Shedding Light on the Dark Corners of the Internet: A Survey of Tor Research

Article · March 2018

DOI: 10.1016/j.jnca.2018.04.002

CITATIONS
17

READS
972

3 authors:

Saad Saleh
University of Groningen
18 PUBLICATIONS **75** CITATIONS
SEE PROFILE

Junaid Qadir
Information Technology University of the Punjab
250 PUBLICATIONS **3,073** CITATIONS
SEE PROFILE

Muhammad Usman Ilyas
University of Jeddah
62 PUBLICATIONS **640** CITATIONS
SEE PROFILE

Shedding Light on the Dark Corners of the Internet: A Survey of Tor Research

Saad Saleh[a,*], Junaid Qadir[b], Muhammad U. Ilyas[a,c]

[a] *School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad-44000, Pakistan*
[b] *Information Technology University, Lahore, Pakistan*
[c] *Department of Computer Science, Faculty of Computing and Information Technology, University of Jeddah, Jeddah, Mecca Province 21589, Saudi Arabia*

Abstract

Anonymity services have seen high growth rates with increased usage in the past few years. Among various services, Tor is one of the most popular peer-to-peer anonymizing service. In this survey paper, we summarize, analyze, classify and quantify 26 years of research on the Tor network. Our research shows that 'security' and 'anonymity' are the most frequent keywords associated with Tor research studies. Quantitative analysis shows that the majority of research studies on Tor focus on 'deanonymization' the design of a breaching strategy. The second most frequent topic is analysis of path selection algorithms to select more resilient paths. Analysis shows that the majority of experimental studies derived their results by deploying private testbeds while others performed simulations by developing custom simulators. No consistent parameters have been used for Tor performance analysis. The majority of authors performed throughput and latency analysis.

Keywords: Tor; Security; Anonymity; Survey; Analysis; Deanonymization; Breaching; Path selection; Performance analysis.

1. Introduction

The Internet has revolutionized the world by transforming it into a global entity. Widespread advantages of Internet have spawned new industries and services. However, this connectivity comes at the cost of privacy. Every Internet client has a unique identity in the form of an Internet protocol (IP) address which can be translated to its location by the local Internet service provider (ISP). This lack of privacy has serious implications, particularly for journalists, freedom fighters and ordinary citizens.

Lack of privacy has lead to the use of anonymous communication networks (ACN). ACNs hide client IP addresses through various techniques. There are a number of ACNs including Tor, Java anonymous proxy (JAP), Hotspot Shield and UltraSurf etc. Among various ACNs, Tor is the one of the most popular network, owing to its distributed nature which makes it difficult to connect the two end points of a session. Recently, Tor has been used for bomb hoax at Harvard [1]. Similarly, it has been used by the Russians to bypass online censorship [2]. A number of attempts are being made by FBI and other organizations to breach Tor network [3][4][5].

In this paper, we survey various studies conducted on the Tor network covering the scope of these studies. We quantify the studies into three broad but distinct groups, including (1) deanonymization, (2) path selection, (3) analysis and performance improvements, and several sub-categories. To the authors' best knowledge, this is the most comprehensive attempt at analyzing Tor network research with a focus over its anonymity mechanism. Table 1 presents a comparison of this survey with previous surveys covering the scope of researches and implementation (experiments), verification (simulations) and analysis of various research works. Categorization of first column is made by listing all Tor areas considered in our study. AlSabah and Goldberg [6] presented the most comprehensive study covering complete Tor network and our paper is complementary to their survey paper. However, our paper pays more focus to the anonymity and breaching aspects of Tor than their paper. Their research paper presents only twenty references related to anonymity while we present more than 120[1] references.

Analysis of keywords used in various studies shows that anonymity, security and privacy have been used the most. Our study shows that majority of the research works have been made in the field of "deanonymization"

*Corresponding author
Email addresses: saad.saleh@seecs.edu.pk (Saad Saleh), junaid.qadir@itu.edu.pk (Junaid Qadir), usman.ilyas@seecs.edu.pk, milyas@uj.edu.sa, usman@ieee.org (Muhammad U. Ilyas)

[1] This paper has 146 references, some of the references are to tools rather than research works; in all, we are considering a research corpus of 120 references.

Table 1: Comparison of other surveys with this survey. Tick mark indicates coverage while blank space indicates non-coverage of topics in various research works. Coverage (Studies) indicate number of research works referred in any survey.

Topic↓	Sub-Topic↓	Research, Year, Coverage (Studies)							
		<This Paper>, 2017, 120	AlSabah and Goldberg [6], 2016, 120	Koch *et al.* [7], 2016, 10	AlSabah and Goldberg [8], 2015, 99	Conrad and Shirazi [9], 2014, 40	Jagerman *et al.* [10], 2014, 37	Ren and Wu [11], 2010, 109	Johnson and Kapadia [12], 2007, 32
Deanonymization	Hidden services	✓	✓		✓			✓	✓
	Finger printing	✓	✓		✓				
	Attacks	✓	✓		✓	✓	✓	✓	✓
	Traffic analysis	✓	✓	✓	✓	✓	✓		✓
	Improvements	✓	✓		✓				
	Bypassing Tor	✓	✓			✓	✓	✓	✓
Path Selection	Algorithm design	✓	✓		✓				
	Path analysis	✓	✓		✓				
Analysis	General	✓		✓	✓	✓		✓	✓
	Modelling	✓							
	Evaluation	✓		✓					
	Improvement	✓	✓						
	Mobile Tor	✓							
	Parameters	✓							✓
Experiments	Private Setup	✓							
	PlanetLab	✓							
	Cloud services	✓							
	OpenFlow	✓							
	UC framework	✓							
Simulations	Custom Simulator	✓							
	ExperimenTor	✓							
	Shadow simulator	✓							
	ModelNet	✓							

track, followed by "performance analysis and architectural improvements". In the deanonymization track, a major chunk of research is devoted to *breaching attacks* followed by *traffic analysis*. In the path selection track, most research works focused on the development of new algorithms. Relays, protocol messages and traffic interception have been the most frequently exploited factors in the Tor's deanonymization track. In the path selection track, performance and anonymity have been the most commonly used factors. Performance, relay selection and anonymity have been the most studied parameters in the performance analysis and improvement track. Analysis over simulations and experiments shows that 60% of studies used experiments and 86% of those experiments were carried out on private testbed networks. Among simulations, 75% of the studies developed their own simulator to analyze Tor network. Analysis of simulation parameters shows that there is no distinct pattern of parameters. However, majority of the studies used bandwidth and latency.

Table 2 presents a glossary of the important abbreviations used in our survey paper. This paper is organized as follows: Section 2 introduces the architecture of Tor network and its comparison with other anonymity services. Section 3 presents the studies covering deanonymization, path selection, and performance analysis and architectural improvements. Section 4 presents the simulations and experiments conducted in previous studies. Section 5 presents the Tor performance metrics, our findings and open research areas in Tor. Finally, section 6 concludes the paper.

2. Architecture and comparison of Tor with other anonymity services

In this section, we present and discuss Tor and other anonymity tools. In the first part, we present the architecture of Tor network before presenting details of the research in Tor. In the second part, we present the comparison and working mechanism of other anonymity tools which compete with Tor.

2.1. Architecture of Tor network

Tor network is composed of a decentralized distributed network of relays operated by volunteer users [13]. In July 2016, nearly 10,000 users (per day) participated in the Tor network (as Tor relays and Tor bridges) to provide anonymity services to nearly half a million users daily [14]. History of Tor dates back to late 1990's when Goldschlag, Reed and Syverson presented the architecture and implemented onion routing in several papers [15, 16, 17, 18] which laid the foundation of Tor network by providing

Table 2: Glossary of the important abbreviations used in the text.

Glossary			
ACK	Acknowledgement	MRA	Multi-Resolution Analysis
ACN	Anonymous Communication Network	NAT	Network address translation
ADSL	Asymmetric digital subscriber line	NTP	Network Time Protocol
AES	Advanced Encryption Standard	OP	Onion Proxy
AS	Autonomous System	OR	Onion Router
CGI	Computer-generated imagery	PGP	Pretty Good Privacy
CSRF	Cross site request forgery	POP3	Post Office Protocol 3
DHCP	Dynamic Host Configuration Protocol	PPTP	Point-to-Point Tunneling Protocol
DNS	Domain Name System	P2P	Peer-to-Peer
DoS	Denial of Service	QoE	Quality of Experience
DS	Directory Server	QoS	Quality of Service
DSL	Digital Subscriber Line	ROC	Region of Convergence
EWMA	Exponentially weighted moving average	RRD	Round Robin Database
FIFO	First In First Out	RTT	Round Trip Time
FN	False Negative	SMTP	Simple Mail Transfer Protocol
FP	False Positive	SSH	Secure Shell
HTML	HyperText Markup Language	SVM	Support Vector Machine
HTTP	Hypertext Transfer Protocol	TAP	Tor Authentication Protocol
IMAP	Internet Message Access Protocol	TCP	Transmission Control Protocol
ICMP	Internet Control Message Protocol	TMT	Tunable mechanism of Tor
IP	Internet Protocol	Tor	The Onion Router
ISP	Internet Service Provider	TP	True Positive
I2P	Invisible Internet Project	TTL	Time To Live
JAP	Java Anonymous Proxy	URL	Uniform Resource Locator
JVM	Java virtual machine	VDE	Virtual Distributed Ethernet
LAN	Local area network	VM	Virtual Machine
L2TP	Layer 2 Tunneling Protocol	VPN	Virtual Private Network
ML	Machine Learning	VPS	Virtual Private Server

proxy servers which are resilient to eavesdropping and effectively hide client's IP address.

The Tor network consists of routers which cooperate with each other to provide low latency anonymity services to users. Central servers help Tor establish and update links between Tor routers. User participation as Tor relays (router) is optional, but it is recommended because it improves the chances of staying anonymous, because it increases the traffic to the user.

Tor's architecture has three types of components, namely onion proxy (OP), onion router (OR) and directory server (DS). OPs are used by Tor users to obtain up-to-date information of operating relays from DS. OPs also creates connections using the information from a DS. Users may configure OPs to select specific routers.

ORs are Tor relay routers, operated by volunteer users, to act as entry (guard), middle and exit relays. Information of all online relays is available at DS. To counter attacks on Tor that block Tor relays, a secret group of Tor relays exists with the DS, called *bridges*. A set of three bridge relays is available through unique *Gmail* addresses. Once a connection is established, every OR knows only immediate predecessor and successor node.

Nine authorities acting as Tor DSs keep an up-to-date record of all available ORs and broadcast the bandwidth, IP, public key, exit policies etc. to OPs.

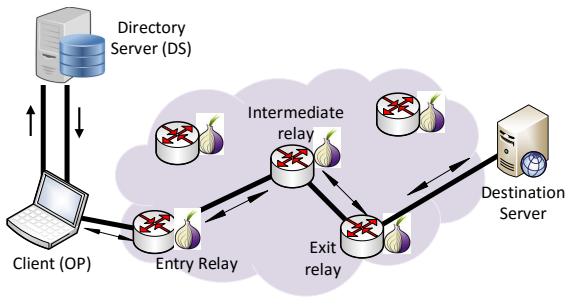

Figure 1: Architecture of Tor network.

Figure 1 shows a circuit established from a user to a server through three Tor relays. The various steps involved in circuit establishment are listed below:

1. OP sends HTTP requests to DS for information about Tor relays.
2. OP selects three Tor routers (entry relay, intermediate relay and exit relay) using Tor's path selection algorithm considering maximum anonymity and performance.
3. OP sends a *Create Cell* request (containing half of Diffie-Hellman handshake [19]) to the entry node. The entry node replies with the hash of the negotiated key.

4. Next, OP sends a *Extend Cell* request to the entry node containing the address of the intermediate relay and encryption key. Entry node forwards the cell to the intermediate node. Similar to previous case, the intermediate node replies with *Created cell* response. Similar process continues till client negotiates the key with the exit relay.
5. OP constructs IP packet $P1$ with source and destination IP addresses of exit relay and destination server, respectively, and packet size of 512 bytes.
6. OP encrypts the packet further with the key $E3$, negotiated between client and exit relay, and containing source and destination addresses of intermediate relay and exit relay.
7. Next, OP encrypts with the key $E2$, negotiated between client and intermediate relay, and containing source and destination IPs of entry relay and intermediate relay.
8. Finally, OP constructs packet $P2$ encrypted with key $E1$, negotiated between client and entry relay, and containing source and destination IPs of OP and entry relay.
9. Packet $P2$ is transmitted from the entry relay, which decrypts the packet and forwards it to intermediate relay. All relays decrypt the packet using their specialized decryption keys and forward it towards the destination.

Cell refers to the *Tor Packet*, comprising of payload data and headers, with an aggregate size of 512 bytes [20, 21, 22]. Padding is used to fill cells with less data.

Tor relays communicate with each other by pairwise TCP connections. Traffic multiplexing is used to transfer data between any pair of relays. Tor employs token buckets to rate limit connections. Buckets are filled and removed with tokens based on configured bandwidth limits and data read, respectively. TCP buffers are read using a round-robin scheduling mechanism. For flow control, edges (client and exit node) keep track of data flow by maintaining an active window about the packets in flight. Data packets are processed in a first-in-first-out (FIFO) manner from the queues of Tor relays. Multiplexing of packets, from Tor relays to relay links, is performed using exponentially-weighted moving average (EWMA) scheduler.

2.2. Comparison with other anonymization services

In this section, we present the features and working mechanisms of other deanonymization services which compete with Tor network. Table 3 presents a comparison of Tor with other deanonymization services. Table 3 shows that Tor is the only anonymity service which provides various services (http, https, visible TCP port, remote DNS, hides IP and user-to-proxy encryption) under all circumstances. On the contrary, JonDo, I2P, CGI and socks5 provide some services in limited circumstances only. A summary of various anonymity services is presented in following subsections.

2.2.1. Cross Platform Anonymity Tools

A number of cross platforms anonymity tools are used now-a-days. In below lines, we summarize the basic working mechanism of prominent anonymity tools.

- Java Anonymous Proxy (JAP or JonDonym) [23]: Users can select among several Mix Cascades, different from P2P.
- PacketiX.NET [24]: Virtual LAN card and Virtual HUB by Ethernet and can provide layer 2 VPN virtualization.
- JanusVM [25]: Uses Tor for all TCP based connections including DNS requests and provides web browser security.
- proXPN [26]: A VPN provider which hides the IP address of user and encrypts the data.
- USAIP [27]: A VPN service provider with servers in Switzerland, Luxembourg and Hungary etc.
- VPNReactor [28]: Uses a VPN connection with time limits for free and pro service and user logs are kept for 5 days.

2.2.2. Windows Based Anonymity Tools

A number of windows based anonymity tools are large competitors of Tor network. Basic mechanisms of prominent anonymity tools are summarized in below lines:

- xB Browser [29]: A browser designed to run over the Tor network and XeroBank anonymity network.
- Hotspot Shield [30]: Uses VPN. Hosts web servers accessible through proxy and has a central server that can be compromised.
- AdvTor [31]: Acts as a portable client and server for the Tor network. Improvements include the UNICODE path, HTTP and HTTPS protocols, estimates AS paths etc.
- SecurityKISS [32]: A VPN service based on OpenVPN, PPTP and L2TP.
- UltraSurf [33]: Uses HTTP proxy to bypass censorship and uses encryption protocols for privacy.
- CyberGhost VPN [34]: OpenVPN based proprietary client, Centralized server with VPN using 1024-SSL encryption.
- Freegate [35]: Uses range of proxy servers (called Dynaweb) along with encryption.

Figure 2: Word cloud of *keywords* used in Tor research works.

Table 3: Comparison of Tor with other anonymization services ('✓⋆' refers to 'In limited circumstances').

Proxy/ Anon. Service	HTTP	HTTPS	Visible TCP Port	UDP	Remote DNS	Hides IP	user-to-proxy encryption
https	✓	✓			✓	✓⋆	
socks4	✓	✓				✓	
socks4a	✓	✓			✓	✓	
socks5	✓	✓		✓	✓	✓	
CGI	✓⋆	✓⋆			✓	✓⋆	✓⋆
I2P	✓⋆	✓⋆		✓	✓	✓	✓
JonDo	✓	✓		✓⋆	✓	✓	✓
Tor	✓	✓	✓		✓	✓	✓

2.2.3. Linux based solutions

Linux, being the prominent platform, is used by various anonymity tools to guerentee anonymity to its users. Working mechanism of some tools is summarized in below lines:

- Tails (Amnesic Incognito Live System) [36]: Has a Debian Linux distribution using Tor network.
- Privatix [37]: Provides encryption with anonymous web browsing (using Tor, Torbutton and firefox).

2.2.4. Anonymous Search Engines

Several search engines also provided anonymity to their users by collecting no user information. Some of these anonymous search engines are as follows:

- Ixquick [38]: Opens all search results through a proxy for anonymity.
- DuckDuckGo [39]: Provides searcher's privacy and avoids "filter bubble". Displays same information to all users for a given search.

2.2.5. Anonymous Emails

Anonymous emails are provided by many servers. In following lines, we have summarized the tools provided anonymous email capability to their users.

- Anonymous E-mail [40]: Sends the emails with anonymous senders.
- Safe-mail [41]: Provides a secure communication, storage, distribution and sharing system for the internet.
- HushMail [42]: Provides a PGP encrypted email service.
- 10 Minute Mail [43]: Gives an email address which expires after 10 minutes to counter spam emails.
- Yopmail [44]: Provides a disposable email address with no registration and password.

3. Tor Research Areas

A large number of networks have utilized Tor networks for various purposes. To get an overview of the research

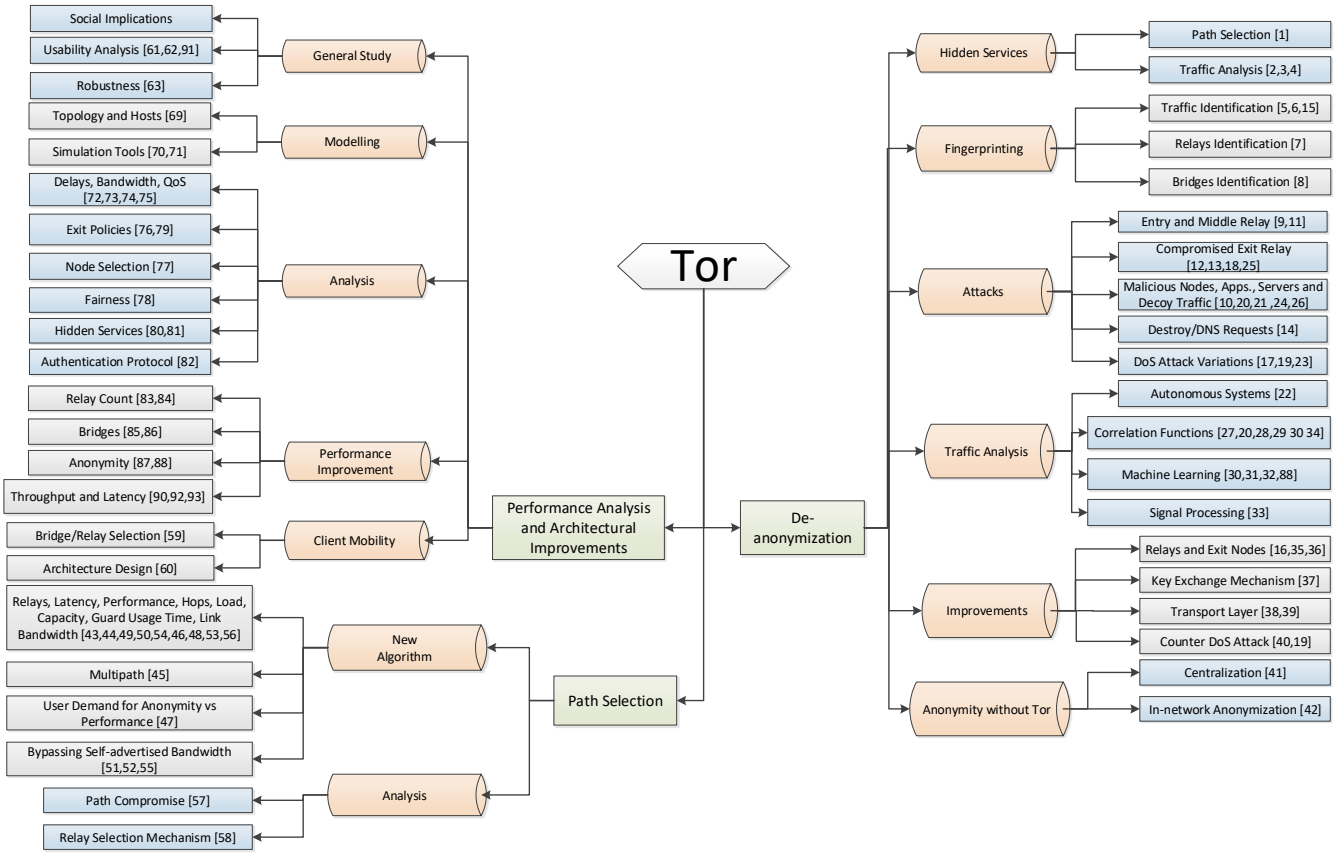

Figure 3: Taxonomy of Tor research. Tor literature can be broadly classified into three areas: deanonymization, path selection, and performance analysis and architectural improvements.

areas dealt in various studies, we created a word cloud of the keywords of these studies. Figure 2 shows the word cloud of keywords (on log-scale) of all studies cited in this paper. Data of keywords shows that anonymity, privacy and security are the most important terms dealt in various studies.

In our review, we observed that research works on Tor could be broadly classified into three tracks/categories which include (1) deanonymization, (2) path selection, and (3) Analysis and performance improvement. Figure 3 shows the classification of our survey paper along with a list of all research works present in various subcategories.

In deanonymization track, research works were observed in six different categories covering (1) Hidden services which limit their scope to hidden servers identification, (2) Fingerprinting which are based on pinpointing Tor network, (3) Attacks which are focused over breaching Tor network, (4) Traffic analysis which analyze Tor traffic to pinpoint the weaknesses, (5) Studies studying improvements in Tor to avoid deanonymization, and (6) Anonymity without Tor which suggest alternate methods to provide anonymity by pinpointing weaknesses in Tor.

In the path-selection track, all research works are either based upon (1) Development of new algorithms providing better efficiency and anonymity, and (2) Analysis of Tor's algorithm to study its strong and weak points in circuit establishment mechanism.

Lastly, analysis and performance improvement track focuses on four sub-areas which include (1) Generalized studies over Tor providing usability and social implications, (2) Modelling studies which focus on the development of model for analysis of Tor, (3) Analysis studies which cover QoS, relays, servers, etc., (4) Performance improvement studies provide modification in relays and architecture to provide better QoS, and (5) Development of efficient mechanisms for Tor clients with mobility.

Figure 4 shows the classification of various research areas studied in the Tor network. It is pertinent to mention that all numeric values used for all pie charts, figures and tables in this paper have been calculated by the authors. Source of all numeric values is the 'Reference' section at the end of the paper, which includes scholarly research articles. Moreover, references have been collected by the authors from Tor repository[2] with a particular bias towards papers covering 'Tor' network only. Span of collection varies from 2007 to 2017 in reputed international conferences and journals. We also included the important studies in this field before 2007 which play helpful role in

[2]https://www.freehaven.net/anonbib/

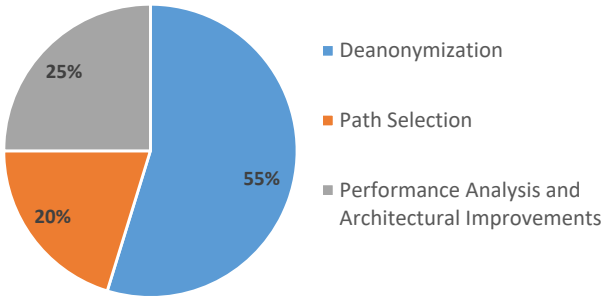

Figure 4: Classification of Tor research areas.

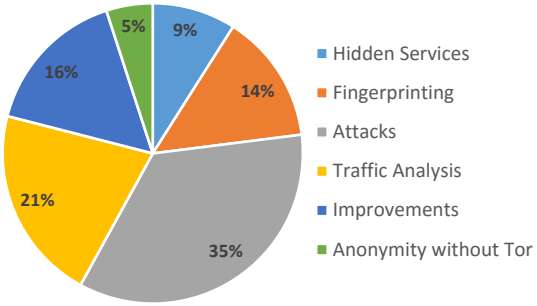

Figure 5: Classification of Tor's deanonymization approaches.

understanding of Tor network, such as [15][17]. Many articles were also collected from 'ACM digital library' and 'IEEE Xplore digital library' with a particular focus towards anonymity and security in Tor.

3.1. Tor Deanonymization

Breaching the Tor network is one of the most widely studied research problems. In fact, the majority of the studies describe deanonymization attacks without identifying any counter-measures [45]. Research works covering deanonymization can be subdivided into a number of subcategories including (1) Hidden services identification, (2) Tor traffic identification, (3) Attacking Tor network, (3) Traffic fingerprinting, (4) Focusing over Tor improvements, and (6) Providing anonymity without Tor. Classification of various research problems is shown in the pie chart in Figure 5.

Table 4 presents a comparison of various research works in the Tor's deanonymization track. Prominent patterns show that relay compromise and traffic interception are the most frequent factors in deanonymizing Tor. This suggests that relays and traffic are more susceptible for exploitation than other factors. Individual details of various research works in following subsections would explain this exploitation in much detail.

3.1.1. Tor Hidden Services

An important feature of the Tor network is provisioning of Tor service through a hidden server. A series of protocols used by hidden server and Tor users can make location of hidden server invisible to client [86]. However, several studies listed below address the deanonymization of hidden servers.

Locating Hidden Server: Overlier and Syverson [46] presented new attack strategies to detect the location of hidden servers using only one Tor node. They proposed changes in route selection and relay selection to increase anonymity. The average duration of the attack varied from minutes to a few hours. The various attacks they considered included the timing signature analysis attack, service location attack, predecessor attack and distance attack. Their proposed solution included introducing middleman nodes to connect to rendezvous points, introducing dummy traffic, extending hidden server path to rendezvous point and using guard entry nodes.

Timing Signature Attack: Elices *et al.* [47] presented a fingerprint analysis attack for Tor's hidden services. They used timestamps from logs of machines hosting hidden services on the Tor network to generate detectable fingerprints. The authors studied delay properties of the Tor network and other users' log entries to make the fingerprint attack feasible.

Application Layer Correlation Attack: Zhang *et al.* [48] described an application level HTTP-based attack for Tor's hidden services. Time correlation was used to assess the resemblance between web accesses and the traffic generated in a compromised Tor router. This attack assumes that the compromised onion router can operate as an entry relay.

Detection, Measurement and Deanonymization of hidden services: Biryukov *et al.* [49] analyzed weaknesses in hidden services which can be exploited by attackers to detect, measure and deanonymize hidden services running over the Tor network. Services of three different applications were analyzed, (1) Botnet for command and control, (2) Silk Road[3] and (3) DuckDuckGo[4]. The study identified major flaws of Tor that included the inflation and cheating of bandwidth by a corrupted relay node, and cheating marking mechanism of flags in Tor network from attacker relay node.

3.1.2. Tor Traffic Detection

A number of studies focus their research on the identification of Tor traffic form other network traffic. These studies suggest that differentiation of traffic can ultimately be used to block Tor traffic, as done by China a number of times in the recent past [55]. Various approaches for traffic identification are summarized as follows.

Tor Traffic Identification from Network Traffic: Bai *et al.* [50] studied the traffic identification mechanisms

[3]Silk road was an online market place which provided anonymity to its customers by way of the Tor network. It was used in great part for the sale of drugs and illegal materials and was shutdown by the FBI. Defunct Website: `http://silkroad6ownowfk.onion`

[4]DuckDuckGo is a search engine that does not track its users and provides anonymity to users by giving same search results for any query to all users. Website: `https://duckduckgo.com`

Table 4: Research works focused on Tor's deanonymization. Table entries symbolize attacks (Att.), counter-attacks (Cou. Att.), Analysis (Ana.), relays (Rel.), Autonomous systems (AS), browser (brows.), server (serv.), decoy traffic (Dec. Traf.), protocol messages (Prot. Mess.), traffic interception (Traf. Interc.), Flag cheating (Flag Cheat.).

Research	Focus			Exploited Tor's weakness								Idea
	Att.	Cou. Att.	Ana.	Compromised Rel.	Compromised AS	Compromised brows.	Compromised serv.	Dec. traf.	Prot. mess.	Traf. interc.	Flag Cheat.	
Overlier and Syverson [46]	✓		✓							✓		Timing signature analysis attack, service location attack, predecessor attack and distance attack
Elices *et al.* [47]	✓									✓		Fingerprint analysis attack
Zhang *et al.* [48]	✓									✓		Application level time correlation attack
Biryukov *et al.* [49]	✓	✓									✓	Using corrupted relay node and cheating Tor's flag marking mechanism
Tor's Traffic identification												
Bai *et al.* [50]	✓									✓		Packet examination, context checking and matching
Barker *et al.* [51]	✓									✓		Using unsupervised machine learning techniques over packet sizes
AlSabah *et al.* [52]	✓									✓		Application level time correlation attack
Houmansadr *et al.* [53]	✓							✓				Passive and active attacks to bypass traffic imitation technique
Chakravarty *et al.* [54]	✓	✓										Observing bandwidth fluctuations through compromised node
Winter and Lindskog [55]	✓									✓		Using port tuples
Tor attacks												
Sulaiman and Zhioua [56]	✓			✓								Using unpopular ports over compromised relays
Chan-Tin *et al.* [57]	✓			✓			✓					Using malicious servers to observe traffic fluctuations over relays
Pries *et al.* [20]	✓			✓				✓				Passing duplicate cells through compromised entry relay
Wang *et al.* [58]	✓			✓								Returning malicious page through compromised exit relay
Wagner *et al.* [59]	✓			✓		✓						Compromised exit node to send images which is used by semi-supervised learning algorithm
Benmeziane and Badache [60]	✓								✓			Using destroy and DNS requests for man-in-the-middle attack
Jansen *et al.* [61]	✓								✓			Using valid protocol messages over relays to perform denial of service attack
Abbot *et al.* [62]	✓			✓		✓						Use compromised relay and user's browser for man-in-the-middle attack
Evans *et al.* [63]	✓			✓								Use exit relay to inject javascript for DoS attack
Bauer *et al.* [64]	✓			✓								Advertise low bandwidth to divert traffic towards malicous nodes
Edman and Syverson [65]	✓				✓							Using autonomous systems to breach Tor traffic
Barbera *et al.* [21]	✓						✓					Perform DoS attack by placing large load on Tor routers
Le Blond *et al.* [66]	✓			✓								Using peer-to-peer applications with compromised exit relays to deanonymize users
Geddes *et al.* [67]	✓			✓								Exploiting compromised exit node to advertise high stats to attract large traffic
Chakravarty *et al.* [68]		✓							✓			Using decoy traffic to detect traffic interception
Tor traffic analysis attacks												
Johnson *et al.* [69]	✓									✓		Correlation based attacks using a single compromised relay
Murdoch and Danezis [70]	✓			✓						✓		Use timing signature attack by passing large traffic from corrupted node to Tor relays
Chakravarty *et al.* [71]	✓						✓			✓		Generate traffic between two servers and map relays through correlation
Zhang *et al.* [72]		✓								✓		Suggested priority queue algorithm to bypass correlation between load and latency
Song *et al.* [73]	✓									✓		Use time and stream size with k-means algorithm to deanonymize users
Panchenko *et al.* [74]	✓									✓		Use volume, time and direction for classification
Wang and Goldberg [22]	✓								✓			Use Tor cells for website fingerprinting
Jin and Wang [75]	✓									✓		use wavelet based decomposition to estimate timing distortion
Gilad and Herzberg [76]	✓								✓			Breach by off-path TCP connection or eavesdrop on clients
Tor Improvements												
Gros *et al.* [77]		✓	✓	✓								Proposed Honeywall to rank node's reliability
Winter and Lindskog [78]		✓		✓								Proposed exit relay scanner to avoid misuse of exit node
Xin and Neng [79]		✓		✓								Proposed a tuning mechanism to keep a track of reliable nodes
Backes *et al.* [80]		✓	✓						✓			Identified flaws in current key exchange mechanisms
Marks *et al.* [81]		✓							✓			Suggested separate bi-directional TCP links to increase anonymity
Nowlan *et al.* [82]		✓	✓						✓			Suggested use of uTCP and uTLS to avoid head of line blocking problem
Danner *et al.* [83]		✓	✓	✓								Investigated DoS attack and proposed improvements to avoid it
Anonymity without Tor												
Herzberg *et al.* [84]		✓	✓					✓				Suggested camouflaged web server by mimicking GMAIL traffic
Mendonca *et al.* [85]		✓			✓							Proposed concealed source identifier through network service provider

of popular anonymity tools, i.e., Tor and Web-Mix. Authors used fingerprint identification (packet examination and packet context checking) followed by matching to identify the traffic. Key attributes used for traffic identification from other network traffic include specific strings, packet length and packet transmission frequency in the network.

Differentiate Tor Traffic from Encrypted Traffic: Barker *et al.* [51] showed that traffic from the Tor network can be differentiated form encrypted traffic in the network. They captured regular HTTPS, Tor HTTPS and HTTP traffic routed through Tor and analyzed their packet sizes and developed an unsupervised machine learning (ML) classifier that operates only on packet size attribute with 97.54% true positive (TP) and 1.06% false positive (FP) rates.

Differentiating Tor Traffic: AlSabah *et al.* [52] developed an ML classifier to differentiate web traffic from bulk download traffic. AlSabah *et al.* used the following four features to classify Tor traffic: (1) Circuit lifetime, (2) data transferred, (3) cell inter-arrival times, and (4) recently sent cells. They tested naïve Bayes, Bayesian networks and decision tree classifiers. Using the proposed classification method, they reported 75% improvement in responsiveness and 86% decrease in download rates.

Fingerprinting Tor traffic: Houmansadr *et al.* [53] aimed to differentiate the traffic of anonymous networks from other network traffic. They claimed that mimicking other traffic is an obsolete way for anonymity. They devised a number of passive and active attack strategies to breach anonymous networks. Their study suggested the use of partial imitation and use of new strategies by incorporating popular protocols like HTTPS email etc.

Tor Proxy Node Identification: Chakravarty *et al.* [54] described a novel attack that identifies all Tor relays participating in a given circuit. The attack modulates the bandwidth of an anonymous connection through a compromised server, router or an entry point and observes the resultant fluctuations in the Tor network using *LinkWidth* [87]. LinkWidth sends a train of pulses comprising of alternate TCP-SYN and TCP-RST packets and capacity is computed at the receiver end by estimating packet dispersion. Authors reported a 59.46% TP rate and 10% true negatives rate for compromised Tor relays using the proposed strategy.

Identification of Tor Bridges: Winter and Lindskog [55] conducted an extensive investigation into the blocking of Tor relays and bridges by China. Their investigation showed that Tor bridges were blocked by port tuples, rather than IP addresses and that bridges were blocked only when they were active. Their investigation also showed that adversaries did not conduct traffic fingerprinting for domestic traffic and that packet fragmentation could be used to circumvent China's firewall.

Fingerprinting Keywords in Search Queries: Oh *et al.* [88] investigated the viability of keyword fingerprinting attacks in the Tor network. Study showed that effective feature selection can help any passive adversary in figuring out the identity of the user. Time and volume of traffic play the most crucial role in determining the identity of the user. Among other features keyword sets, incremental search and high security search are other features used for classification. For experiments, authors collected a keyword dataset containing $160,000$ search queries of google. For one of 300 targeted keywords of Google, experimental results demonstrated recall, precision and accuracy of 80%, 91% and 48%, respectively.

3.1.3. Tor Attacks

Attacking the Tor network is an interesting research dimension which ultimately aims to block access to it. Several attempts by China and other countries have failed in the recent past because Tor is being improved continuously [55]. In this subsection, we summarize various studies covering Tor attacks.

Unpopular Ports Attack: Sulaiman and Zhioua [56] described an attack they developed which can compromise circuits in the Tor network. Their attack takes advantage of unpopular ports in the Tor network. Sulaiman and Zhioua added a small number of compromised entry /exit relays to the Tor network (~ 30 relays) which permit the use of unpopular ports. By doing so, 50% of developed circuits can be compromised, which significantly decreases the anonymity of the Tor network.

Circuit Clogging Attack: Chan-Tin *et al.* [57] proposed an attack that can identify the Tor routers used in any circuit. For the proposed attack a client connects to a malicious server which sends data to the client in large bursts and in small amounts. During large bursts, Tor routers take long times to process the extra amount of data. Authors showed that continuous monitoring of all Tor relays can identify the Tor relays used in the particular circuit. A mechanism to detect the behavior of malicious routers by the client was also evaluated, which measured network latency of the client.

Replay Attack: Pries *et al.* [20] suggested a replay attack to detect the exit routers in the Tor network. The replay attack assumes that the entry onion router is compromised. The replay attack duplicates packets coming from a sender. Tor uses counter-mode Advanced Encryption Standard (AES-CTR)[89][5] for encryption and decryption, any duplicate cells will give a cell recognition error at the exit routers. This behavior leaks exit router information to the entry router by simple correlation.

Flow Multiplication Attack: Wang *et al.* [58] designed a flow multiplication attack similar to a man-in-the-middle attack. The attack assumes that the exit router is compromised. Whenever a client sends a request to target server, the exit router returns a malicious page which triggers certain fetch requests in the client browser over the same circuit. An accomplice at the entry router can see

[5] Advanced Encryption Standard (AES) is an encryption standard which is based upon substitution-permutation technique. It has three members Rijndael family each with block size of 128 bits and key lengths of 128, 192 and 256 bits.[90]

the requests, and together with knowledge of the exit relay, identify the complete Tor circuit.

Attack Using Game Theory and Data Mining: Wagner *et al.* [59] proposed an attack which exploits the exit malicious exit node to cluster observed traffic flows using an active tag injection scheme. The proposed method has two steps, (1) image tags are injected into HTML replies from the exit node to the user, and (2) a semi-supervised learning algorithm based upon deep data mining is used to reconstruct the entire browsing session of the user. The authors model the Tor network in form of a game theoretical concept where all Tor users and rogue nodes play a game for identification of malicious node. Once a rogue node has been identified, it's game is over because no other user uses it due to presence of special flag in it. Authors main aim is to work over over the equilibrium between rogue nodes and Tor users.

Attack using Destroy and DNS Requests: Benmeziane and Badache [60] investigated possible breaches of Tor targeting its network requests. They exploited *destroy requests* (Tor's circuit destruction requests) and DNS requests to break anonymity. Destroy requests are not encrypted, which poses a serious threat to Tor. Moreover, a local eavesdropper can use the man-in-the-middle attack strategy against DNS requests, which are unprotected.

The Sniper Attack: Jansen *et al.* [61] presented the Sniper attack, a low-resource denial-of-service (DoS) attack against the Tor network which can disable arbitrary relays. The adversary builds a Tor circuit through the target relay and starts obtaining a large file by continuously sending the SENDME cells (protocol messages for continuously receiving the file), which increases the congestion window size. By repeating over multiple circuits, memory of host of target relay would exhaust which can disrupt the functioning of Tor relay. Experiments showed that an adversary can consume $2,187$ KB/s memory of a victim relay at the cost of very little bandwidth and decrease Tor network bandwidth by as much as 35%.

Browser-based Attacks: Abbot *et al.* [62] proposed a novel attack that tricks a user's web browser into sending a distinctive signal over the Tor network (by installing a Java or HTML script). An attacker that controls an exit relay can use it in a man-in-the-middle attack to mirror and forward duplicated traffic to a malicious server. By analyzing the data, the malicious server can deanonymize the Tor user. However, this study makes two significant assumptions: the ability to control the exit relay and the ability to configure / compromise a targeted user's web browser.

Congestion Attack using Long Paths: Evans *et al.* [63] proposed an extension to the congestion attack proposed by Murdoch and Danezis [70] owing to the enormity of the current Tor network. Evans *et al.* proposed the combination of Javascript injection and DoS attack. A Tor exit relay is used by the attacker to inject Javascript code into a user's browser, which makes the browser send a response every second. They suggested modifications such as disabling JavaScript, thwarting DoS attack by disabling ability to control latency of routers. In the modified design, routers keep a track of all paths with flags and disable any request for latency by using flags.

Exploiting Routing Algorithm: Bauer *et al.* [64] exploited Tor's routing algorithm to steer a disproportionate number of users towards selecting their entry and exit relays from a set of malicious Tor routers. Bauer *at al.* suggested that low-latency constraints force Tor's routing algorithm to prefer nodes advertising high bandwidths. Instead of performing complex traffic analysis techniques, the authors suggested to collect detailed flow logs from malicious nodes (both entry and exit nodes) and use the information of node selection to deanonymize flows.

Analyzing Autonomous Systems for Tor Path Selection: Edman and Syverson [65] analyzed the effect of autonomous systems (AS) for path selection in Tor network. They studied the selection of AS residing in different countries and found it quite effective. Traffic analysis of the Tor network showed that majority of traffic passes through a few ASs because all established paths focus over latency and anonymity which occurs better in some ASs. Analysis shows that increase in relays has not increased the diversity to a large extent.

DoS Attack using Cell Flooding: Barbera *et al.* [21] presented a novel attack which generates a few circuits requiring large computing and networking resources. Their study showed that this attack requires only 0.2% resources for old routers and $1 - 16\%$ router resources for new attacks, which makes it an inexpensive attack to execute. Barbera *et al.* proposed a mitigation scheme by placing an upper cap on the utilization of resources at routers.

Exploiting peer-to-peer application: Le Blond *et al.* [66] suggested that peer-to-peer applications can be exploited to trace IP addresses of users running Tor. Moreover, scan of malicious Tor exit relays should be used to correlate various user streams for deanonymization. Experiments showed that their 'bad apple' attack was able to identify 193% more streams, including 27% HTTP streams, and reveal IP addresses of $10,000$ Tor users. This constituted 9% of all the flows passing through the exit relays under their control.

Induced Throttling Attacks: Geddes *et al.* [67] proposed a new attack which breaches the Tor network by exploiting its selection bias in favor of high capacity relay nodes. Authors showed that induced throttling at the corrupt exit node by exploiting congestion or traffic shaping algorithms can induce similar traffic patterns at other relays associated with the corrupted exit relay.

Using Decoy Traffic: Chakravarty *et al.* [68] used decoy traffic on anonymous networks to detect traffic interception. The proposed strategy is based on the idea of injecting traffic containing bait credentials for decoy services requiring user authentication. Chakravarty *et al.* set up decoy IMAP and SMTP servers and identified ten instances of traffic interception over ten months.

3.1.4. Tor Traffic Analysis Attacks

A few studies have focused on the analysis of Tor traffic for breaching this network. Analysis shows that traffic analysis can provide an efficient mechanism for deanonymization. A few of these studies are summarized as follows.

Traffic Correlation Attacks: Johnson *et al.* [69] conducted a thorough analysis of the Tor network with a deep focus on the development of a threat model. They built the Tor path simulator (TorPS) to assess Tor's vulnerability to correlation based attacks. Their study suggested that a single Tor relay adversary can deanonymize 80% of users within six months. This research showed that set of relays is dependent upon the user's application which reduces security of the Tor network.

Using Timing Signature: Murdoch and Danezis [70] presented a simple mechanism to evaluate the Tor nodes being used in a circuit. In the proposed scheme, a malicious node sends probe data to the Tor relays. All Tor relays used in the circuit will experience a delay and client-server communication will be modulated. Hence, correlation between delay and modulation gives insight about the relays being used in a circuit.

Traffic Analysis Attack: Chakravarty *et al.* [71] used NetFlow data to analyze the effectiveness of traffic analysis attacks against Tor network. Their proposed attack creates variations in traffic at the server end and observes the effects at a colluding server at the other end. They reported 81.4% accuracy in real-world experiments with 6.4% FP rates.

Queue Scheduling and Resource Allocation: Zhang *et al.* [72] proposed a priority queue scheduling mechanism to reduce the correlation between high load and high latency which would ultimately increase the level of anonymity. However, increase in anonymity comes at the cost of latency which degrades quality of service at the user end. Extensive experiments using the proposed mechanism showed an increase in anonymity due to decrease in correlation between load and latency.

Correlation Using K-means Algorithm: Song *et al.* [73] applied machine learning techniques to deanonymize Tor flows at the first hop and last hop in the network. They used the time / stream size tuple of attributes together with the *k-means* algorithm to deanonymize by matching first hop traffic with last hop traffic. Their results showed that as little as 8 packets are enough to deanonymize a Tor stream with greater than 99% accuracy.

Website Fingerprinting Using Machine Learning: In [74], Panchenko *et al.* suggested the use of machine learning approaches requiring feature selection and classification for website fingerprinting. Authors used Support Vector Machine (SVM) classifier with various features including packet sizes (except 52 size packets because of excess use in acknowledgements), packet size markers to express direction of flow, HTML markers, total transmitted bytes, number markers, occurring packet sizes, percentage incoming packets and number of packets. Extensive research showed that volume, time and direction of the traffic were the most promising features and classification of close-world and open-world dataset gave 55% detection rate. However, camouflaging the traffic decreased the detection rate to 3%.

Website Fingerprinting Using New Metrics: Wang and Goldberg [22] proposed the use of Tor cells as a unit of data transfer rather TCP/IP packets for website fingerprinting. Authors collected data using realistic assumptions on adversaries from client to entry guard node. The study suggested the removal of SENDME cells as they do not play any significant role in improving performance. Proposed metrics use the observation that dynamic content is present in only incoming packets and it is present at the end of the packets. Upto 95% recall rate and 0.2% FP rate is observed using SVM classifier.

Wavelet Decomposition Attack: Jin and Wang [75] suggested a wavelet based decomposition mechanism to estimate the distortion in timing at the receiver end of Tor network. Authors showed that wavelet based multi-resolution analysis (MRA) captures the variability of the timing distortion at all levels, with better granularity than traditional estimation of timing distortion. Deanonymization rate of 96% was obtained for Tor at a packet rate of 4 pkts/sec in 3 minutes without changing established paths (circuits) of Tor. Analysis showed that Tor circuit rotation could decrease the accuracy of deanonymization to 72% after 5 Tor circuit rotations in 3 minutes.

Exploiting Side-channels to Identify Clients: Gilad and Herzberg [76] exploited three kinds of side-channels including (1) globally incrementing IP identifiers, (2) packet processing delays, and (3) bogus-congestion events. Sequential port allocation is also used to identify the clients. Two scenarios for breaching have been presented including (1) fully off-path attack to detect TCP connections, and (2) detecting Tor connections by eavesdropping on clients.

3.1.5. Tor Improvements

In this section, we present some miscellaneous studies about improvement in Tor network by focusing on Tor relays, path selection mechanisms, transport layer protocols and application layer improvements.

Misusing Tor Exit Node: Gros *et al.* [77] studied the abuse and misuse of Tor exit nodes to compromise the anonymity of the Tor network. Authors proposed a mechanism, called *Honeywall*, to avoid misuse of any Tor exit node. According to Honeywall, whenever any exit node detects a malicious behavior, it lowers the reputation of the immediate predecessor router and also sends an alert to it. Similarly, the intermediate router lowers the reputation of its predecessor router. Through this strategy, all "bad" nodes eventually end up with lower reputations and all "good" nodes have higher reputation.

Exposing Exit Relays: Winter and Lindskog [78] detected malicious exit Tor relays and profiled their behavior. An exit relay scanner was built to identify all outgoing Tor

traffic and identify the malicious nodes and avoid man-in-the-middle attacks. Patches were built for the Tor browser bundle to collect certificates through multiple paths to check authenticity of the destination server.

Tuning mechanism for Tor: Xin and Neng [79] showed that Tor lacks the evaluation system for the node store. Authors presented and theoretically analyzed a tuning mechanism for Tor. The proposed tuning system included the establishment of an evaluation system and optimization of Tor node store and output mode. Through the evaluation system, all nodes are ranked based on their anonymity, uptime, bandwidth and latency. In the optimization stage authors suggested to use a fixed number of circuits such that traffic load has least effect on latency.

Increasing Security of Tor Network: Backes *et al.* [80] conducted research on the security of Tor network for anonymous browsing and presented a novel security protocol. Authors elaborated the concept of security in anonymity softwares. Their study showed that current key exchange algorithms are inefficient and a number of security enhancements were suggested including cryptographic requirements for secure browsing.

Transport Layer Improvements: Marks *et al.* [81] studied TCP based deficiencies in the Tor network. By studying the transmission mechanism of Tor, authors proposed to split bidirectional links into two separate TCP links. Experiments with separate TCP links showed 100% increase in throughput with a decrease in throughput variance from 43000 KB/s to 10000 KB/s [91].

Switching to uTCP and uTLS: Nowlan *et al.* [82] probed into the cross-stream head of line blocking problem of TCP in the Tor network. Their study suggested the use of unordered TCP (uTCP) and unordered TCL (uTLS) for reducing inter-dependence in inter-leaving streams, due to the requirement of low latency in the Tor network.

Feasibility of DOS attack over Tor: Danner *et al.* [83] conducted a deep investigation on the feasibility of Denial-of-Service (DoS) attack (proposed in a previous study by Borisov *et al.* [92]) over Tor network. Authors showed through simulations and analytical evaluations that corrupted relay nodes can be used to exploit Tor network and perform DoS attack. Authors suggested the use of reliable guard nodes (entry and exit) which can decrease the probability of selection of a corrupt Tor relay.

3.1.6. Anonymity without Tor

To present a glimpse of studies providing anonymity without Tor, we present a few studies focusing on packet encapsulation and central server based anonymity mechanism.

Anonymity Using a Central Server: Herzberg *et al.* [84] proposed a camouflaged browsing design using a camouflaged server. The basic idea is to communicate with the camouflaged server using a manner similar to popular web services. Encrypted communication, URLs of GMAIL with packet frequency and sizing similar to GMAIL can easily pass unnoticed through any adversary. Although this design provides better anonymity, it suffers from a single point of failure.

In-Network IP Anonymization Service: Mendonca *et al.* [85] presented a novel idea of user anonymity by working with a network service provider. Proposed service *AnonyFlow* used an in-network IP anonymization service. The fundamental idea was to conceal the source identifier from the other side of the network. An OpenFlow based implementation was used for performance evaluation. However, anonymity could be breached by compromising the network service provider.

3.2. Tor Path Selection

Tor selects three relays based upon its path selection algorithm which incorporates anonymity and reliability characteristics of relays and users [93]. By compromising the path selection mechanism, the complete anonymity mechanism of Tor can be breached. In this subsection, we present studies covering (1) new algorithms for path selection, and (2) analysis of path selection algorithms. An overview showing the classification of major studies is shown in Figure 6.

Table 5 presents a comparison of various research works in Tor's path selection track. Comparison shows that performance and anonymity were the most frequently studied parameters for path selection. However, majority studies neglected autonomous systems, relay locations, hop counts, multi-path mechanisms, load and relay capacity. Moreover, majority research works focus on the development of new algorithms while few studies analysed the current path selection algorithms.

3.2.1. New Path Selection Algorithms

LASTor - Low Latency with Better Anonymity Algorithm: Akhoondi *et al.* [94] proposed a new path selection algorithm namely *LASTor*. LASTor incorporates the locations of relays before choosing paths and does not always select the shortest path as it reduces the entropy of path selection. Moreover, LASTor avoids paths passing through ASs which can compromise anonymity of the system by traffic correlation.

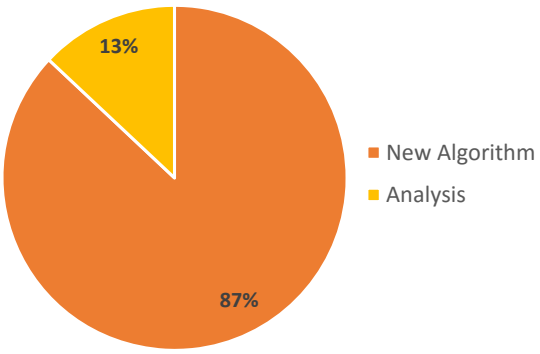

Figure 6: Focus of published research on Tor's Path Selection approaches.

Table 5: Research works on Tor's path selection. Table entries symbolize New algorithms (New Algo), Analysis (Anal.), Autonomous Systems (AS), Relay Locations (Relay loc.), Hops, Performance-Latency-Bandwidth (Perf., Lat, BW), Multi-path, Load, Relay Capacity (Rel. Cap.) and Anonymity (Anon).

Research	Focus		Path Selection Parameters								Idea
	New Algo	Anal.	AS	Relay loc.	Hops	Perf. Lat, BW	Multi-path	Load	Rel. Cap.	Anon.	
New Path Selection Algorithms											
Akhoondi *et al.* [94]	✓		✓	✓							Included relay locations and autonomous system reliability
Chen *et al.* [95]	✓			✓	✓	✓				✓	Included hops and geographic distance in path selection
Karaoglu *et al.* [96]	✓					✓	✓				Studied multipath design
Panchenko *et al.* [97]	✓					✓		✓	✓		Studied Load and Capacity at nodes
Li *et al.* [98]	✓					✓				✓	Proposed tunable mechanism varying between anonymity and performance
Panchenko *et al.* [99]	✓	✓		✓		✓			✓		Studied latency, link capacity and load at nodes
Liu and Wang [100]	✓					✓		✓		✓	Proposed random walk based circuit building protocol
Liu and Wang [101]	✓					✓				✓	Proposed new relay selection mechanism with backup circuit algorithm
Snader and Borisov [102]	✓					✓	✓			✓	Studied malicious nodes, proposed balance between anonymity and performance
Li *et al.* [103]	✓					✓				✓	Proposed relay recommendation system
Tang and Goldberg [104]	✓					✓					Suggested the use of bursty circuits instead of busy paths
Wang *et al.* [105]	✓					✓					Included latency as a measure of congestion in path selection
Snader and Borisov [106]	✓					✓				✓	Suggested opportunistic bandwidth measurement with priority based traffic handling
Elahi *et al.* [107]	✓	✓		✓							Discouraged short term entry guard churn and time-based entry guard rotation
Analysis of Path Selection											
Bauer *et al.* [108]		✓		✓					✓		Suggested random or Snader-Borisov approach for router selection
Wacek *et al.* [109]		✓				✓				✓	Suggested bandwidth weighted relay selection and avoidance of congested circuits

Using MultiPath Routing Karaoglu *et al.* [96] evaluated the multipath design for Tor network to avoid congestion and overcome the limitations in Tor's circuit construction. Evaluations revealed a four-fold increase in throughput with better load balancing and traffic mixing. However, high buffer costs at the Tor proxies were the major limitations of multipath design.

Path Selection Using Load *and* Capacity *of Nodes:* Panchenko *et al.* [97] studied the delays in the Tor network and provided new measures in path selection to improve user experience. Two factors used for path selection design are "load" at the nodes and maximum "capacity" at the nodes. Authors showed that these factors can increase the performance by 70%. Their study concludes that nodes, not edges, are the deciding factors for performance.

Tunable Mechanism of Tor: Li *et al.* [98] emphasized the development of a tunable mechanism for Tor users depending on *anonymity* and *performance* required by users. Authors used "path length" as a metric to tune user requirements based upon anonymity and performance followed by client side modifications of Tor protocol. Results showed that browsing time deteriorates quickly from 14.4 to 140.1secs with a 37.3% increase in failure rate by increasing the path length from 2 to 6. The proposed mechanism requires only client side modification.

Using Latency and Link Capacity: Panchenko *et al.* [99] evaluated the impact of different factors on the performance of the Tor network. Factors considered included overloaded nodes and links and geographical diversity of nodes. Authors presented a novel path selection algorithm based on latency experienced by the nodes and link capacity. Metrics used for evaluations included circuit setup duration, round trip time (RTT), stream throughput and influence of penetration.

Random Walk Based Circuit Building Protocol: Liu and Wang [100] presented a random walk based circuit building protocol (RWCBP) which is a two-step method: circuit construction, followed by application message transmission. Network latency, computational latency and transmission loads were used to analyze the performance of the proposed protocol. Using indexes of performance and anonymity, resilience of the proposed protocol was ana-

lyzed.

New Circuit Building Protocol: Liu and Wang [101] studied the current protocol of the Tor network and proposed a novel circuit building design with two phases: selection of user selectable relay nodes and circuit construction. Authors presented enhancements in the selection of relay nodes, fast circuit construction and backup circuit algorithm. Better performance and user experience are obtained with the new protocol while achieving the same level of anonymity.

Tunable Path Selection for Better Security and Performance: Snader and Borisov [102] addressed the issue of the selection of malicious nodes in the path selection due to self-advertised bandwidth. Authors proposed an algorithm which is based upon the anonymity and performance in the Tor network. Significant performance gains were observed using the proposed strategy with single and multipath route selection.

Relay Recommendation System: Li *et al.* [103] proposed a relay recommendation system to provide reliable information about all relays for building circuits (paths). Its main goals include the mitigation of low-resource attacks, better performance and tradeoffs between anonymity and performance. Authors proposed path selection algorithms for increased anonymity. Significant performance gains with increase in anonymity were observed in the simulations of the proposed scheme.

Preferring Bursty Circuits over Busy Circuits: Tang and Goldberg [104] proposed a new algorithm which suggests the use of bursty circuits instead of busy circuits. Authors suggest that bursty circuits (such as web browsing) can provide less latency than the busy circuits (used for bulk data transfer). Proposed circuit selection algorithm uses exponentially weighted moving average (EWMA) of cells sent on any path and uses the path with lowest EWMA (because new and bursty paths have high EWMA).

Incorporating Congestion in Path Selection: Wang *et al.* [105] proposed a novel path selection algorithm which incorporates the latency of nodes as a measure for congestion. The proposed algorithm favors nodes which provide lower latency. Study suggests that node latency is greater than the link latency in majority of the cases. Authors conclude that the proposed algorithm can reduce latency by upto 40%.

Opportunistic Bandwidth Measurement Algorithm: Snader and Borisov [106] addressed Tor's shortcoming of favoring high bandwidth nodes based on advertised bandwidth. Their study showed that an opportunistic measurement of bandwidth for all routers by other connected routers can reduce the vulnerability risk by any adversary in Tor. Moreover, priority based traffic handling, i.e., high performance or high anonymity can reduce the risk of partitioning attacks.

Analyzing and Improving Entry Guard Selection: Elahi *et al.* [107] conducted an in depth investigation on the selection of entry guards in Tor network. The study showed that short-term entry guard churn and explicit time-based entry guard rotation result in an increased usage of entry guards in clients, which results in a greater number of profiling attacks.

Trust-Aware Path Selection Algorithm: Johnson *et al.* [110] proposed a path selection algorithm which uses the probability based distribution to keep itself aware of the location of adversaries in the Tor network. In developing trust based model, authors take the relays uptime as the most trustworthy factor in determining the selection of the path. Bypassing the paths containing adversaries can mitigate the traffic analysis attacks conducted by the adversaries.

Investigating Tor's Exit Policies: Liu and Wang [111] studied the exit policies of the exit nodes and addressed the short-comings in the current Tor architecture. A new protocol was proposed which comprised of three parts: (1) reporting misbehavior protocol, (2) building global blacklist protocol, and (3) blocking misbehavior protocol for users. User experience, performance and anonymity were the key indexes used for evaluation.

3.2.2. Analysis of Path Selection

Predicting Path Compromise: Bauer *et al.* [108] showed that the current mechanism of Tor is vulnerable to path compromise because Tor selects paths based on bandwidth capabilities of routers. Study shows that the application level protocol is a significant factor to predict path compromise. Research suggests that router selection should be random or through Snader-Borisov approach to avoid any bias in router selection. Study showed that most robust applications for path compromise are HTTP and HTTPs applications while the most vulnerable are peer-to-peer applications.

Optimizing Hops, Performance flags and Geographic Distance: Chen and Pasquale *et al.* [95] studied the path selection mechanism by varying the number of hops, performance ratings and changing the geographic distance between routers. Trade-offs between anonymity and other parameters (latency etc.) were extensively evaluated. The authors concluded that reduction in hops and geographic distance can increase throughput and decrease anonymity.

Empirical Evaluation of Relay Selection: Wacek *et al.* [109] evaluated the relay selection mechanism of Tor to estimate latency. Performance and anonymity were analyzed for a number of relay selection techniques under varying load conditions. The authors suggest that a combination of bandwidth-weighted relay selection and avoidance of congested circuits can provide better throughput and less latency.

3.3. Tor Analysis and Performance Improvements

In this section, we cover studies on Tor dealing with its analysis and performance improvement mechanisms. Classification of various studies is shown in Figure 7.

Table 6 presents a comparison of various generalized studies covering the modelling of Tor network. Comparison shows that majority studies focused over analysis of

Tor network. Moreover, usability analysis and anonymity analysis were the most frequently studied topics followed by performance analysis. Very few research works focused over the sociability issues of Tor network.

3.3.1. General Studies of Tor

Several studies covering pros and cons of Tor and analyzing statistics of Tor referring to users' quality of experience are summarized in the paragraphs below.

Understanding Challenges and Social Factors: In [112], Dingledine *et al.* described the challenges in implementation of Tor and discussed social issues. Tor network design and its details were also discussed with reference to the previous state-of-the-art. Possible avenues for improvements in the Tor network and flaws in the current system were presented including abuse, security implications and perceived social value.

Who is More User Friendly ? Abou-Tair *et al.* [113] focused on the usability of different anonymizing solutions including Tor, I2P[6], JAP/JonDo (Java Anonymous Proxy)[7] and Mixmaster[8]. The installation of all softwares was analyzed with regard to ease-of-use. They measured the bandwidth consumption of all softwares. The authors concluded that I2P and Mixmaster provide better anonymity but are more complex. On the contrary, Tor and JAP are easy to use but comprise somewhat on the degree of anonymity they provide.

Usability Analysis of Tor: Clark *et al.* [114] conducted usability analysis for deployment of Tor and software tools associated with Tor including Vidalia, Privoxy, Torbutton and FoxyProxy. Research showed that all implementations have associated pros and cons. The study presented guidelines for future implementations for maximum usability of anonymity tools. Research spanned over the installation, configuration, usage menu, verification and switch-off features of various anonymity tools.

Safeplug vs. Tor: Edmundson *et al.* [115] analyzed the security provided by Safeplug in comparison to the Tor network. Safeplug[9] is a plug-and-play network device which is plugged into the router and it acts as an HTTP proxy by directing all web traffic through the Tor network. Safeplug was launched to provide ease in access for Tor users. On the contrary, Tor network can be accessed through Tor browser bundle provided by Tor. Study showed that Safeplug was vulnerable to first and third-party trackers, through which users can be deanonymized. Attacker can modify the settings of Safeplug externally through cross-site request forgery (CSRF). Safeplug provided more latency and less protection than Tor.

Robustness of Tor: Barthe *et al.* [116] argued that *robustness* has always been neglected while *privacy* is the issue that receives most attention. Authors defined general and flexible definitions for robustness and studied the Golle and Juels protocol. By identifying the weaknesses in the current protocol, novel enhancements were also proposed for robustness.

Anonymity and Monitoring on Tor: Mulazzani *et al.* [117] addressed the *Monitoring* and *Anonymity* issues in the current Tor network. A dataset was collected over a period of six months. Analysis showed that a sinusoidal pattern in users is observed with half of servers located in Germany and United States. A proposed implementation has been added into *TorStatus*[10], which is the project displaying Tor network status, available routers, bandwidths, hosts and availability history.

Tor Traffic Statistics: Huber *et al.* [118] analyzed the HTTP usage of the Tor network. Research showed that 78% of Tor users do not use Tor using TorButton, which can be used for deanonymization. 1% of Tor requests are vulnerable to piggybacking attacks. 7% requests, pertaining to social networks, contain identifiable information. The authors suggested the use of HTTPS instead of HTTP for secure communication.

Tor Usage Statistics: McCoy *et al.* [119] studied the applications, user countries and usage of Tor network. Statistics collected from Tor showed that non-interactive protocols (BitTorrent traffic), comprising of a minority of connections, consumed majority of resources. Non-secure protocols like HTTP can be exploited by the exit router to log sensitive information. The study suggested a protocol for identification of all exit routers capturing POP3 traffic. Usage statistics revealed that USA, Germany and China are major users of Tor.

Statistical Data of Tor: Loesing *et al.* [120] collected the statistics from the live Tor network to measure two aspects of communication, i.e., (1) country wise usage, and (2) traffic port numbers for exiting traffic. Both these

[6] I2P is an anonymous overlay network which supports both TCP and UDP traffic. Web: `https://geti2p.net/en/`

[7] Java Anon Proxy allows web browsing with pseudonymity using its proxy based system. Web: `https://anonymous-proxy-servers.net/`

[8] Mixmaster is a Chaumian mix network which is an anonymous remailer providing security against traffic analysis and sender deanonymization. Web: `http://mixmaster.sourceforge.net/`

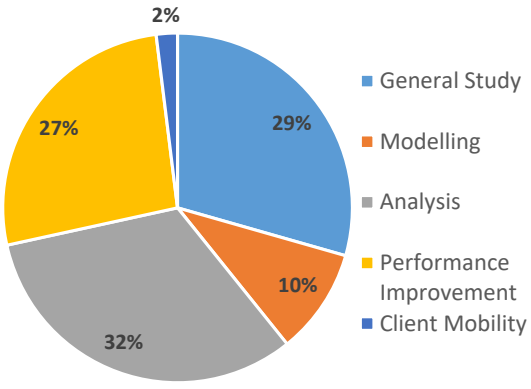

Figure 7: Focus of various research works on the analysis of Tor.

[9] `https://pogoplug.com/safeplug`

[10] `https://torstatus.blutmagie.de`

Table 6: Research works on general study and modelling of Tor. Table entries symbolize Discussion (Dis.), Analysis (Anal.), Propose (Propos.), Sociability Issues (Soci. Issu.), Usability Issues (Usab.), Performance Latency (Perf. Lat.), Performance - bandwidth (Perf. BW), Anonyimty (Anon.).

Research	Study focus			Research Parameters					Idea
	Dis.	Anal	Propos	Soci. issu.	Usab. [St./eas]	Perf. lat	Perf. BW	Anon.	
General Study over Tor									
Dingledine *et al.* [112]	✓			✓					Discussed challenges and social issues, and studied Tor network
Abou-Tair *et al.* [113]	✓	✓			✓		✓	✓	Studied usability, bandwidth and anonymity over anonymous networks
Clark *et al.* [114]		✓			✓			✓	Performed usability analysis of Tor with other anonymity tools
Edmundson *et al.* [115]		✓				✓		✓	Compared anonymity and performance of *Safeplug* with Tor
Barthe *et al.* [116]		✓				✓	✓		Studied robustness in Tor network
Mulazzani *et al.* [117]		✓			✓			✓	Analysed monitoring and anonymity issues in Tor
Huber *et al.* [118]		✓			✓			✓	Studied anonymity using HTTP usage statistics
McCoy *et al.* [119]		✓			✓			✓	Studied applications, usage statistic and misusage of Tor
Loesing *et al.* [120]		✓			✓			✓	Studied country and port usage of Tor
Chen *et al.* [121]			✓	✓				✓	Proposed anonymous payments over anonymous network
Modelling Tor network									
Jansen *et al.* [122]		✓	✓			✓	✓		Proposed graph based Tor topology
Jansen and Hopper [123]			✓			✓	✓		Developed discrete event Tor simulatork
Bauer *et al.* [124]			✓			✓	✓		Developed emulation toolkit *ExperimenTor* for Tor

statistics can be used for future improvements in the Tor network for better anonymity services. The study also revealed that port 80 receives most traffic.

Micropayments Using Tor: Chen *et al.* [121] proposed a novel mechanism of anonymous payments for network services. The proposed mechanism allows users to make untraceable micro-payments to each other. Authors included features of offline verification, overspending prevention, aggregation and low overheads. Experiments showed only 4% overhead for the proposed strategy.

3.3.2. Modeling Tor Network

In this section, we present the modeling techniques used for analyzing Tor.

Modeling Topology and Hosts of Tor: Jansen *et al.* [122] developed a model of Tor which closely resembled the Tor network. Authors developed a graph for Tor topology where vertexes related to downstream bandwidth, upstream bandwidth and packet loss, and edges related to latency, jitter and packet loss. All hosts including relays, authorities, clients and Internet servers were mapped to the developed graph based on characteristics obtained from Tor.

Shadow: Simulating Tor Network Jansen and Hopper [123] developed an open source discrete event simulator for simulating the network layer of Tor on a single machine. Authors compared the performance of *Shadow* with real-world simulation results from the PlanetLab testbed.

Emulation Toolkit for Tor Experimentation: Bauer *et al.* [124] developed ExperimenTor, an emulation toolkit for Tor network. Their research was focused on the toolkit rather than the analysis of the Tor network.

3.3.3. Analysis of Tor

Analysis of Tor network has been a part of many studies covering delays, bandwidth, quality of service, relay selection and authentication protocols. Several studies covering these areas are summarized in following sections.

Table 7 presents a comparison of various research works in analysis and performance improvement track. Comparison shows that relay selection and latency analysis are the most frequently studied topics followed by anonymity, bandwidth and quality of service analysis. Very few studies focused over queues, traffic shaping techniques and protocol messages.

Understanding Delays in Tor: Dhungel *et al.* [125] analyzed the delays in the entire Tor network. Authors suggested that overlay network plays the most significant role in Tor. The study revealed that 11% of Tor routers are overloaded with traffic which resulted in very high delays. In 7.5% of circuits, overall latency introduced a 450ms delays. *Guard* routers incorporate more delay than *non-guard* routers. There is high fluctuation in delay for all routers except for those having high bandwidths.

Measurement and Statistics: Loesing *et al.* [126] studied the latencies inside the Tor network. A deep investigation was conducted to evaluate the individual delays and QoS properties. The authors showed that circuit building time (Introduction and Rendezvous) is the most crucial

Table 7: Research studies on analysis and performance improvements of Tor. Table entries symbolize New algorithms (New Algo), Analysis (Anal.), Relay Selection (Relay Sel.), Performance Latency (Perf. Lat.), Performance Bandwidth (Perf. BW.), Quality of Service (QoS), Queues, Protocol Messages (Prot. Msgs.), Traffic Shaping (Traff. Shap.), Anonymity (Anon.).

Research	Focus		Path Selection Parameters								Idea
	New Algo	Anal.	Relay Sel.	Perf. Lat.	Perf. BW.	QoS	Queues	Prot. Msgs	Traff. Shap.	Anon.	
Analysis of Tor network											
Dhungel *et al.* [125]		✓	✓	✓							Analysed latency for Tor relays
Loesing *et al.* [126]		✓		✓		✓					Analysed latency, QoS and performance of Tor
Ehlert *et al.* [127]		✓		✓	✓						Studied bandwidth and latency in Tor
Pries *et al.* [128]		✓			✓	✓					Investigated bandwidth for various path selection algorithms
Liu and Wang [111]	✓		✓	✓		✓				✓	Proposed relay reliability mechanism considering performance anonymity and QoS
Wang *et al.* [129]		✓	✓								Performed an empirical analysis over family nodes
Tschorsch and Scheuermann [130]	✓			✓			✓				Proposed fairness model for efficient and fair resource allocation
Chaabane *et al.* [131]		✓	✓								Studied misuse of Tor's exit nodes as proxies
Hopper *et al.* [132]		✓						✓			Analysed Tor's performance considering key exchange mechanisms
Lenhard *et al.* [133]		✓		✓							Studied communication overhead in low bandwidth networks for Tor's hidden services
Goldberg [134]		✓	✓					✓			Analysed anonymity with Tor;s authentication protocol
Tor performance improvement											
Jansen *et al.* [135]	✓		✓								Proposed token based performance mechanisms for recruiting more relays
Dingledine *et al.* [136]	✓		✓								Proposed priority based traffic handling for relays
Wang *et al.* [137]	✓		✓							✓	Proposed node reliability mechanism to avoid blockage of bridges
Smits *et al.* [138]	✓		✓					✓		✓	Proposed packet authorization based mechanism to protect bridges from eavesdroppers
Moghaddam *et al.* [139]	✓								✓	✓	Proposed traffic morphing (using Skype traffic) to avoid censorship
Weinberg *et al.* [140]	✓								✓	✓	Proposed traffic shaping (by assembling regular HTTP traffic) to avoid deanonymization
Gopal and Heninger [141]	✓			✓							Suggested latency reduction by separate TCP connections for interactive and bulk traffic
AlSabah *et al.* [142]	✓			✓			✓				Proposed traffic morphing (using Skype traffic) to avoid censorship
Jansen *et al.* [143]	✓			✓						✓	Proposed throttling mechanisms for reducing latency by avoiding bulk traffic

delay period in Tor. Fréchet and exponential distributions were combined to analyze the response times.

Comparing Bandwidth with Latency: Ehlert [127] compared the bandwidth and latency performance of Tor network with the popular I2P network. Authors measured the core latency (HTTP GET requests durations), average latency (webpage download times including external threads and pictures) and bandwidth (download speeds). This research showed that I2P network provides lower core latency and Tor network excels in average latency and bandwidth, owing to the nodes distribution and penetration of the Tor network.

Tor QoS with Path Selection Strategy: Pries *et al.* [128] suggested that TCP suffers severe performance degradation from the random path selection of Tor. Slight QoS improvement is achieved with Tor's bandwidth weighted path selection algorithm. The main reason attributed for small improvements is low bandwidth of Tor routers.

Behavior of Family Nodes: Wang *et al.* [129] presented an empirical analysis of Tor family nodes. A rich dataset of live Tor network comprising of three years was used to study the impact of family nodes. The study suggested that family nodes provide stable and better service than other nodes. Moreover, attacks on family nodes can disrupt the Tor network more severely than random Tor nodes.

Fairness in Tor: Tschorsch and Scheuermann [130] analyzed the fairness issues in the current Tor network. Large unfairness was observed in the current resource allocation mechanism of the Tor network. Authors proposed a max-min fairness based model for efficient and fair scheduling of resources. The proposed design was analyzed with Tor's $N23$ congestion feedback mechanism.

Misuse of Tor: Chaabane *et al.* [131] showed that Tor network was being used for transmitting P2P traffic (Bit torrent etc.) over the Tor network. HTTP and Bit torrent were analyzed on the Tor network. The study showed that Tor exit nodes are being used as one hop SOCKS proxies through tunneling. New techniques were devised to detect such abnormalities in exit nodes' behavior. Research showed that simple crawling over exit nodes can be used to collect as many bridge identities as needed.

Challenges for Hidden Services of Tor: Hopper [132] conducted research on the botnet attacks on Tor through hidden services. Tor exhibits poor network performance due to increased load on relays under such attacks. Hopper attributed the poor performance to the key exchange mechanism of Tor. Study showed the possible research dimensions of limiting request rates from botnets, throttling entry guard, reusing failed partial circuits and isolating hidden services circuits.

Tor Hidden Services in Low Bandwidth Access Networks: Lenhard *et al.* [133] conducted a measurement and statistical analysis for estimating the communication overhead of Tor hidden services in low bandwidth access networks. Research showed that boot strapping time, RTT and circuit building time were the major bottlenecks to performance. Due to numerous delays, an increase in timeout value was suggested to avoid repeated retransmissions.

Analysis of Tor Authentication Protocol: Goldberg [134] analyzed the security of Tor's authentication protocol (TAP). The authors argued that any security breach by a single malicious Tor relay can deanonymize users' sessions. Through empirical evaluations, research showed that TAP is secure in random oracle model.

Statistics Collection Mechanism of Tor: Mani and Sherr [144] analysed the data collection mechanism of Tor through 'PrivEx'. They showed that statistics of PrivEx can be easily compromised by the present of adversary nodes in the Tor network. As a result of shortcomings of PrivEx, authors proposed 'HisTor', a privacy preserving statistics collection mechanism of Tor which is much more diverse than PrivEx. HisTor uses the count of queries by exit nodes and relays in form of a histogram where individual nodes have little control over the aggregate statistics.

3.3.4. Tor Performance Improvement

Owing to the increasing demand for Tor, various studies have proposed performance improvements to cope with future demands. In this section, we present these studies covering Tor node selection, traffic distribution and latency management etc.

Node Recruitment for Tor: Jansen *et al.* [135] focused their research on recruitment of new Tor relays, motivated by the fact that only 1.5% nodes participate as relays. Authors proposed BRAIDS which is a token based mechanism providing high bandwidth to those users who employ BRAIDS. Proposed scheme characterizes traffic into high throughput, low latency and normal traffic. Based upon usage of BRAIDS and node networking stats, tickets are generated which can be used to increase bandwidth.

Encouraging Tor nodes for traffic relaying: Dingledine *et al.* [136] proposed a mechanism to encourage Tor nodes for traffic relaying. Study suggested a priority based traffic handling, which gives more weight (in form of bandwidth and delays) to those nodes contributing resources to Tor. However, all Tor relays carry an additional load of priority based traffic handling. Directory authorities need to assign priority levels to all Tor users participating in Tor relays.

Improving Distribution Mechanism of Tor Bridges: Wang *et al.* [137] improved the distribution mechanism of Tor bridges by implementing node reliability statistics to avoid the blockage of bridges by corrupt nodes. The uptime of assigned bridges is used to give reputation points to users. In case of any blockage of a bridge, a new bridge address is given on payment of earned credit. To ensure anonymity, reputation information is stored on users' systems by using a privacy-preserving technique which cannot be circumvented by malicious users.

Packet Authorization for Tor Bridges: Smits *et al.* [138] proposed BridgeSPA, a packet authorization based mechanism, to protect users of Tor hosting Bridges. All Tor user hosting bridges are susceptible to traffic analysis attacks. To counter this attack, the authors suggest the

transmission of a bridge key by bridge distribution authorities which is valid for a limited time, as determined by the bridge. For any communication with the bridge, Tor users should use that key within the assigned time period.

SkypeMorph - Tor traffic Shaping: Moghaddam *et al.* [139] proposed a new mechanism namely SkypeMorph to avoid the censorship of Tor bridges. The fundamental idea was to hide Tor traffic as Skype video traffic (a widely used protocol). SkypeMorph, which runs side by side with Tor, makes it hard to distinguish Tor traffic from Skype traffic. Two schemes were suggested for traffic morphing, (1) using the target stream attributes, (2) incorporating both source and destination streams to incorporate packet timings. Both streams provided nearly identical performance, but the former had lower complexity.

StegoTorus - Steganographing Tor Traffic: Weinberg *et al.* [140] proposed a novel technique to bypass censorship on Tor. Their scheme is based upon the idea of chopping Tor traffic into multiple streams, resembling HTTP traffic, before passing through the censor. StegoTorus acted as a proxy on Tor clients.

Torchestra - Separate connections for Interactive and Bulk Traffic: Gopal and Heninger [141] proposed the transmission of interactive and bulk traffic over two separate TCP connections among all nodes in the Tor network. Exponentially weighted moving average (EWMA) algorithm was used to distinguish between interactive and bulk traffic on all circuits. Upto 40% reduction in delays was observed as compared to standard Tor for the proposed strategy.

Reducing Latency in Tor: AlSabah *et al.* [142] proposed a mechanism for congestion control and flow control in order to reduce latency in the Tor network. The study suggested the use of small fixed size windows and small dynamic windows which can reduce the packets in flight. For flow control, the study proposed an N23 algorithm which caps the queue lengths of Tor routers and provided a 65% increase in webpage responses and 32% decrease in page loading time.

Throttling Tor bulk users: Jansen *et al.* [143] addressed the poor performance of Tor network using bulk data transfers. Three dynamic throttling algorithms were proposed for reducing network congestion and latency. The guard relay capped the bandwidth capacity of nodes, so, only local relay information was used. Simulations showed that throttling reduces the web page latency and increases the anonymity of Tor network.

3.3.5. Tor Client Mobility

In this section, we study the research works focused on the mobility of Tor network with a particular emphasis on anonymity. Table 8 shows the research works in path selection track and shows that performance and anonymity have been the most frequently studied parameters. Details are presented in below paragraphs.

Using Bridge Relays: Doswell *et al.* [145] analyzed the performance of Tor for wireless devices roaming across multiple networks. Analysis showed that the *speed* of mobile wireless devices significantly affects the circuit building time and Tor's performance. Authors studied the use of bridge relays to provide persistent Tor connections for mobile devices.

New Architectural Designs: Andersson *et al.* [146] proposed several new architectural designs for a mobile Tor network. A trade-off between anonymity and performance was evaluated. Several criteria used in performance estimation included usability, availability, trust and practicality. The study concluded that the single Tor client option offers lowest degree of anonymity.

4. Platforms for Tor Research

In this section, we study the platforms used to study Tor network. Our observations spanning over decades of anonymity research shows that all research works have studied the Tor network using three different techniques, (1) Experiment, (2), Simulations, and (3) Analysis. Figure 8 shows that 60% of the studies used in this paper conducted experiments. Only 27% of the studies conducted experiments. In the experiment section, majority studies developed their own testbed followed by experiments on cloud services and PlanetLab testbeds. In the simulations section, majority research works used extensive simulations to study Tor network. Finally, some studies analyze Tor network by collecting statistics and discussing the sociability and usability issues of Tor network. These three classification categories are elaborated in Figure 9 which shows the platforms used to study Tor network.

4.1. Tor Experiments

Studies covering Tor experiments have focused over several areas including (1) private setup establishment, (2) PlanetLab experiments, (3) cloud services (4) OpenFlow networks and (5) universal composability framework.

Table 9 presents the clients, relays, servers, Tor services and Tor implementation used by various research works. Comparison shows that majority studies deployed their private testbeds with 1-2 clients and 1-2 servers. Several studies deployed limited number of relays for experiments. Number of clients were increased drastically in the PlanetLab and cloud setup for Tor experiments. Moreover, traffic

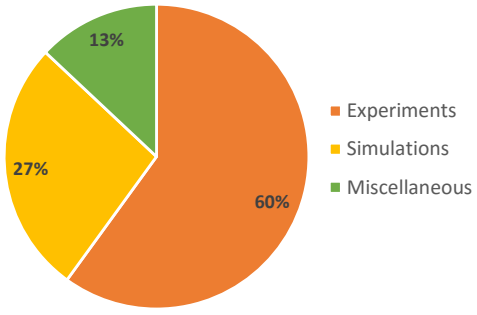

Figure 8: Classification of platforms for Tor's research.

Table 8: Research works on Tor's client mobility. Table entries symbolize New algorithms (New Algo), Analysis (Anal.), Autonomous Systems (AS), Relay Locations (Relay loc.), Hops, Performance-Latency-Bandwidth (Perf., Lat, BW), Multi-path, Load, Relay Capacity (Rel. Cap.) and Anonymity (Anon).

Research	Focus		Path Selection Parameters								Idea
	New Algo	Anal.	AS	Relay Loc.	Hops	Perf. Lat, BW	Multi-path	load	Rel. Cap.	Anon.	
Doswell *et al.* [145]		✓				✓					Suggested bridge relays to avoid bandwidth issues while roaming
Andersson *et al.* [146]	✓					✓				✓	Proposed trade-of between anonymity and performance

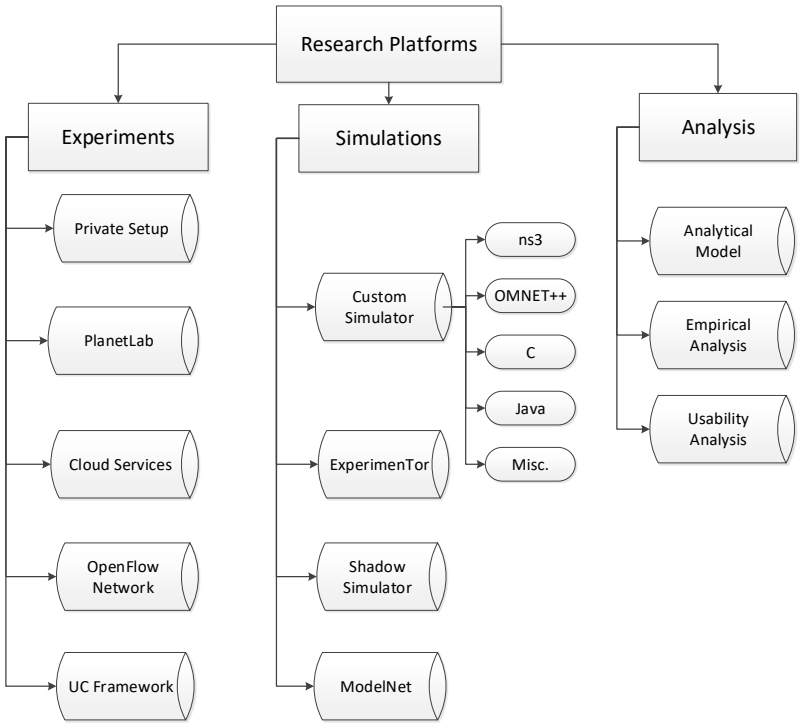

Figure 9: Taxonomy of platforms employed in Tor research.

analysis was the most frequently studied topic. Majority research works used the default Tor setup without any modifications. Figure 10 shows the classifications of experiments on Tor. Analysis of figure shows that majority of studies deployed their own private testbeds.

4.1.1. Private Setup connected with Tor

Overlier and Syverson [46] performed experiments by setting up two nodes (one in Europe and other in US) running hidden services at two ends of the Tor network. Access to webpages and images was provided using these services. The client PC was setup both as a client and a middleman node, and all sampling takes place at this client node.

Andersson and Panchenko [146] performed experiments to verify the performance of their proposed mobile protocol. Mobile Tor was setup on a laptop connected to the Tor network. The content server hosting the files was placed at Karlstadt University. Experiments used OnionCoffee, which is a Java project developed under the PRIME project.

Panchenko *et al.* [99] performed experiments using a Pentium Dual Core 1GHz CPU with 2GB RAM as a client nodes. Two existing Tor implementations (default Tor and OnionCoffee) were used on the client nodes. The Internet connection had a 10Gbps bandwidth while the local backbone was 100Gbps. Actual Tor relays were used to analyze the performance.

Pries *et al.* [128] performed experiments by downloading a 458kB file from a school web server. Command line utility *wget* was used as the downloading tool. *wget's* http-proxy and ftp-proxy were configured to download all files through *Privoxy* from the server. Tor release `0.1.1.26` was configured on the exit and entry nodes.

Wagner *et al.* [59] implemented a novel architecture using Tor. Three machines were setup running Tor exit

Table 9: Experimental setups used in different research works.

Research	Servers	Relays	Clients	Service	Tor implementation
Private Setup					
Overlier and Syverson [46]	2		1	Hidden service	default Tor
Andersson and Panchenko [146]	1		1	Mobile Tor	Onion Coffea
Panchenko *et al.* [99]			1	Download Service	Def. Tor + Onion Coffea
Pries *et al.* [128]	1		1	Download Service	Privoxy
Wagner *et al.* [59]	2	1	1	Log processing	WebProxy
Chan-Tin *et al.* [57]	1		2	Traffic Analysis	Def. Tor
Pries *et al.* [20]	1	2	1	TCP data coll.	Tor mod.
Herzberg *et al.* [84]			1	Web page download	Def. Tor
Bauer *et al.* [108]		6+	1+	Path compromise	Def. Tor
Song *et al.* [73]		6	1	Traffic Analysis	Tor mod.
Dhungel and Steiner [125]	1	2	1	Traffic Analysis	Def. Tor
Gros *et al.* [77]			2+	Traffic Analysis	Honeywall
Wang *et al.* [58]		2		Traffic Analysis	Privoxy
Zhang *et al.* [48]	1	3	1	Hidden Service	Polipo
Loesing *et al.* [126]		1	1+	Access Attempt	Def. Tor
Chen and Pasquale [95]	1		10	Download	Def. Tor
Panchenko *et al.* [97]	1+		1+	Traffic Analysis	Def. Tor
Houmansadr *et al.* [53]	4		3	Traffic Analysis	–
Li *et al.* [98]	1		1	Download Analysis	Def. Tor
Chakravarti *et al.* [54]	1		2	Download Analysis	Def. Tor
Mulazzani *et al.* [117]			1+	Traffic Analysis	Tor Status
Chaabane *et al.* [131]	1+	6	1+	Traffic Analysis	Def. Tor
Bai *et al.* [50]	2		6	Traffic Analysis	Def. Tor
Barker *et al.* [51]	3	15	1+	Traffic Analysis	Def. Tor
Marks *et al.* [81]	3		3	Download Analysis	
Jin and Wang [75]	1		1	Traffic Analysis	Tor mod.
Tang and Goldberg [104]	1	1	1	Download Analysis	Def. Tor
Alsabah *et al.* [52]			1 (3 Apps)	Traffic Analysis	Def. Tor
Moghaddam *et al.* [139]			2+	Traffic Analysis	SkypeMorph
Weinberg *et al.* [140]	1		1	Download Analysis	StegoTorus
Evans *et al.* [63]			1	Traffic Analysis	Def. Tor
Wang *et al.* [105]			1	Traffic Analysis	Def./Mod. Tor
Ehlert [127]			1	Traffic Analysis	Def. Tor
Barbera *at al.* [21]		2	4	Traffic Analysis	Def. Tor
Winter and Lindskog [55]		2	2+	Traffic Analysis	Tor mod.
Edmundson *et al.* [115]			1	Download Analysis	Def. Tor
Huber *et al.* [118]		1		Traffic Analysis	Def. Tor
Blond *et al.* [66]		6	1+	Traffic Analysis	Def. Tor
Lenhard *et al.* [133]			1+	Hidden Service	Def. Tor
McCoy *et al.* [119]		3		Traffic Analysis	Tor mod.
Chakravarty *et al.* [68]			1	Traffic Analysis	Def. Tor
Snader and Borisov [106]	1		1	Traffic Analysis	Tunable Tor + Vanilla
Gilad and Herzberg [76]	1		1	Traffic Analysis	Def. Tor
Loesing *et al.* [120]			1+	Traffic Analysis	Def. Tor
Chen *et al.* [121]		3+ (VMs)	2+ (VMs)	Traffic Analysis	Def. Tor
Panchenko *et al.* [74]			1+	Traffic Analysis	Def. Tor
Wang and Goldberg [22]			200 cores	Traffic Analysis	Def. Tor
PlanetLab Setup					
Akhoondi *et al.* [94]			50	Traffic Analysis	LASTor
Murdoch and Danezis [70]	1		2	Traffic Analysis	Tor Mod.
Bauer *et al.* [64]	6	2-6	40-90	40 node network	
	6	3-6	60-90	60 node network	
Cloud Setup (Amazon EC2)					
Sulaiman and Zhioua [56]	1			Traffic Analysis	Def./Mod Tor
Karaoglu *et al.* [96]	1		4	Traffic Analysis	Def. Tor
Biryukov *et al.* [49]			50	Hidden Services	Def. Tor

node, BIND (DNS server with tcpdump), and Apache webserver, respectively. All machines were synchronised by NTP. Connected to Tor network, WebProxy was implemented in Perl. `iptables` was used to re-route traffic from Tor exit node to Perl proxy server. All processing of web server logs and proxy logs was performed using Perl, sqlite and modified tcpick.

Chan-Tin *et al.* [57] setup a limited network for probing Tor network using client, burst server and probe machines. Entry and middle routers were chosen randomly while exit node was forced by choice. Four Tor relays were probed for the experiment and data of probes was collected after every 5secs. Five connections were setup by the client using multi-threading.

Pries *et al.* [20] setup client, server, entry malicious router and exit malicious router by setting up four devices. A TCP client application was built which sent and received TCP data. Test server used port 41 and received and displayed data on the screen. The client used `tsocks` to transport its TCP stream through onion proxy. The Tor configuration file was configured to select designated Tor entry and exit routers.

Herzberg *et al.* [84] implemented their proposed camouflaged browsing design over a test machine with an ADSL connection to the Internet with 1,269kBps downlink and 103 kBps uplink bandwidth. Four different URLs were tested with 100 measurements and access time for browsers was recorded. `wget` was used to download web pages.

Bauer *et al.* [108] built an extensive experimental setup by establishing circuits for different kinds of applications with a number of malicious routers. The simulator generated 10,000 circuits with 6 to 106 malicious routers. The path compromise rate for different applications was estimated by the selection of malicious routers.

Song *et al.* [73] used an *Au3* script to capture Tor traffic. Six nodes located at distinct places (India, Romania, Luxemburg, New Zealand, Chile, and Russia) were deployed as exit nodes. Onion proxy running on a local PC was configured to use the deployed exit nodes. Traffic of all routers was captured to analysis.

Dhungel and Steiner [125] measured delay of Tor network by setting up two relays instead of three. Client, exit router and destination server were fixed while the entry router was selected from the list of available routers. To cope with varying network characteristics, the experiment was repeated for eight months with each duration of 40 minutes. All 1,426 available routers were pinged five times for measurements.

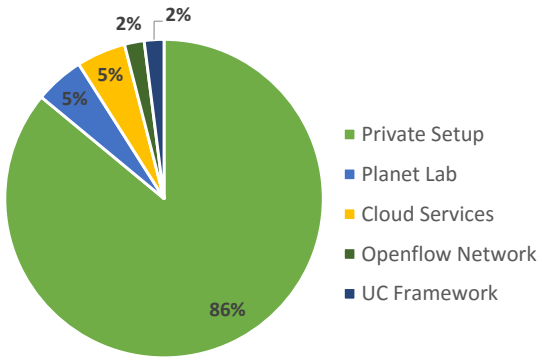

Figure 10: Classification of platforms used in experimental Tor research.

Gros *et al.* [77] performed experiments by using the proposed Honeywall mechanism. All vulnerable clients using Tor were placed on one side of the Honeywall and Internet cloud was present on the other side of the Honeywall. All Tor clients had distinct private addresses while Honeywall had a single public IP address.

Wang *et al.* [58] conducted experiments in a partially controlled environment. The OP code was modified to use the designated entry and exit Tor routers. Entry and exit Tor nodes were configured to record the data relayed through them. Internet Explorer was used at the Tor client through Privoxy. The middle Tor router was selected through Tor router selection algorithms.

Zhang *et al.* [48] used Mozilla Firefox on Fedora 11 to access the hidden service using *Polipo*. The hidden server was configured to use the bridge whose traffic was being logged continuously. Clients and bridges were configured to record the circuit ID, command, stream ID and arrival time.

Loesing *et al.* [126] conducted experiments on Tor by configuring the Tor client to use fixed first / entry relay, which was being monitored continuously. Second and third Tor relays were chosen randomly by Tor's router selection algorithm. A single access attempt was performed by creating new Tor clients after every five minutes over a 72 hour duration.

Chen and Pasquale [95] analyzed the throughput by downloading a 100kB file through nearly 100 unique paths with 10 times repeated downloads over each path. 10 Tor clients were configured over PlanetLab testbed distributed around the globe. A file server containing 100kB file was hosted in the US using `thttpd`. `cURL` was used to conduct downloads. Python was used to write measurement scripts using the `TorCtl` library for Tor control port. Tor circuits were configured to be replaced after every 30 minutes instead of 10 minutes so that no middle replacement takes place.

Panchenko *et al.* [97] performed experiments in the current Tor network with the estimation of delay and throughput. In first experiment, onion routers are fixed but links connecting the circuit are variable. 2,000 sets were built with random onion routers. In the second experiment, ICMP Ping was used to measure the delay between sending a *SYN* and receiving a *SYN-ACK* packet. Each ping was iterated 20 times to calculate the mean value.

Houmansadr *et al.* [53] conducted a deep experimental investigation on the Tor network. Application layer softwares (Skype, CensorSpoofer) were executed in VirtualBox virtual machines (VMs) on a Funtoo Linux machine. Various VMs were connected through virtual distributed

Ethernet (VDE). Authors built their own plugin for VDE which could drop packets at variable rates and also modify packet contents. Various VDE switches were connected to the central switch which provides DHCP connectivity to the Internet.

Li *et al.* [98] tested their proposed tunable mechanism of Tor (TMT) over the real Tor network. Two virtual private servers (VPSes), acting as client and server, were configured on *Linode*. The client was configured with the TMT enhancement while the server hosted a web page. Time to load the file, number of attempts and number of failure attempts were measured to estimate the performance of TMT.

Chakravarti *et al.* [54] setup their own client, server and probing host machine at three distinct locations inside US. 100MB file was placed at the web server which gave sufficient downloading time to the client. Linux traffic controller was used to shape the client-server bandwidth. 26 distinct Tor circuits were created and probed through different locations and compromised links were detected.

Mulazzani *et al.* [117] collect data by using *TorStatus* and updating its script *tns-update.pl* and *network-history.php*. *RRDtool* was used to store the values in a round robin database (RRD). The collected dataset was used for basic network monitoring.

Chaabane *et al.* [131] conduct a deep traffic analysis of Tor using HTTP and Bit Torrent protocols. The authors created and monitored six Tor relay nodes (placed in US, Germany, France, Japan, Taiwan) advertising 100kB available bandwidth for 23 days. On average 20GB of data is provided by each server on every day. Data was collected at entry and exit relays.

Bai *et al.* [50] setup eight PCs with one PC running Tor and one PC running java anonymous proxy (JAP). Dummy traffic was generated from the other six PCs. Traffic was captured through `ethereal`. *Winsock Packet Editor* was used to record packets generated by a specific application. Duration of the test was about 120mins with five repetitions.

Barker *et al.* [51] collected Tor network traces by developing a complete Tor setup. Firefox running on Ubuntu was used on all machines. Using the Selenium browser testing framework, 170 simulations were executed by accessing 30 websites. Three directory servers with 15 relays were configured to be used for experiments. Regular HTTPs traffic and HTTP and HTTPs traffic through private Tor network were collected.

Marks *et al.* [81] conducted a simple experiment using three PCs running Linux kernel (`2.6.26` and CUBIC TCP). All three machines were connected via an Ethernet switch. All Ethernet interfaces were configured to be 10Mbps full-duplex links. The first and last two devices setup TCP connections. First device sent data to the second for 250secs while the second retransmitted after 50secs delay for a duration of 250secs.

Jin and Wang [75] conducted extensive experiments by monitoring both anonymous traffic and Tor traffic during two experiments. In the first experiment, an Apache webserver on a Dell PC using Redhat Enterprise 4 Linux was configured. A watermark encoder was installed on the Apache proxy. A Dell Precision 390 was configured as a NAT router to route traffic between client and the anonymous server. In the second experiment, an SSH server and watermark encoder were installed on one machine acting as server. SSH client and watermark decoder were installed on another machine. Three random characters were sent every second from one machine to the other through Tor. Entry and exit relays were fixed.

Tang and Goldberg [104] setup their own node (acting as the middle node). Entry and exit nodes were selected from the Tor nodes of the directory server. Authors avoided the use of PlanetLab testbed because majority nodes were providing only 100KB/s. *Webfetch* was used to download the target file (87KB) from author's web server. Connecting circuits and load was varied to verify the proposed path selection strategy.

Alsabah *et al.* [52] performed real world experiments by collecting offline data of 200 circuits from three distinct application traces. All three applications (BitTorrent client, web browsing client and stream client) were setup on the same machine which was configured to use a specified Tor node as the entry node. All 200 circuits included browsing (122), BitTorrent (49) and streaming circuits (28). All applications collected 24 hours of data over a 6 week period with periodic intervals.

Moghaddam *et al.* [139] implemented their proposed SkypeMorph technique on Linux using C and C++ with boost libraries. Authors collected traces of Skype data set for modeling using multiple machines. The proposed SkypeMorph scheme was tested by downloading multiple files with and without it.

Weinberg *et al.* [140] implemented the proposed scheme *StegoTorus* by deploying an experimental setup. The client was a desktop PC in California with DSL link to the Internet (downstream 5.6Mb/s, upstream 0.7Mb/s) and the virtual host was situated in New Jersey inside a commercial data center. 1MB files were downloaded over several trials to test the performance.

Evans *et al.* [63] performed experiments on the real Tor network for their proposed congestion attack. The victim user (to be breached) was using Javascript on her browser. Entry node was fixed but the other two Tor relay nodes were selected at random (by Tor's router selection algorithm).

Wang *et al.* [105] measured the network delays during congestion by collecting delay readings of all Tor routers for 72 hours in August 2011. At the next stage, authors collected RTT measurements of the modified and unmodified Tor client to setup 255 circuits. In all experiments, client machines were modified to incorporate the proposed algorithm and measure the delay.

Ehlert [127] measured the performance of I2P and Tor network. For I2P network, experimental setup consisted of two machines, acting as dedicated outproxy and client. 500

most visited websites were used for downloading webpages. For Tor, a client machine was connected to the Tor network and performance parameters were measured similar to I2P proxy.

Barbera *at al.* [21] conducted controlled experiments by setting up 100Mb/s network connected to four hosts (possessing 2.66GHz Core 2 Duo CPUs). For real time network experiments, the authors used their two OR nodes, acting as Tor relays. CellFlood attacks were performed on these routers and performance of attack and mitigation scheme was analyzed.

Winter and Lindskog [55] deployed one relay in Russia and two bridges in Singapore and Sweden. Multiple clients were present in China for connection setup to Tor through designated bridges and relays. In Singapore, a Tor relay was hosted in an Amazon EC2 cloud. Bridge and relay in Sweden and Russia were hosted by an institution and data center, respectively. For vantage points in China, 32 SOCKS proxies and a VPS running Linux was used.

Edmundson *et al.* [115] analyzed the security of Safeplug and Tor by conducting separate experiments for both applications. Authors measured the latency of the system, and investigated the effect of cookies and third party trackers over both applications.

Huber *et al.* [118] deployed a Tor exit node which logs the HTTP requests. Nine million HTTP requests were recorded in several weeks. All requests were analyzed for available patterns and statistics were presented in the research.

Blond *et al.* [66] conducted experiments by deploying Tor exit nodes. Authors instrumented and monitored six Tor nodes for a period of three weeks. One exit node was configured to accept TCP connection for Bit torrent, in order to perform the hijacking attack.

Lenhard *et al.* [133] ran Tor processes on their devices connected to the Tor network. The hidden services were accessed through low bandwidth access network edge. A modem provided a data rate of 56kb/s downstream and 44kb/s upstream. For EDGE, data rate was around 230kb/s. The broadband network provided 100Mb/s.

McCoy *et al.* [119] setup their router connected to 1Gb/s network link with a rank of top 5% Tor routers and flagged as *Running*. At most 20bytes were logged to avoid information breaching laws. Setup was configured for both experiments separately covering (1) exit router and (2) non-exit router. Entrance and middle router traffic was logged for 15 days comprising of time stamp, previous hop's IP, TCP port, next hop's IP and circuit identifier. For exit traffic logging, `tcpdump` was used over the router which relayed 709GB of traffic and only the first 150bytes of packet were logged. `Ethereal` was used for protocol analysis.

Chakravarty *et al.* [68] transmitted decoy traffic over a custom client supporting IMAP and SMTP protocols. The client was implemented using Perl and service protocol emulation was provided by *Net::IMAPClient* and *Net::SMTP*. The client hosted on Intel Xeon CPU running Ubuntu Server Linux `v8.04`.

Snader and Borisov [106] performed experiments on Tor by downloading 1MB files over HTTP connections through various exit routers. All other entities including guard routers, client and web server remain fixed for the entire duration of the experiment. $20,000$ and $40,000$ trials were performed for tunable Tor and standard Tor respectively spanning a duration of two months.

Gilad and Herzberg [76] conducted an empirical investigation for the performance of proposed attacks in the Tor network. Indirect rate reduction attack was evaluated by experiments in the live network. For experiments, a Linux machine ran an Apache web server. Data at the rate of 0.5KBps was transmitted.

Loesing *et al.* [120] collected Tor statistics by following the legal requirements, user privacy, ethical approvals, informed consent and community acceptance. Authors collected data from the Tor network and evaluated the port numbers and country of origin of the obtained IP addresses.

Chen *et al.* [121] developed ORPay which uses out-of-band communication for payment primitives and control messages. The "bank" was built using C language and OpenSSL for encryption. Authors performed controlled experiments consisting of a set of interconnected PCs running directory servers and Tor routers on VMs. Inter-client bandwidth was $500 - 600$KB/s with $1 - 2$ms average latency and 0.5ms for inter-VMs on the same machine. One micropayment was made for every 20 packets.

Panchenko *et al.* [74] using standard PCs for fetching websites using Firefox with disabled active components (Java, Flash etc.) and Chickenfoot used as the default plugin. The closed-world dataset was collected from previous studies, to obtain labeled ground truth dataset.

Wang and Goldberg [22] performed experiments on SHARCNET, a parallel computing cluster. Upto 200 cores were used for computation of SVM kernel matrix. `torrc` was configured to close the circuits manually instead of fixed 10mins duration and fixed entry guard selection was disabled. iMacros and Tor controller was used to automate site accesses. For closed world circuits, fingerprinting was performed on 100 sites with 40 instances each and using 10-fold cross validation. For open-world experiments, Alexa's top $1,000$ sites list was used.

4.1.2. PlanetLab Experiments

Akhoondi *et al.* [94] performed experiments in the real Tor network by modifying the Tor Client with their proposed *LASTor* protocol. `LASTor` is a Java application controlling the Tor client through *Control Port*. 50 *PlanetLab* nodes running `LASTor` were used as Tor clients to access top 200 websites. Both latency and anonymization were tested by collecting the traces of data set at the client nodes.

Murdoch and Danezis [70] performed experiments on the Tor network by setting up a probe PC. A modified version of Tor was used in the probe PC to choose routes of

length one. A TCP client was also established at the node which connects to the SOCKS interface of Tor using *socat*. Original Tor relays were used with a corrupt destination Tor server recording the traffic traces. The probe server ran at the University of Cambridge Computer Laboratory while victim and corrupt server were run on PlanetLab nodes. Data from 13 probing Tor nodes was collected and analyzed in *GNU R*.

Bauer *et al.* [64] performed experiments over PlanetLab testbed by setting up two independent node networks comprising of 40 and 60 nodes, respectively. Two and six malicious nodes were added in the 40 node network while three and six malicious nodes were added in the 60 node network. Traffic was generated by six machines running 60 and 90 clients (requesting files of less than 10MB size using HTTP protocol) in the 40 and 60 node network, respectively. To avoid flooding of network requests, clients sleep in the $0 - 60$sec interval for random periods after every random number of web requests.

4.1.3. Cloud Services

Sulaiman and Zhioua [56] performed extensive experiments using Amazon *EC2* cloud services. An Apache web server was used to host a simple web page. `socket.io` with `node.js.Socket.io` was installed which supported WebSocket to help users' browsers in using OP and using unpopular ports. For path selection, simulations were also performed for entrance router selection algorithm and non-entrance router selection algorithm. Several experiments were conducted on a number of unpopular ports with $1,500$ circuits established per experiment and compromised links were detected.

Karaoglu *et al.* [96] implemented a unidirectional scenario of client uploading a file to a web server. A client established multiple socket connections for multipath transmissions. A 1.5MB file was uploaded through clients. To incorporate geo-diversity, client softwares were installed in the US and at Amazon EC2 sites in Singapore, Ireland and North Virginia. A web server, placed at the Emulab Utah facility, listened on multiple ports.

Biryukov *et al.* [49] performed deanonymization by spending less than 100 USD on Amazon EC2 cloud. 50 Amazon EC2 instances were generated which captured $59,130$ publication requests. Data from 120 running hidden services from the Tor network was collected. Collected data was used to identify the vulnerability of Tor hidden services.

4.1.4. OpenFlow Enabled Network

Mendonca *et al.* [85] used OpenFlow implementation for their proposed AnonyFlow scheme. An experimental testbed used Linux to connect the two subnetworks. Each subnetwork was connected to two OpenFlow enabled switches and two Net FPGA based switches. All these OpenFlow switches were governed by a NOX controller. `iperf` was used at the two client hosts with each running for nearly 25secs.

4.1.5. Universal Composability framework

Backes *et al.* [80] provided security enhancements to the currently used Tor network. New algorithms were provided and setup in the universal composability (UC) framework.

4.2. Tor Simulations

Tor simulations have been performed by (1) developing custom simulator (2) using ExperimenTor, (3) employing Shadow simulator, and (4) using ModelNet, as shown in Figure 11. Figure 11 shows that 75% of the researches (comprising of simulations) used in this study developed their custom simulator. Only 13% used ExperimenTor. 8% and 4% of the studies used Shadow simulator and Modelnet, respectively.

4.2.1. Custom Simulator

Tschorsch and Scheuermann [130] conducted simulations on *ns-3* to implement Tor network with and without *N23* modifications. To replicate the onion routers environment, all onion routers were connected to a central node. Access links of all onion routers had an 80ms delay and 100Mbps bandwidth. Sending hosts generate data at a rate of 400kbps and Tor nodes had a maximum bandwidth limit of 600kbps.

Doswell *et al.* [145] used the generic network simulator OMNET++ to simulate mobile Tor. Wireless access points were placed 75m apart and results were estimated using linear mobility. Average throughput (kbps) was selected as the performance metric. A 300kB webpage was downloaded after every 2secs over the time-frame of 600secs. An artificial latency was also introduced to incorporate congestion.

Edman and Syverson [65] implemented the multi-thread path selection algorithm in C. Relationships between different ASs were borrowed from predecessor studies. RIBs collected by University of Oregon's RouteViews project were used.

Ngan *et al.* [136] built a discrete event simulator, in Java, for the Tor network. 64-bit AMD Opteron 252 dual core processors were used with 4GB RAM and operating

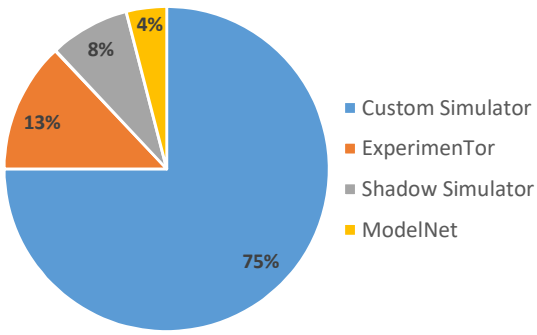

Figure 11: Classification of simulations on Tor.

on Sun's JVM and RedHat Enterprise Linux. Tor network with 150 relays was simulated and all cells from every client were simulated at every hop. Link latency was 100ms and link capacity was 500KB/s. All scenarios were tested comprising of Tor's original design, proposed design and a hybrid mechanism.

Benmeziane and Badache [60] built their own simulator which incorporates public communications, DNS requests, and anonymous communications by Tor. The authors used 500 senders using 100 Tor relay nodes with 10 executions per sender. Authors increased the number of recipients to 200. Much of the simulator details were skipped.

Li *et al.* [103] developed their own discrete event simulator for Tor network. Key data structures and algorithms of Tor were used to simulate several thousand nodes. However, authors did not perform encryption, decryption and data transmission to avoid complexity. Moreover, simulations were driven by initialization and termination events. For a closer look, realistic values of bandwidth and uptime were obtained from the Tor metrics portal. Effective bandwidth of relays was set to 155kBps with a 750 standard deviation. 3000 relays with millions of clients were used for simulations.

Snader and Borisov [102] developed a custom flow-level simulator for the Tor network. Using the Tor metric portal, bandwidth of actual Tor relays was used to simulate a $1,000$ node network. $10,000$ flows were simulated for each time unit of the simulator. Fair queueing was used for flow scheduling.

Jansen *et al.* [135] built a discrete event simulator for the Tor network comprising of $19,400$ web clients, 300 Tor relays, $2,000$ servers and 600 file sharing nodes. For file sharing, web traffic comprised of 12Mbps downstream with 1.3Mbps upstream bandwidth.

Johnson *et al.* [69] built the *TorPS* simulator for selection of Tor paths. Simulations were carried out for six months with an adversary model containing one guard relay and one exit relay having 83MBps and 16.7MBps. For analysis of client behavior, $50,000$ Monte Carlo simulations were carried out spanning a period of three months.

Nowlan *et al.* [82] developed a setup for a small virtual Tor network to estimate the performance of the proposed modification. Tor network comprises of three directory authorities, three relay servers and single onion proxy. The link delay had a mean of 50ms with a 5% path loss for the second onion router.

Jansen *et al.* [122] performed extensive simulations for their proposed model on both small and large-scale networks. Loss rate and latency have been borrowed from Ookla and iPlane estimation services. For small scale network, 50 relays and 500 clients have been configured using 50 HTTP file servers. For large scale networks, 100 relays and $1,000$ clients are linked with 100 HTTP servers. Files of 320KB and 5MB are downloaded for performance analysis.

Danner *et al.* [83] carried out extensive simulations of their proposed analytical model. However, authors do not focus on experiments or discrete event simulations.

Wang *et al.* [137] analyzed the performance of their proposed bridge distribution mechanism on an event-based simulator. Aggressive blocking, conservative blocking and event-driven based blocking of bridges were tested. The authors also developed an analytical model for performance prediction.

Jansen and Hopper [123] developed a discrete event simulator to replicate the real-world Tor network in software running on a single machine. Performance was validated against 402 node PlanetLab network. Through HTTP client and server plugins, data was transferred through Shadow for verifications of simulations.

Smits *et al.* [138] developed an open source implementation of the proposed mechanism. Implementation is based on Linux version *2.6.4*. Bridge distribution authorities needed to be reconfigured for distribution of keys.

Elahi *et al.* [107] simulated Tor entry guard selection and rotation mechanism on multicore servers with each simulation run comprising of $80,000$ users. The entry guard data was collected from real Tor network spanning a duration of eight months.

Zhang *et al.* [72] developed a complete Tor setup containing client, server and three onion routers. A probe server and user nodes were deployed in different network segments. Tor code in the nodes was configured to use the designated three relay nodes. Data from the probe server was sent in bursts after every 0.2secs while corrupt server sends data after every $10 - 15$secs.

4.2.2. ExperimenTor

Bauer *et al.* [124] built a toolkit for emulation of Tor network named by *ExperimenTor*. The *ModelNet* network emulation platform has been used as the baseline approach. Scalability is one of the issues in ExperimenTor, owing to high resource consumption for large number of nodes.

Wacek *et al.* [109] performed network experiments over ExperimenTor for a variety of network topologies. Authors also performed simulations on their simulator which modeled a $1,524$ relay network.

Gopal and Heninger [141] used ExperimenTor framework for simulations of their proposed Torchestra approach. ExperimenTor was setup on two physical machines working as edge node and emulator. For performance analysis, small and large files were downloaded starting from 300KB. In the following stage, web and SSH traffic were simulated.

4.2.3. Shadow Simulator

Geddes *et al.* [67] used the Shadow simulator with real Tor code on a simulated network. The simulated network consisted of 160 exit relays, 240 non-exit relays, 2375 web clients, 125 bulk clients, 150 small and medium `Torperf` clients and 400 HTTP servers. Experiments consisted of downloads of a 320KB file from random servers after random delays ($1-60$secs). Bulk clients downloaded 5MB file

without any wait time. For `TorPerf` clients, 50KB, 1MB and 5MB files were downloaded after every ten minutes.

Jansen *et al.* [123] performed simulations over Shadow with a setup of 200 HTTP servers, 950 Tor web clients, 50 Tor bulk clients and 50 Tor relays. Bulk clients downloaded 5MB file while web clients downloaded 5KB page. Latency of network was borrowed from the latency of PlanetLab nodes. Performance was estimated by varying the load from 25 (light) to 50 (medium) to high (100) bulk users.

4.2.4. ModelNet

AlSabah *et al.* [142] used the `ModelNet` network emulation platform along with practical traffic models for performance evaluations. For small-scale experiments, 200 downloads are made of the 300KB and 5MB files by two clients in two separate experiments. For large scale experiment, 20 Tor routers are deployed with real Tor networks' bandwidth. Each link has 80ms RTT delay. Ten clients download 1–5MB file and 190 clients download 100–500KB file.

4.3. Tor's Analysis

A number of studies limited their research works to the analysis of current Tor network instead of simulations and experiments of Tor. Analysis occurs in the subfields of usability of Tor, path selection mechanism, empirical analysis and development of theoretical model. In below lines, we present the individual studies comprising of Tor's analysis.

4.3.1. Analytical Model

Several studies develop analytical model for analysis of Tor network. These studies are presented in following lines.

Anti-misbehaviour Policy Analysis: Liu and Wang [111] proposed anti-misbehavior policies and analyzed it with the original Tor architecture. No simulations or experiments were conducted.

Security Analysis: Goldberg [134] built an analytical model for analyzing the security of Tor's authentication protocol. Authors focused on analytical evaluations rather than simulations or experiments.

Botnet Abuse Analysis: Hopper [132] analyzed the various possibilities for avoiding botnet abuse in the Tor network. Majority schemes were discussed only, and a few schemes were tested to verify the performance. Several schemes were analyzed analytically.

Anonymity Model: Xin *et al.* [79] developed a theoretical model to increase the anonymity of the Tor network. They aimed to implement the proposed system on PlanetLab testbed, in future.

4.3.2. Empirical Analysis

A number of studies performed empirical analysis of Tor network without performing simulations or experiments. In following lines, we present the findings of these studies.

Statistical Analysis: Wang *et al.* [129] performed empirical analysis and used data available from "Tor Metric Portal" for analysis. There were no simulations or experiments performed in the research. In another study, Elices *et al.* [47] analyzed their attack on Tor using empirical analysis. Access logs from seven web servers were obtained to analyze user request pattern. Moreover, Abbott *et al.* [62] conducted a statistical evaluation by measuring the probabilities of breaching Tor using the proposed scheme.

Robustness Analysis: Barthe *et al.* [116] analyzed the robustness of the Tor network and proposed enhancements in the current network. Cryptographic enhancements were evaluated without any simulation or experimental validations.

Path Selection Protocol: Liu and Wang [101] presented an improved circuit building protocol with no simulations or experiments. Proposed algorithm was analyzed considering various aspects. In another study, Liu and Wang [100] presented random walk based algorithm for Tor circuit construction. Anonymity and performance were the key metrics evaluated in their study. However, the scope of this study did not cover simulations or experimental evaluations.

4.3.3. Usability Analysis

Usability analysis of Tor has not been carried out by a lot of studies. However, some studies referring to usability analysis are summarized in below lines.

Clark *et al.* [114] conducted a usability analysis by installing various components of Tor including Vidalia, Privoxy, Torbutton and Foxyproxy on a standard machine. In another study, Abou-Tair *et al.* [113] presented the usability analysis of the various anonymous service applications including Tor. Various anonymity tools were installed on a machine and usability, ease of installation and use was analyzed.

5. Discussion

In this section, we present the discussion and our findings of Tor network. In the first part, we present the performance metrics used to evaluate the Tor network in different research works. In the second part, we present our findings of Tor research works referred in this study. In the last part, we show our findings for open research areas in the field of Tor network which may be used for future research works.

5.1. Tor Performance Metrics

Analyzing the performance metrics is a crucial task for future research, analysis, simulations and experiments

in the Tor network. Table 10 presents the performance metrics of Tor used in various studies. No clear patterns were observed, so, authors described the metrics used in individual studies. A brief overview of the table shows that throughput (bandwidth) and latency are the most frequently used metrics. However, every research formalized its own performance metric based upon the requirement of the experiment.

5.2. Survey Findings

In this section, we summarize our findings for the onion router by comparing all studies with a deep focus over the key concepts and ideas used in different research works. We divide our research evaluations in three subcategories considering (1) research areas, (2) research platforms, and (3) performance metrics.

5.2.1. Research Areas

The majority of Tor research (nearly 55%) covering anonymity is focused over deanonymization of Tor network. Around 20% studies are related to the path selection mechanism. Only 25% research studies are on performance analysis and improvement mechanism of Tor network. According to Dingledine (the co-founder of Tor project), majority research works focus their attention on the breaching Tor.

Deanonymization:

In the deanonymization track, 35% of the studies design deanonymization attacks for Tor while 21% deanonymize Tor using traffic analysis. 16% focus on improvements to bypass deanonymization while 14% study fingerprinting mechanisms to identify Tor traffic on the Internet. Only 9% identify hidden services while 2% focus on anonymity mechanisms without using Tor.

All deanonymization related Tor studies have exploited its inherent weaknesses. Compromised relays are the most exploited weaknesses followed by traffic interception and protocol messages. Very few studies focus on the compromised autonomous systems, browsers, servers, decoy traffic, and flag cheating.

Path Selection:

In the path selection track, 87% of the studies focus on the design of new path selection algorithms and 13% research works analyze currently developed algorithms.

Our analysis shows that anonymity and performance (bandwidth and latency) are the most important parameters used in the design and analysis of path selection algorithms. Relays have been incorporated in the design of path selection algorithms covering both location and capacity of relays. Other parameters include autonomous systems, hops, multi-path mechanism and load.

Performance Analysis and Architectural Improvements:

In the performance analysis and improvement track, 32% of the research works cover analysis and 27% of studies focus on performance improvement mechanisms. 29% of the studies provide general analysis of Tor covering usability and sociability issues. 10% of the research works focus on modeling of the Tor network while 2% address client mobility.

Analysis of various research studies show that performance (latency and bandwidth), relay selection, and anonymity are the most used parameters. Other studies also pay attention to queues, QoS, protocol messages and traffic shaping.

5.2.2. Research Platforms

An interesting feature revealed in analysis is the fact that 60% of the research works were conducted by performing real-world experiments on the Tor network. Although special measures were taken to protect the identity of users but majority research works failed to analyze legal or ethical requirements of capturing user data and performing experiments by developing attacks in real network. Only 27% of studies developed their own simulator and 13% conducted analysis without experiments or simulations.

Experiments:

Our survey shows that 86% of the research works developed their own testbed for experiments. The majority of studies deployed 1-2 clients with 1-3 servers for experiments. Research works covering relays used 1-3 relays. However, some research works increased the number of relays by using virtual machines and PlanetLab. A limited number of studies used cloud-based setups.

Simulations:

Interestingly, 75% of the research works developed their own simulator without any common parameters used for Tor network. ns-3, OMNET, C and Java were used for the development of custom simulators. 13% of the research works used ExperimenTor. Our research shows that ExperimenTor is the most common toolkit used by majority of the research. 8% and 3% of the studies used Shadow simulator and ModelNet, respectively.

5.2.3. Performance Metrics

We considered the performance metrics used in various works. Analysis shows that no hard and fast rule exists for use of performance metrics. Every study developed its own metrics to measure performance, anonymity and QoS. Moreover, no baseline techniques exist for the comparison of results.

5.3. Open Research Areas

Our survey shows that majority of the research works are concentrated in a few domains. However, a number of major challenges exist owing to the peer-to-peer nature of Tor. A number of key areas have also been identified by the Tor team. We identified the following areas which require further research.

1. *Data Estimation:* Estimation of key network statistics is the most critical task in the Tor network because it is a peer-to-peer network. No one can see the entire traffic so it is not possible to estimate the size of Tor network. Some of the statistics requiring attention are as follows:

Table 10: Performance metrics used in various research works.

Domain	Performance Metrics	Research Works
Quality of Service of Tor	Throughput; Bandwidth; Packet rate; Bit rate; Goodput	Mendonca *et al.* [85], Zhang *et al.* [48], Pries *et al.* [20], Jin and Wang [75], Marks *et al.* [81], Panchenko *et al.* [99], Chen and Pasquale [95], Karaoglu *et al.* [96], Li *et al.* [103], Panchenko *et al.* [97], Snader and Borisov [102], Houmansadr *et al.*, Pries *et al.* [128], Tschorsch and Scheuermann [130], Andersson and Panchenko [146], Doswell *et al.* [145], Jansen *et al.* [135], Moghaddam *et al.*, Weinberg *et al.* [140], Jansen *et al.* [122], Ehlert [127], Barbera *et al.* [21], Hopper [132], Nowlan *et al.* [82], Ngan *et al.* [136], Panchenko *et al.* [99], Wang *et al.* [129], Jansen *et al.* [135], Tang and Goldberg [104], AlSabah *et al.*, Wack *et al.* [109], Geddes *et al.* [67], AlSabah *et al.* [142], Jansen and Hopper *et al.* [123], Jansen *et al.* [143]
	Latency; Webpage loading time; Round trip time; Download Time; Router latency; Circuit setup duration; Boot strap duration; Time to first byte; Time to last byte; Ping reply delay; Per hop latency; SYN and SYN ACK difference; Delay per cell; Jitter; Inter packets delay distribution	Mendonca *et al.* [85], Overlier and Syverson [46], Loesing *et al.* [126], Chan-Tin *et al.* [57], Herzberg *et al.* [84], Murdoch and Danezis [70], Zhang *et al.* [72], Akhoondi *et al.* [94], Panchenko *et al.* [99], Li *et al.* [98], Dhungel and Steiner [125], Andersson and Panchenko [146], Doswell *et al.* [145], AlSabah *et al.*, Moghaddam *et al.*, Weinberg *et al.* [140], Evans *et al.* [63], Wang *et al.* [105], Jansen *et al.* [122], Ehlert [127], Hopper [132], Winter and Lindskog [55], Edmundson *et al.* [115], Ngan *et al.* [136], Panchenko *et al.* [99], Lenhard *et al.* [133], Wack *et al.* [109], Geddes *et al.* [67], AlSabah *et al.* [142], Jansen and Hopper *et al.* [123], Snader and Borisov [106], Jansen *et al.* [143], Chen *et al.* [121], Smits *et al.* [138], Gopal and Heninger [141], Panchenko *et al.* [97], Panchenko *et al.* [99]
Performance of Tor's breaching attempts	True Positive; True Negative; False Positive; False Negative; Region of Convergence; Recognition rate; Mis-recognition rate; Accuracy; Recall; Precision; F-measure	Chakravarti *et al.* [54], Barker *et al.* [51], Chan-Tin *et al.* [57], Akhoondi *et al.* [94], Danner *et al.* [83], Gilad and Herzberg [76], Panchenko *et al.* [74], Song *et al.*, Wang and Goldberg [22], Wang and Goldberg [22], Elices *et al.* [47], Bai *et al.* [50], AlSabah *et al.*, Wagner *et al.* [59]
	Timing Attack correlation	Overlier and Syverson [46], Zhang *et al.* [48], Pries *et al.* [20], Wang *et al.* [58], Houmansadr *et al.*, Murdoch and Danezis [70], Song *et al.*, Panchenko *et al.* [99]
	Compromised relays; Compromised circuits; Compromised Streams; Time for first compromised stream; Failure rate; Compromise time; Compromised links; Compromised Tunnels; Detection rate; Compromised Clients; Compromised router bandwidth; Compromise probability; Congestion attack time	Overlier and Syverson [46], Sulaiman and Zhioua [56], Chen and Pasquale [95], Li *et al.* [103], Li *et al.* [98], Bauer *et al.* [64], Johnson *et al.* [69], Evans *et al.* [63], Danner *et al.* [83], Chakravarty *et al.* [68], Snader and Borisov [106], Panchenko *et al.* [74], Elahi *et al.* [107], Abbott *et al.* [62], Bauer *et al.* [108], Wang *et al.* [105]
Analysis of Tor	Packet Sizes; Probability difference plots; Energy plots; Recipient probabilities; Queued messages length; Anonymity vs performance; Router bandwidth; HTTP content distribution; Tor servers; Tor traffic; Generated Paths; Client resource usuage; Tor load per circuit; Node Connection pattern; IP TTL difference; Service, browser, file format usuage; Tor location usuage; Boot strap time; Exit traffic stats; Tor bridges statistics; hidden service descriptor request rate; Botnet decay rate; Tor overhead; Router statistics	Barker *et al.* [51], Loesing *et al.* [126], Benmeziane *et al.*, Jin and Wang [75], Zhang *et al.* [72], Liu and Wang [100], Liu and Wang [111], Dhungel and Steiner [125], Chaabane *et al.* [131], Mulazzani *et al.* [117], Moghaddam *et al.*, Edman and Syverson [65], Barbera *et al.* [21], Hopper [132], Winter and Lindskog [55], Huber *et al.* [118], Blond *et al.* [66], Lenhard *et al.* [133], McCoy *et al.* [119], Wang *et al.* [137], Biryukov *et al.* [49], Loesing *et al.* [120], Chen *et al.* [121], Panchenko *et al.* [74], Elahi *et al.* [107], Marks *et al.* [81]
	Empirical Evaluations: Usability analysis, security model analysis, general discussion, proposed mechanism validation	Clark *et al.* [114], Abou-Tair *et al.* [113], Goldberg [134], Kuhn *et al.*, Barthe *et al.* [116], Gros *et al.* [77]

- Number of clients in the network: Peer-to-peer networks make it impossible to estimate the total traffic statistics because no user can see the complete traffic.
- Capabilities of relays: There is limited information available about the relays which are the most crucial parameters in path selection. Incorporation of relay capabilities into anonymity of Tor and performance model is a key research area as done in a number of studies.
- Performance of the network: Estimation of network performance at any given time is a crucial task. Owing to the P2P nature, only health of relays is known to the Tor administration. *How is the network performing at any given instant?* is still a crucial task.
- Number of clients connecting via bridges: Tor authorities provide secret relay addresses to clients who can't access Tor due to blockage of relays in their location. However, very little is known about the quantity of clients connecting through bridges and their traffic statistics.
- Exit network traffic: Significant research is required about the exit network traffic. All clients pass their data through relays and very little is known about the statistics of traffic exiting exit relays.

2. *Analysis:* Deep analysis of the current Tor network is required. Analysis may be based upon an extension of previous research into path length, anonymity, latency, etc. Analysis of the optimal performance parameters is required.
3. *Measurement and Attack tools:* Development of novel attack methodologies to identify the shortcomings of the current Tor network. Tor has no automatic mechanism to identify anomalies and assess the health of the network. Attack tools should be developed which should prevent attacks occurring from compromised relays and servers. Comprised relays are vulnerable to botnet based attacks comprising of DDOS attacks, fingerprinting attacks etc. Despite large amount of research in botnet attacks, it is still open to research which would make Tor a more stable and secure network.
4. *Defenses against Attacks:* Develop novel defense methodologies to counter attacks on the Tor network. Although majority research works have focused on the development of novel attack methodologies, very little is known about viable counter-measures. Our survey shows that relays are mostly vulnerable because they can be deployed by any eavesdropper. Countermeasures against congestion attacks, latency measuring attacks, throughput measuring attacks, etc. can help in the improvement of Tor.

6. Conclusion

This paper deals with the survey, classification, quantification and comparative analysis of various research works covering Tor network. To the author's best knowledge, no other survey/research has performed such a deep and thorough analysis of Tor studies. Our study shows that Tor research areas can be broadly classified into (1) deanonymization, (2) path selection, (3) analysis and performance improvements. More than half studies carried out address 'deanonymization' with major subdivisions into deanonymization 'attacks' and 'traffic analysis' attacks. In the 'path selection' area, more than 85% of the studies have focused on the development of new algorithms. In the 'analysis and performance improvement' area, the majority of studies are a mixed bag, followed by analysis, followed by performance improvement studies. Our analysis of Tor platforms shows that 60% of studies performed experiments while 27% performed simulations. Among experiments, 86% of the studies deployed private testbeds. Among simulations, 75% developed their own simulators. Analysis of parameters (used in various studies) shows that their is no little consistency across various studies. However, a majority of the studies used variations of throughput and latency for performance analysis.

7. Bibliography

[1] Russell Brandom. FBI agents tracked Harvard bomb threats despite Tor. *THE VERGE. Available at http://www.theverge.com/2013/12/18/5224130/fbi-agents-tracked-harvard-bomb-threats-across-tor*, December 18, 2013.

[2] Ilya Khrennikov. Russians Find Ways to Bypass Latest Web Ban. *Bloomberg. Available at http://www.bloomberg.com/news/articles/2016-02-01/russians-turn-to-tor-and-anonymox-to-bypass-web-blocking*, February 2, 2016.

[3] Alex Hern. US defence department funded Carnegie Mellon research to break Tor. *The guardian. Available at https://www.theguardian.com/technology/2016/feb/25/us-defence-department-funding-carnegie-mellon-research-break-tor*, 25 February 2016.

[4] Darien Graham-Smith. Extreme online security measures to protect your digital privacy a guide. *The guardian. Available at https://www.theguardian.com/technology/2016/jul/03/online-security-measures-digital-privacy-guide*, 3 July 2016.

[5] Dave Neal. Mozilla will have to wait to find out how the FBI cracked Tor. *The Inquirer. Available at http://www.theinquirer.net/inquirer/news/2458121/mozilla-wants-to-know-how-the-fbi-cracked-tor*, 18 May 2016.

[6] Mashael AlSabah and Ian Goldberg. Performance and security improvements for tor: A survey. *ACM Computing Surveys (CSUR)*, 49(2):32, 2016.

[7] R. Koch, M. Golling, and G. D. Rodosek. How Anonymous Is the Tor Network? A Long-Term Black-Box Investigation. *Computer*, 49(3):42–49, Mar 2016.

[8] Mashael AlSabah and Ian Goldberg. Performance and Security Improvements for Tor: A Survey. *IACR Cryptology ePrint Archive*, 2015:235, 2015.

[9] Bernd Conrad and Fatemeh Shirazi. A Survey on Tor and I2P. *Proceedings of 9th International Conference on Internet Monitoring and Protection (ICIMP)*, page 22, 2014.

[10] Rolf Jagerman, Wendo Sabee, Laurens Versluis, Martijn de Vos, and Johan Pouwelse. The fifteen year struggle of decentralizing privacy-enhancing technology. *arXiv preprint arXiv:1404.4818*, 2014.

[11] Jian Ren and Jie Wu. Survey on anonymous communications in computer networks. *Computer Communications*, 33(4):420–431, 2010.

[12] Peter C Johnson and Apu Kapadia. From Chaum to Tor and Beyond: A Survey of Anonymous Routing Systems. *Dartmouth College (Technical Report), Available at http://www.cs.dartmouth.edu/ ccpalmer/classes/cs55/Content/Resources/JohnsonKapadiaSurvey.pdf*, 2007.

[13] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing. *Communications of the ACM*, 42(2):39–41, 1999.

[14] TorMETRICS. Relays and bridges in the network. *Available at https://metrics.torproject.org/*, July 13, 2016.

[15] Michael G Reed, Paul F Syverson, and David M Goldschlag. Proxies for anonymous routing. In *Annual Computer Security Applications Conference*, pages 95–104. IEEE, 1996.

[16] David M Goldschlag, Michael G Reed, and Paul F Syverson. Hiding routing information. In *International Workshop on Information Hiding*, pages 137–150. Springer, 1996.

[17] Michael G Reed, Paul F Syverson, and David M Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected areas in Communications*, 16(4):482–494, 1998.

[18] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an analysis of onion routing security. In *Designing Privacy Enhancing Technologies*, pages 96–114. Springer, 2001.

[19] Emmanuel Bresson, Olivier Chevassut, David Pointcheval, and Jean-Jacques Quisquater. Provably authenticated group diffie-hellman key exchange. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 255–264. ACM, 2001.

[20] Ryan Pries, Wei Yu, Xinwen Fu, and Wei Zhao. A new replay attack against anonymous communication networks. In *International Conference on Communications (ICC)*, pages 1578–1582. IEEE, 2008.

[21] Marco Valerio Barbera, Vasileios P Kemerlis, Vasilis Pappas, and Angelos D Keromytis. Cellflood: Attacking Tor onion routers on the cheap. In *European Symposium on Research in Computer Security - ESORICS*, pages 664–681. Springer, 2013.

[22] Tao Wang and Ian Goldberg. Improved website fingerprinting on Tor. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES)*, pages 201–212. ACM, 2013.

[23] Adam Gordon and Steven Hernandez. *The Official (ISC) 2 Guide to the SSCP CBK*. John Wiley & Sons, 2016.

[24] SoftEther VPN Project. University of Tsukuba. *https://www.softether.org*, Last Accessed: July 2016.

[25] JanusVM: An Internet Privacy Appliance. *http://janusvm.peertech.org*, Last Accessed: July 2016.

[26] proXPN. *https://www.proxpn.com*, Last Accessed: July 2016.

[27] USAIP. *https://usaip.eu*, Last Accessed: July 2016.

[28] VPNreactor. *https://www.vpnreactor.com/*, Last Accessed: July 2016.

[29] Xero Networks AG and Steve Topletz. xB Browser. *http://xerobank.com/*, Last Accessed: 2009.

[30] AnchorFree Inc. Hotspot Shield. *http://www.hotspotshield.com/*, Last Accessed: July 2016.

[31] Advanced Onion Router. *https://sourceforge.net/projects/advtor/*, Last Accessed: July 2016.

[32] SecurityKISS. *https://www.securitykiss.com/*, Last Accessed: July 2016.

[33] UltraReach. UltraSurf. *http://ultrasurf.us/*, Last Accessed: July 2016.

[34] CyberGhost S.R.L. CyberGhost VPN. *http://www.cyberghostvpn.com/*, Last Accessed: July 2016.

[35] Dynamic Internet Technology Inc. (DIT). Freegate. *http://dit-inc.us/freegate.html*, Last Accessed: July 2016.

[36] Tails. *https://tails.boum.org/*, Last Accessed: July 2016.

[37] Privatix. *http://www.mandalka.name/privatix/*, Last Accessed: July 2016.

[38] Surfboard Holdings B.V. Ixquick. *https://www.ixquick.com/*, Last Accessed: July 2016.

[39] Inc. DuckDuckGo. DuckDuckGo. *https://duckduckgo.com/*, Last Accessed: July 2016.

[40] Anonymous Email. *https://anonymousemail.me/*, Last Accessed July 2016.

[41] Safe-mail. *http://www.safe-mail.net/*, Last Accessed: July 2016.

[42] Hushmail. *https://www.hushmail.com/*, Last Accessed: July 2016.

[43] 10minutemail. *https://10minutemail.com*, Last Accessed: July 2016.

[44] YOPmail. *http://www.yopmail.com*, Last Accessed: July 2016.

[45] Daniel Arp, Fabian Yamaguchi, and Konrad Rieck. Torben: Deanonymizing Tor communication using web page markers. Technical report, IFI-TB-2014-01, University of Göttingen, 2014.

[46] Lasse Overlier and Paul Syverson. Locating hidden servers. In *Symposium on Security and Privacy*, pages 15–pp. IEEE, 2006.

[47] Juan A Elices, Fernando Perez-Gonzalez, and Carmela Troncoso. Fingerprinting Tor's hidden service log files using a timing channel. In *International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2011.

[48] Lu Zhang, Junzhou Luo, Ming Yang, and Gaofeng He. Application-level attack against Tor's hidden service. In *International Conference on Pervasive Computing and Applications (ICPCA)*, pages 509–516. IEEE, 2011.

[49] Alex Biryukov, Ivan Pustogarov, and R Weinmann. Trawling for Tor hidden services: Detection, measurement, deanonymization. In *Symposium on Security and Privacy*, pages 80–94. IEEE, 2013.

[50] Xuefeng Bai, Yong Zhang, and Xiamu Niu. Traffic identification of Tor and web-mix. In *International Conference on Intelligent Systems Design and Applications (ISDA)*, volume 1, pages 548–551. IEEE, 2008.

[51] John Barker, Peter Hannay, and Patryk Szewczyk. Using Traffic Analysis to Identify the Second Generation Onion Router. In *International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 72–78. IEEE, 2011.

[52] Mashael AlSabah, Kevin Bauer, and Ian Goldberg. Enhancing Tor's performance using real-time traffic classification. In *Proceedings of the conference on Computer and Communications Security*, pages 73–84. ACM, 2012.

[53] Amir Houmansadr, Chad Brubaker, and Vitaly Shmatikov. The parrot is dead: Observing unobservable network communications. In *Symposium on Security and Privacy*, pages 65–79. IEEE, 2013.

[54] Sambuddho Chakravarty, Angelos Stavrou, and Angelos D Keromytis. Identifying proxy nodes in a Tor anonymization circuit. In *International Conference on Signal Image Technology and Internet Based Systems (SITIS)*, pages 633–639. IEEE, 2008.

[55] Philipp Winter and Stefan Lindskog. How the great firewall of China is blocking Tor. *Free and Open Communications on the Internet (FOCI)*, 2012.

[56] Muhammad Aliyu Sulaiman and Sami Zhioua. Attacking Tor through Unpopular Ports. In *International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 33–38. IEEE, 2013.

[57] Eric Chan-Tin, Jiyoung Shin, and Jiangmin Yu. Revisiting Circuit Clogging Attacks on Tor. In *International Conference on Availability, Reliability and Security (ARES)*, pages 131–140. IEEE, 2013.

[58] Xiaogang Wang, Junzhou Luo, Ming Yang, and Zhen Ling. A novel flow multiplication attack against Tor. In *International Conference on Computer Supported Cooperative Work*

in Design (CSCWD), pages 686–691. IEEE, 2009.

[59] Cynthia Wagner, Gerard Wagener, Radu State, Alexandre Dulaunoy, and Thomas Engel. Breaking Tor anonymity with game theory and data mining. *Concurrency and Computation: Practice and Experience*, 24(10):1052–1065, 2012.

[60] Souad Benmeziane and Nadjib Badache. Tor network limits. Technical report, CERIST Digital Library, Availablle at http://dl.cerist.dz/handle/CERIST/317, 2010.

[61] Rob Jansen, Florian Tschorsch, Aaron Johnson, and Björn Scheuermann. The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network. In *Network and Distributed Systems Security Symposium (NDSS), San Diego, CA, USA*, 2014.

[62] Timothy G Abbott, Katherine J Lai, Michael R Lieberman, and Eric C Price. Browser-based attacks on Tor. In *Privacy Enhancing Technologies*, pages 184–199. Springer, 2007.

[63] Nathan S Evans, Roger Dingledine, and Christian Grothoff. A Practical Congestion Attack on Tor Using Long Paths. In *USENIX Security Symposium*, pages 33–50, 2009.

[64] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against Tor. In *Proceedings of the workshop on Privacy in electronic society*, pages 11–20. ACM, 2007.

[65] Matthew Edman and Paul Syverson. AS-awareness in Tor path selection. In *Proceedings of the conference on Computer and Communications Security*, pages 380–389. ACM, 2009.

[66] Stevens Le Blond, Pere Manils, Chaabane Abdelberi, Mohamed Ali Dali Kaafar, Claude Castelluccia, Arnaud Legout, and Walid Dabbous. One bad apple spoils the bunch: exploiting P2P applications to trace and profile Tor users. *arXiv preprint arXiv:1103.1518*, 2011.

[67] John Geddes, Rob Jansen, and Nicholas Hopper. How low can you go: Balancing performance with anonymity in Tor. In *International Symposium on Privacy Enhancing Technologies (PETS)*, pages 164–184. Springer, 2013.

[68] Sambuddho Chakravarty, Georgios Portokalidis, Michalis Polychronakis, and Angelos D Keromytis. Detecting traffic snooping in Tor using decoys. In *Recent Advances in Intrusion Detection (RAID)*, pages 222–241. Springer, 2011.

[69] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users get routed: Traffic correlation on Tor by realistic adversaries. In *Proceedings of the conference on Computer & Communications Security (CCS)*, pages 337–348. ACM, 2013.

[70] Steven J Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *Symposium on Security and Privacy*, pages 183–195. IEEE, 2005.

[71] Sambuddho Chakravarty, Marco V Barbera, Georgios Portokalidis, Michalis Polychronakis, and Angelos D Keromytis. On the Effectiveness of Traffic Analysis Against Anonymity Networks Using Flow Records. In *International Conference on Passive and Active Measurement (PAM)*, pages 247–257. Springer, 2014.

[72] Jia Zhang, Haixin Duan, and Jianping Wu. A Novel Method to Prevent Traffic Analysis in Low-Latency Anonymous Communication Systems. In *Proceedings of the International Conference on Computer and Electrical Engineering*, pages 906–911. IEEE, 2008.

[73] Ming Song, Gang Xiong, Zhenzhen Li, Junrui Peng, and Li Guo. A de-anonymize attack method based on traffic analysis. In *International ICST Conference on Communications and Networking in China (CHINACOM)*, pages 455–460. IEEE, 2013.

[74] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the annual workshop on Privacy in the Electronic Society (WPES)*, pages 103–114. ACM, 2011.

[75] Jing Jin and Xinyuan Wang. On the effectiveness of low latency anonymous network in the presence of timing attack. In *International Conference on Dependable Systems & Networks (DSN)*, pages 429–438. IEEE, 2009.

[76] Yossi Gilad and Amir Herzberg. Spying in the dark: TCP and Tor traffic analysis. In *International Symposium on Privacy Enhancing Technologies*, pages 100–119. Springer, 2012.

[77] Stjepan Groš, Marko Salkić, and Ivan Šipka. Protecting TOR exit nodes from abuse. In *Proceedings of the International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1246–1249. IEEE, 2010.

[78] Philipp Winter, Richard Köwer, Martin Mulazzani, Markus Huber, Sebastian Schrittwieser, Stefan Lindskog, and Edgar Weippl. Spoiled Onions: Exposing Malicious Tor Exit Relays. In *International Privacy Enhancing Technologies Symposium (PETS)*, pages 304–331. Springer, 2014.

[79] Liu Xin and Wang Neng. Design Improvement for Tor Against Low-Cost Traffic Attack and Low-Resource Routing Attack. In *International Conference on Communications and Mobile Computing (CMC)*, volume 3, pages 549–554. IEEE, 2009.

[80] Michael Backes, Ian Goldberg, Aniket Kate, and Esfandiar Mohammadi. Provably secure and practical onion routing. In *Computer Security Foundations Symposium (CSF)*, pages 369–385. IEEE, 2012.

[81] Daniel Marks, Florian Tschorsch, and Björn Scheuermann. Unleashing Tor, BitTorrent & co.: How to relieve TCP deficiencies in overlays. In *Conference on Local Computer Networks (LCN)*, pages 320–323. IEEE, 2010.

[82] Michael F Nowlan, David Wolinsky, and Bryan Ford. Reducing latency in Tor circuits with unordered delivery. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2013.

[83] Norman Danner, Sam Defabbia-Kane, Danny Krizanc, and Marc Liberatore. Effectiveness and detection of denial-of-service attacks in Tor. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):11, 2012.

[84] Amir Herzberg, Ely Porat, Nir Soffer, and Erez Waisbard. Camouflaged Private Communication. In *Privacy, security, risk and trust (PASSAT), international conference on social computing (socialcom)*, pages 1159–1162. IEEE, 2011.

[85] Marc Mendonca, Srini Seetharaman, and Katia Obraczka. A flexible in-network IP anonymization service. In *International Conference on Communications (ICC)*, pages 6651–6656. IEEE, 2012.

[86] Steven J Murdoch. Hot or not: Revealing hidden services by their clock skew. In *Proceedings of the conference on Computer and Communications Security (CCS)*, pages 27–36. ACM, 2006.

[87] Sambuddho Chakravarty, Angelos Stavrou, and Angelos D Keromytis. Linkwidth: a method to measure link capacity and available bandwidth using single-end probes. *Computer Science Department Technical Report (CUCS Tech Report) CUCS-002, Columbia University*, 2008.

[88] Se Eun Oh, Shuai Li, and Nicholas Hopper. Fingerprinting keywords in search queries over tor. *Proceedings on Privacy Enhancing Technologies*, 2017(4):251–270, 2017.

[89] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[90] NIST-FIPS Standard. Announcing the advanced encryption standard (aes). *Federal Information Processing Standards Publication*, 197:1–51, 2001.

[91] Daniel Marks, Florian Tschorsch, Bjrn Scheuermann, Daniel Marks, Florian Tschorsch, and Bjrn Scheuermann. Unleashing tor, bittorrent and co.: How to relieve tcp deficiencies in overlays (extended version). *Heinrich Heine University, Dusseldorf, Germany*, 2010.

[92] Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of service or denial of security? In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 92–102. ACM, 2007.

[93] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, DTIC

Document, 2004.

[94] Masoud Akhoondi, Curtis Yu, and Harsha V Madhyastha. LASTor: A low-latency AS-aware Tor client. In *Symposium on Security and Privacy (SP)*, pages 476–490. IEEE, 2012.

[95] Fallon Chen and Joseph Pasquale. Toward improving path selection in Tor. In *Global Telecommunications Conference (GLOBECOM)*, pages 1–6. IEEE, 2010.

[96] Hasan T Karaoglu, Mehmet Burak Akgun, Mehmet Hadi Gunes, and Murat Yuksel. Multi Path Considerations for Anonymized Routing: Challenges and Opportunities. In *International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2012.

[97] Andriy Panchenko, Fabian Lanze, and Thomas Engel. Improving performance and anonymity in the tor network. In *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*, pages 1–10. IEEE, 2012.

[98] Chenglong Li, Yibo Xue, Longtao He, and Lidong Wang. TMT: A new Tunable Mechanism of Tor based on the path length. In *International Conference on Cloud Computing and Intelligent Systems (CCIS)*, volume 2, pages 661–665. IEEE, 2012.

[99] Andriy Panchenko, Lexi Pimenidis, and Johannes Renner. Performance analysis of anonymous communication channels provided by Tor. In *International Conference on Availability, Reliability and Security (ARES)*, pages 221–228. IEEE, 2008.

[100] Xin Liu and Neng Wang. RandomWalk-Based Tor Circuit Building Protocol. In *International Conference on Computational Intelligence and Security (CIS)*, volume 2, pages 335–340. IEEE, 2009.

[101] Xin Liu and Neng Wang. An Improved Tor Circuit-Building Protocol. In *International Joint Conference on Artificial Intelligence (JCAI)*, pages 671–675. IEEE, 2009.

[102] Robin Snader and Nikita Borisov. Improving security and performance in the Tor network through tunable path selection. *Transactions on Dependable and Secure Computing*, 8(5):728–741, 2011.

[103] Chenglong Li, Yibo Xue, Yingfei Dong, and Dongsheng Wang. Relay recommendation system (RRS) and selective anonymity for Tor. In *Global Communications Conference (GLOBECOM)*, pages 833–838. IEEE, 2012.

[104] Can Tang and Ian Goldberg. An improved algorithm for Tor circuit scheduling. In *Proceedings of the conference on Computer and Communications Security (CCS)*, pages 329–339. ACM, 2010.

[105] Tao Wang, Kevin Bauer, Clara Forero, and Ian Goldberg. Congestion-aware path selection for Tor. In *Financial Cryptography and Data Security*, pages 98–113. Springer, 2012.

[106] Robin Snader and Nikita Borisov. A Tune-up for Tor: Improving Security and Performance in the Tor Network. In *Network and Distributed System Symposium (NDSS)*, volume 8, page 127, 2008.

[107] Tariq Elahi, Kevin Bauer, Mashael AlSabah, Roger Dingledine, and Ian Goldberg. Changing of the guards: A framework for understanding and improving entry guard selection in Tor. In *Proceedings of the workshop on Privacy in the electronic society*, pages 43–54. ACM, 2012.

[108] Kevin Bauer, Dirk Grunwald, and Douglas Sicker. Predicting Tor path compromise by exit port. In *International Performance Computing and Communications Conference (IPCCC)*, pages 384–387. IEEE, 2009.

[109] Chris Wacek, Henry Tan, Kevin S Bauer, and Micah Sherr. An Empirical Evaluation of Relay Selection in Tor. In *Network and Distributed System Symposium (NDSS)*, 2013.

[110] Aaron Johnson, Rob Jansen, Aaron D Jaggard, Joan Feigenbaum, and Paul Syverson. Avoiding the man on the wire: Improving tor's security with trust-aware path selection. *In the Proceedings of the Network and Distributed Security Symposium - NDSS, 2017*, 2017.

[111] Xin Liu and Neng Wang. Anti-misbehavior System for Tor Network. In *International Joint Conference on INC, IMS and IDC (NCM)*, pages 257–261. IEEE, 2009.

[112] Roger Dingledine, Nick Mathewson, and Paul Syverson. Deploying low-latency anonymity: Design challenges and social factors. *IEEE Security & Privacy*, 5(5):83–87, 2007.

[113] Dhiah el Diehn I Abou-Tair, Lexi Pimenidis, Jens Schomburg, and Benedikt Westermann. Usability inspection of anonymity networks. In *Proceedings of the World Congress on Privacy, Security, Trust and the Management of e-Business*, pages 100–109. IEEE, 2009.

[114] Jeremy Clark, Paul C Van Oorschot, and Carlisle Adams. Usability of anonymous web browsing: an examination of Tor interfaces and deployability. In *Proceedings of the symposium on Usable privacy and security*, pages 41–51. ACM, 2007.

[115] Anne Edmundson, Simpson AKornfeld, Joshua A Kroll, and Edward W Felten. Security Audit of Safeplug "Tor in a Box". In *Workshop on Free and Open Communications on the Internet (FOCI)*. USENIX Association.

[116] Gilles Barthe, Alejandro Hevia, Zhengqin Luo, Tamara Rezk, and Bogdan Warinschi. Robustness Guarantees for Anonymity. In *Computer Security Foundations Symposium (CSF)*, pages 91–106. IEEE, 2010.

[117] Martin Mulazzani, Markus Huber, and Edgar R Weippl. Anonymity and monitoring: how to monitor the infrastructure of an anonymity system. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(5):539–546, 2010.

[118] Markus Huber, Martin Mulazzani, and Edgar Weippl. Tor HTTP usage and information leakage. In *Communications and Multimedia Security*, pages 245–255. Springer, 2010.

[119] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the Tor network. In *Privacy Enhancing Technologies*, pages 63–76. Springer, 2008.

[120] Karsten Loesing, Steven J Murdoch, and Roger Dingledine. A case study on measuring statistical data in the Tor anonymity network. In *Financial Cryptography and Data Security*, pages 203–215. Springer, 2010.

[121] Yao Chen, Radu Sion, and Bogdan Carbunar. Xpay: Practical anonymous payments for Tor routing and other networked services. In *Proceedings of the workshop on Privacy in the electronic society*, pages 41–50. ACM, 2009.

[122] Rob Jansen, Kevin S Bauer, Nicholas Hopper, and Roger Dingledine. Methodically modeling the tor network. In *Workshop on Cyber Security Experimentation and Test (CSET)*. USENIX, 2012.

[123] Rob Jansen and Nicholas Hooper. Shadow: Running Tor in a box for accurate and efficient experimentation. Technical report, Minnesota University, Department of Computer Science and Engineering (No. TR-11-020), 2011.

[124] Kevin S Bauer, Micah Sherr, and Dirk Grunwald. Experimentor: A Testbed for Safe and Realistic Tor Experimentation. In *Workshop on Cyber Security Experimentation and Test (CSET)*. USENIX, 2011.

[125] Prithula Dhungel, Moritz Steiner, Ivinko Rimac, Volker Hilt, and Keith W Ross. Waiting for anonymity: Understanding delays in the Tor overlay. In *International Conference on Peer-to-Peer Computing (P2P)*, pages 1–4. IEEE, 2010.

[126] Karsten Loesing, Werner Sandmann, Christian Wilms, and Guido Wirtz. Performance measurements and statistics of Tor hidden services. In *International Symposium on Applications and the Internet (SAINT)*, pages 1–7. IEEE, 2008.

[127] Mathias Ehlert. I2P Usability vs. Tor Usability A Bandwidth and Latency Comparison. In *Seminar Report, Humboldt University of Berlin*, 2011.

[128] Ryan Pries, Wei Yu, Steve Graham, and Xinwen Fu. On performance bottleneck of anonymous communication networks. In *International Symposium on Parallel and Distributed Processing (IPDPS)*, pages 1–11. IEEE, 2008.

[129] Xiao Wang, Jinqiao Shi, Binxing Fang, and Li Guo. An empirical analysis of family in the Tor network. In *International Conference on Communications (ICC)*, pages 1995–2000. IEEE,

2013.

[130] Florian Tschorsch and Björn Scheuermann. Tor is unfair – And what to do about it. In *Conference on Local Computer Networks (LCN)*, pages 432–440. IEEE, 2011.

[131] Abdelberi Chaabane, Pere Manils, and Mohamed Ali Kaafar. Digging into anonymous traffic: A deep analysis of the Tor anonymizing network. In *International Conference on Network and System Security (NSS)*, pages 167–174. IEEE, 2010.

[132] Nicholas Hopper. Challenges in protecting Tor hidden services from botnet abuse. In *International Conference on Financial Cryptography and Data Security*, pages 316–325. Springer, 2014.

[133] Jörg Lenhard, Karsten Loesing, and Guido Wirtz. Performance measurements of Tor hidden services in low-bandwidth access networks. In *Applied Cryptography and Network Security*, pages 324–341. Springer, 2009.

[134] Ian Goldberg. On the security of the Tor authentication protocol. In *Privacy Enhancing Technologies*, pages 316–331. Springer, 2006.

[135] Rob Jansen, Nicholas Hopper, and Yongdae Kim. Recruiting new tor relays with braids. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 319–328. ACM, 2010.

[136] Roger Dingledine, Dan S Wallach, et al. Building incentives into Tor. In *Financial Cryptography and Data Security*, pages 238–256. Springer, 2010.

[137] Qiyan Wang, Zi Lin, Nikita Borisov, and Nicholas Hopper. rbridge: User Reputation based Tor Bridge Distribution with Privacy Preservation. In *Network and Distributed System Security Symposium (NDSS)*, 2013.

[138] Rob Smits, Divam Jain, Sarah Pidcock, Ian Goldberg, and Urs Hengartner. Bridgespa: Improving Tor bridges with single packet authorization. In *Proceedings of the workshop on Privacy in the electronic society*, pages 93–102. ACM, 2011.

[139] Hooman Mohajeri Moghaddam, Baiyu Li, Mohammad Derakhshani, and Ian Goldberg. Skypemorph: Protocol obfuscation for Tor bridges. In *Proceedings of the conference on Computer and Communications Security (CCS)*, pages 97–108. ACM, 2012.

[140] Zachary Weinberg, Jeffrey Wang, Vinod Yegneswaran, Linda Briesemeister, Steven Cheung, Frank Wang, and Dan Boneh. StegoTorus: a camouflage proxy for the Tor anonymity system. In *Proceedings of the conference on Computer and Communications Security (CCS)*, pages 109–120. ACM, 2012.

[141] Deepika Gopal and Nadia Heninger. Torchestra: Reducing interactive traffic delays over Tor. In *Proceedings of the workshop on Privacy in the electronic society*, pages 31–42. ACM, 2012.

[142] Mashael AlSabah, Kevin Bauer, Ian Goldberg, Dirk Grunwald, Damon McCoy, Stefan Savage, and Geoffrey M Voelker. DefenestraTor: Throwing out windows in Tor. In *Privacy Enhancing Technologies*, pages 134–154. Springer, 2011.

[143] Rob Jansen, Paul F Syverson, and Nicholas Hopper. Throttling Tor Bandwidth Parasites. In *USENIX Security Symposium*, pages 349–363, 2012.

[144] Akshaya Mani and Micah Sherr. Historε: Differentially private and robust statistics collection for tor. In *Network and Distributed System Security Symposium (NDSS)*, 2017.

[145] Stephen Doswell, Nauman Aslam, David Kendall, and Graham Sexton. The novel use of Bridge Relays to provide persistent Tor connections for mobile devices. In *International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 3371–3375. IEEE, 2013.

[146] Christer Andersson and Andriy Panchenko. Practical anonymous communication on the mobile internet using Tor. In *International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm)*, pages 39–48. IEEE, 2007.