possibility to compromise all of the nodes on the user's path, the better the anonymity level. In addition, what information the service provider (operator) has about the user and the operator's willingness to provide this information when asked to do so is also important in measuring the level of anonymity.

TABLE I. DEFAULT BROWSER'S SETTINGS FOR ANNONYMITY SERVICES

	JonDoFox	**Tor Browser**
Cookies	Disabled	Enabled
Third-party Cookies	Disabled	Disabled
JavaScript	Disabled	Enabled
Flash-Cookies (LCO)	Disabled	Disabled
WebGL	Enabled	Disabled
Flash Plug-in	Disabled	Enabled
Java	Disabled	Enabled
Silverlight	Disabled	Enabled

E. *Threat Models*

In the ideal case, the anonymity services provide anonymity to their users and protect their privacy. However, there are possible threats that could break the anonymity of such services. The anonymity services are based on the separation between the user identity and the data sent or received by the user. One of the threats that face such services is to correlate the user data and the final destination data. This is possible by monitoring the first point in the anonymity network and the last point which connects the user with the final destination (web server). Through the analysis of amount of data, it is possible to correlate the user and his/her final destination when there is variety in the data size. The path on the JonDonym network is known, if the attacker has the ability to monitor the traffic from the first mix and the last mix (out of the last mix), then the correlation between the users of this cascade and the amount of sent and received data is possible. The path on the Tor network is not fixed, but the correlation is also a possible threat. To this end, there are studies on using marking techniques to trace the user activities. They have their limitations to the specific user, or the specific webserver, or even the specific exit node. The attacker could compromise an entry node and an exit node. Then the traffic out of the entry node is marked. The attacker then watches for the mark to appear at the exit node. Indeed, the probability of the user who is using the compromised entry node to also using the compromised exit node is very low, but it is still possible. The mark also might be used to track the webserver instead of the exit node [29]. In this case, the attacker compromises an entry node and watch for the users who are using this entry node to access this specific web server. On the other hand, the design of I2P network makes this kind of correlation a low threat. The path is not fixed or specified; users build inbound and outbound tunnels which do not count on the type of the router. All routers on the networks can be part of any path. The encryption mechanism provides the confidentiality and the integrity of the messages. However, if the attacker has the resources to monitor all routers, then he/she may have enough data to discover paths.

As for the JonDonym network, this type of attack can target mix server. A mix server has a limit on the number of users it can serve. The attacker could use this limit to break the anonymity of the mix server. If the attacker connects to a mix server to fill its capacity (n) to the point (n-1) when the user connects to the only space left in the mix server, the attacker could isolate and detect the user's traffic.

The threat models are not the same for all anonymity services, what is considered as a threat to one service could not be applied to another anonymity service. Even when they share the same threat to a certain saturation point, the level of the risk is not always the same. Therefore, to measure the anonymity of any anonymity service, the threat model should be taken into consideration based on the environment or the purpose that the anonymity service is used for.

Accordingly, evaluating the level of anonymity should be done in a comprehensive way that take into consideration the purpose, the design, and the environment etc. Thus, we aim to use these five factors: the level of information available to the service provider, the obfuscation options, application anonymity, the authority and the logs, and the threat models to measure the level of anonymity of any anonymity service.

V. EVALUATION

In this section, we will discuss how we can use the aforementioned factors to measure the anonymity of Tor, JonDonym, and I2P. These factors are dynamic so they change over time based on many variables such as the user behavior, the anonymity system used, the configuration of the system, the purpose of using the anonymity system, etc. Therefore, we first aim to quantify these factors to be able to measure and compare them with each other. We call this the "Weighted anonymity factor". The following presents our weighted anonymity factor measurement:

A. Factor Calculation

To quantify these factors, we grouped them into three categories as shown in Table II. These categories, namely High, Mid, and Low are then converted into numerical values as 100, 67, and 33, respectively. The exception is for the obfuscation, where we label it as "Yes" or "No" depending on whether an obfuscation is used or not, respectively. The reason is that some of the anonymity systems contain obfuscation techniques and others do not. Also the use of these techniques (if they are available) is optional. Therefore, the value is set to 100 (for "No") and 0 (for "Yes"). The higher these values for the factors the lower the anonymity level of the system. For example, a 100 in the Threat model factor is applied whenever the threat in the case under study is very strong (i.e. highly probable). The three categories is represented by (100, 67, and 33) as an approximation for the High, Mid, and Low. These values could be expanded and detailed to a scale from 10 to 100 to increase the accuracy. More on this on section E.

TABLE II. PROPOSED ANONYMITY FACTORS

Level Of Information	High	Mid	Low
	100	67	33
Obfuscation	Yes	No	
	0	100	
Authority And Log	High	Mid	Low
	100	67	33
Application Configuration	Low Security Configuration	Mid Security Configuration	High Security Configuration
	100	67	33
Threat Model	Low Cost	Mid Cost	High Cost
	100	67	33

B. Weight Calculation

Given that the weights of the factors may vary from one evaluation environment to another, quantifying these factors to measure the anonymity is necessary but not enough. Also, the weights of the factors have to be considered. Therefore, we use the "Pairwise Comparison" technique to evaluate the weight of these factors. Each one of the factors is compared with all other factors, then the weight for the factor is calculated based on these comparisons. The higher the weight of a factor is, the more important it becomes for the anonymity of a given service. Calculating the weights is performed until all factors are evaluated compared to each other as shown in table III.

γ_1 refers to the first factor - "level of information available to the service provider" - , γ_2 refers to the second factor and so on. Table IV shows the weights of the five factors after the comparison and their total value. Based on the weights value, the level of information, the application configuration, and the authority and log factors have the same weights (important). The obfuscation has the lowest importance compared to the other factors. The weights represent the importance of each factor compared to the other factor. Using the pairwise comparison helps in deciding how to rank the factors (weight them) compared to each others.

TABLE III. CALCULATING THE WEIGHTS

γ_1				
γ_2	$2\gamma_1$			
γ_3	γ_1 γ_3	γ_3		
γ_4	γ_4	γ_4	γ_3 γ_4	
γ_5	γ_1	γ_2 γ_5	γ_3 γ_5	γ_4 γ_5

TABLE IV. FINAL WEIGHTS OF THE FACTORS

γ_1	**γ_2**	**γ_3**	**γ_4**	**γ_5**	**Total**
4	1	4	4	3	16

C. *Weighted Anonymity Factor*

Eq.1 and Eq.3 are applied after calculating the values of the factors and calculating the weights. Eq.2 is the total of the weights of the factors. (*f*) represents the value of a factor.

$$\textit{Weighted Anonymity Factor (WF)}= \gamma_1\ f_1 + \gamma_2\ f_2 + \gamma_3\ f_3 + \gamma_4\ f_4 + \gamma_5\ f_5 \qquad (1)$$

$$= \sum_{i=1}^{n} \gamma_i\ f_i \quad \text{where n = number of factors} \qquad (2)$$

$$\textit{Total Weight}\ (\mathrm{T}_\gamma) \qquad = \sum_{i=1}^{n} \gamma_i \qquad (3)$$

The measurements may vary from one environment to another where different factors are applied or when the numerical conversion is different than what we used here, Table II. To generalize measurements, Eq.4 shows converting the calculated values based on the factors used to a percentage by using the minimum and maximum values from Eq.1.

$$\textit{Weighted Anonymity Factor in Percentage}\ (\%) = \left(1 - \frac{WF-Min(wf)}{Max(WF)-Min(WF)}\right) * 100 \qquad (4)$$

For the factors used in this paper, we can rewrite Eq.4 after calculating the weights to the form in Eq.5.

$$\textit{Weighted Anonymity Factor in Percentage}\ (\%) = \Big(1 - \frac{WF-495)}{1600-495}\Big) * 100 \qquad (5)$$

D. *Evaluation Case Study*

In this scenario, we assume three users among whom we will compare the level of anonymity. It is important to note that we do not aim to name what is the best anonymity service; we aim to evaluate the level of anonymity according to the environment that accompanies using these anonymity services.

The first user (A) uses standalone Tor to browse Internet websites. The user configures Chrome browser to work with Tor by setting the browser to access Tor via Socket. To increase the anonymity level, the user adds Scramblesuit as an obfuscation option to his “torrc” file to access Tor via a bridge. The user (A) browses websites on the Internet which include a compromised webserver by an attacker. The webserver injects the coming request to force the browser to request images from another website that belongs to the attacker. The attacker aims to get the identity of the user by forcing the browser to send requests without using the Tor network.

User (B) chooses to use JonDonym as an anonymity service. The user does not have a technical background. All the settings are left to default. The only addition to the default setting is that the user chooses to use the TCP/IP forwarder. The user (B) wants all the activities that she performs on the Internet to be anonymous. Therefore, the user (B) uses JonDoFox to browse all the Internet websites. She usually visits web sites such as news, videos, email, Internet shopping, and accesses her bank account.

User (C) lives in a country where the Internet is censored and some of the websites are blocked. Therefore, user (C) uses Tor to gain access to the blocked Internet blogs. The user (C) browses these blogs and participates on them via Tor. The user is concerned about his identity so he uses the Internet from the company that he works at. It seems that user (C) is the only person who is using Tor on this company. The user manages to organize his time so that he only access Tor at the end of the day between 5-6 pm daily during the weekdays.

According to the scenario above, Table V shows how this scenario is converted to measurable numeric values, using the proposed factors.

TABLE V. EVALUATED FACTORS FOR USERS (A), (B), AND (C)

	Level of Information	**Obfuscation**	**Authority and Log**	**Application Configuration**	**Threat Model**
A	33	0	33	100	67
B	100	0	67	33	100
C	67	100	100	33	67

Table V is calculated based on the given information on the scenario above and how the Users (A), (B), and (C) are using these anonymity services. For example, the user (C) did not include an obfuscation option when using the anonymity service; therefore, the obfuscation value is measured as 100. The user (A) prefers to use his favorite browsers instead of using the default Tor browser. Therefore, the possibility to have a DNS leak is higher specially when accessing suspicious websites or when using any other application other than browsing. Based on that, user (A) gets 100 on the application configuration. Even though, the user (B) uses some sort of obfuscation, she misses the fact that browsing any website that already linked to her real identity such as the email or the bank account even through an anonymity service does not mean that she is anonymous. Furthermore, the information available to the exit node in this case is high even if the information does not contain passwords. The level of information is evaluated as 100 in this case. The same applies to the user (C), he uses Tor on the same time daily from the same place where no one else is using Tor.

Using Table V and the Eq.1, the weighted factors will be calculated as follows:

$$WF = \gamma_1 \, \mathrm{f}_1 + \gamma_2 \, \mathrm{f}_2 + \gamma_3 \, \mathrm{f}_3 + \gamma_4 \, \mathrm{f}_4 + \gamma_5 \, \mathrm{f}_5$$

$$WF = 4\,\mathrm{f}_1 + \mathrm{f}_2 + 4\,\mathrm{f}_3 + 4\,\mathrm{f}_4 + 3\,\mathrm{f}_5$$

$$WF_A = 4 \ast 33 + 0 + 4 \ast 33 + 4 \ast 100 + 3 \ast 67$$

$$= 865$$

$$WF_A\,(\%) = \left(1 - \frac{865-495)}{1600-495}\right) \ast 100$$

$$= 66.5\ \%$$

$$WF_B = 4 \ast 100 + 0 + 4 \ast 67 + 4 \ast 33 + 3 \ast 100$$

$$=1100$$

$$WF_B\,(\%) = \left(1 - \frac{1100-495)}{1600-495}\right) \ast 100$$

$$= 45.2\ \%$$

$$WF_C = 4 \ast 67 + 100 + 4 \ast 100 + 4 \ast 33 + 3 \ast 67$$

$$=1101$$

$$WF_C\,(\%) = \left(1 - \frac{1101-495)}{1600-495}\right) \ast 100$$

$$= 45.16\ \%$$

Based on the above calculations, user (A) has higher level of anonymity compared to both user (B) and (C).

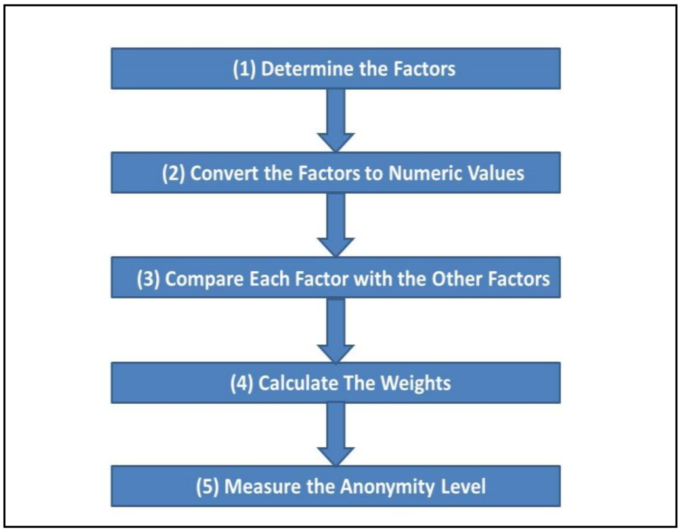
(1) Determine the Factors
(2) Convert the Factors to Numeric Values
(3) Compare Each Factor with the Other Factors
(4) Calculate The Weights
(5) Measure the Anonymity Level

Fig. 4. Sequence for Anonymity Measurement

E. *Expanding the Quantization*

The mechanism we used in this paper to measure the anonymity level is first based on determining the factors. Then, each factor is divided into three levels (two for the obfuscation) to be able to numerically evaluate each factor. The importance of each factor is then determined based on the comparison between all the factors. Fig. 4 shows the sequence for measuring the anonymity. The second step which is converting the factors into measurable values has three levels High, Mid, and Low. According to these values, the factors are converted into a numeric values. This could be considered as the applicable form of measuring the anonymity. However, it is possible to expand this step to improve the accuracy of quantization of the factors by: (1) Instead of using three levels; the factors could be evaluated as a scale, for example, from 10 to 100, (2) At the same time, each value on the scale should represent the level of the anonymity on the factor in a predefined way. This way the value of the factors is determined more accurately. For example, if we like to apply the extended scale to the "Threat Models" factor then the values will be from 10 to 100 instead of (33, 67, and 100). The threats or the attacks on the anonymity systems should be ordered to match the scale from 10 to 100. This requires the study and evaluation of all the possible threats on the anonymity systems and how applicable they are. This way, the scale has predefined values for every possible threat against the anonymity systems in the Threat models factor. The same step could be repeated for the other factors. We believe this type of an approach could increase the accuracy of the proposed factors in measuring the anonymity provided by the anonymity services.

VI. Conclusion

In this paper, we propose and evaluate five important factors that affect the level of privacy in anonymity services. Understanding these factors and knowing how to deal with them is an important step towards improving users' privacy. To this end, three popular anonymity systems, namely Tor, JonDonym, and I2P, were used as case studies to analyze these factors. Our analysis showed that even though these systems aim to provide anonymity to their users, there is still information available in these systems to the operators of the services about the users. Furthermore, the infrastructure and the browser settings vary from one system to another. The setting is configured based on the developers/administrators evaluation of the possible threats. The same threat might be considered high in one system but low in another. We applied a measurable mechanism to evaluate the anonymity of a given situation based on the factors we proposed. The evaluation could be used on any anonymity system using different scenarios. Future research will continue to analyze other anonymity systems based on the proposed five factors, will evaluate them using the expanded quantization approach and under adversarial conditions.

References

[1] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham. "Fingerprinting Information in JavaScript Implementations," *in Proceedings of Web 2.0 Security and Privacy 2011 (W2SP)*, San Francisco, May 2011.

[2] K. Mowery and H. Shacham. "Pixel Perfect: Fingerprinting Canvas in HTML5," in Proceedings of Web 2.0 Security and Privacy (W2SP) ,2012.

[3] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. "Website fingerprinting in onion routing based anonymization networks," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, Chicago, USA, 2011, pp.103-104.

[4] P. Eckersley. "How unique is your web browser?," *In Privacy Enhancing Technologies*, pages 1–18. Springer, 2010.

[5] T. Ries, A. Panchenko, R.State , and T. Engel. "Comparison of low-latency anonymous communication systems: practical usage and performance," *in Proceedings of the Ninth Australasian Information Security Conference*, January 17-20, 2011, Perth, Australia

[6] D. Abou-Tair, L. Pimenidis, J. Schomburg, and B. Westermann. "Usability Inspection of Anonymity Networks," *in Proceedings of the 2009 World Congress on Privacy, Security, Trust and the Management of e-Business,* August 25-27, Saint John, New Brunswick, Canada, 2009.

[7] J. Clark , P. C. Oorschot , and C. Adams. "Usability of anonymous web browsing: an examination of Tor interfaces and deployability," *in Proceedings of the 3rd symposium on Usable privacy and security*, July 18-20, 2007, Pittsburgh, Pennsylvania

[8] R. Wendolsky, D. Herrmann, and H. Federrath. "Performance comparison of low-latency anonymisation services from a user perspective," in Proceedings of the 7th international conference on Privacy enhancing technologies, p.233-253, June 20-22, 2007, Ottawa, Canada.

[9] 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.

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

[11] Tor Bridges. [Online]. Available: https://www.torproject.org/docs/bridges.html.en

[12] Tor Pluggable Transports. [Online]. Available: https://www.torproject.org/docs/pluggable-transports.html.en

[13] A. Lewman, "Tor partially blocked in China". [Online]. Available: https://blog.torproject.org/blog/tor-partially-blocked-china

[14] Obfs3. [Online]. Available: https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs3/obfs3-protocol-spec.txt

[15] T. Wilde. Great firewall Tor probing circa. [Online]. Available: https://gist.github.com/twilde/da3c7a9af01d74cd7de7

[16] P. Winter, and S. Lindskog, "How the great firewall of China is blocking Tor," in Proceedings of the 2nd USENIX Workshop on Free and Open Communications on the Internet , USENIX Association,2012.

[17] P. Winter, T. Pulls, and J. Fuss. "ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship," In Workshop on Privacy in the Electronic Society, Berlin, Germany, 2013. ACM.

[18] Z. Ling, J. Luo, W. Yu, M. Yang, and X. Fu, "Extensive analysis and large-scale empirical evaluation of tor bridge discovery," in INFOCOM, 2012 Proceedings IEEE, 2012.

[19] D. Fifield, N. Hardison, J. Ellithrope, E. Stark, R. Dingledine, D. Boneh, and P. Porras, "Evading Censorship with Browser-Based Proxies," In PETS, 2012.

[20] K. Dyer, S. Coull, T. Ristenpart and T. Shrimpton, "Protocol Misidentication Made Easy with Format-Transforming Encryption," ACM SIGSAC Conference on Computer and Commu- nication Security, CCS'13, pp. 61-72, ACM, 2013.

[21] Meek. [Online]. Available: https://trac.torproject.org/projects/tor/wiki/doc/meek

[22] JAP, Data Collection Techniques. [Online]. http://anon.inf.tu-dresden.de/help/jap_help/en/help/wwwprivacy_technik.html

[23] The Design and Implementation of the Tor Browser [Online]. Available: https://www.torproject.org/projects/torbrowser/design/#philosophy

[24] 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.

[25] R. Dingledine, and N. Mathewson. *Tor Path Specification* [Online]. Available: https://gitweb.torproject.org/torspec.git/tree/path-spec.txt

[26] *HTTPS- Everywhere Extension* [Online]. Available: https://www.eff.org/https-everywhere

[27] *NoScript Firefox Extension* [Online]. Availabe: https://noscript.net/

[28] JonDoFox [Online]. Available: https://anonymous-proxy-servers.net/en/jondofox.html

[29] Z. Ling, J. Luo,W. Yu, and X. Fu, "Equal-sized cells mean equal-sized packets in Tor?," *in Proceedings IEEE ICC*,2011, pp.1 -6

[30] D. McCoy, K. Bauer, D. Grunwald, T. Kohno and D. Sicker. *Shining Light in Dark Places: Understanding the Tor Network*. In Proc. of Privacy Enhancing Technologies Symposium (PETS), Leuven, Belgium, 2008.

[31] JonDonym InfoService. [Online]. Available: https://anonymous-proxy-servers.net/en/help/infoservice.html

[32] K. Shahbar, and A.N. Zincir-Heywood, "Traffic flow analysis of tor pluggable transports," *in Network and Service Management (CNSM), 2015 11th International Conference on* , vol., no., pp.178-181, 9-13 Nov. 2015.

[33] P. Maymounkov, and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," in Proceedings of the 1^{st} international workshop on Peer-to-Peer Systems (IPTPS), 2002.

[34] Garlic Routing . [Online]. Available: https://geti2p.net/en/docs/how/garlic-routing

[35] C. Diaz, S. Seys, J. Claessens, and B. Preneel, “Towards measuring annonymity” in Proceedings of privacy enhancing technologies workshop (PET 2002), 2002, USA.

[36] S. Murdoch, “Quantifying and measuring anonymity,” in Proceedings of the 8th International Workshop on Data Privacy Management and Autonomous Spontaneous Security - Volume 8247, Pages 3-13, 2013.

[37] O. Berthold, A. Pfitzmann, R. Standtke, “The disadvantages of free MIX routes and how to overcome them,” in Federrath, H. (ed.) Anonymity 2000. LNCS, vol. 2009, pp. 30–45. Springer, Heidelberg (2001)

[38] D. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” Communications of the ACM, vol. 4, no. 2, February 1981.

[39] D. Chaum, “The dining cryptographers problem: Unconditional sender and recipient untraceability,” Journal of Cryptology, vol. 1, pp. 65–75, 1988.

[40] A. Serjantov and G. Danezis, “Towards an information theoretic metric for anonymity,” in Proceedings of Privacy Enhancing Technologies (PET2002), ser. Springer-Verlag, LNCS, P. Syverson and R. Dingledine, Eds., vol. 2482, San Francisco, CA, April 2002.

[41] G. T´oth, Z. Horn´ak, and F. Vajda, “Measuring anonymity revisited,” in Proceedings of the Ninth Nordic Workshop on Secure IT Systems, S. Liimatainen and T. Virtanen, Eds., Espoo, Finland, November 2004, pp. 85–90

[42] *TorifyHOWTO* [Online]. Available: https://trac.torproject.org/projects/tor/wiki/doc/TorifyHOWTO

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

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

[45] *I2P’s Threat Model: Harvesting Attacks* [Online]. Available: https://geti2p.net/en/docs/how/threat-model#harvesting