Systematic Analysis: Resistance to Traffic Analysis Attacks in Tor System for Critical Infrastructures

Jeremy A. Stone, Neetesh Saxena, and Huseyin Dogan
Department of Computing and Informatics,
Bournemouth University,
Bournemouth, UK
nsaxena@ieee.org, hdogan@bournemouth.ac.uk

***Abstract* — The threat of traffic analysis attacks against the Tor System is an acknowledged and open research issue, especially in critical infrastructures, motivating the need for continuous research into the potential attacks and countermeasures against this threat. This paper aims to provide an in-depth study into the driving technical mechanisms of the current state-of-art Tor System (Browser Bundle and Network) that aim to provide its benefits to anonymity and privacy online. This work presents the countermeasures that have been proposed and/or implemented against such attacks, in a collated evaluation to determine their effectiveness, suitability to Tor Project, and its design aims/goals.**

***Keywords* — anonymity, privacy, tor, traffic analysis, critical infrastructures.**

I. INTRODUCTION

THE Tor system is one of the most popular low-latency mix-based anonymity systems in use today. In 2012 the network consisted of just over 3000 *relays* and about 1000 *bridges*, today the network has more than doubled in size in terms of the volunteer run overlay network, now consisting of over 7000 relays and over 3000 bridges [1]. The system itself consists of a browser bundle, allowing a variety of users to browse the internet and Tor *hidden services* via the second component, the volunteer run overlay network, used to provide the benefits of anonymity and privacy for user's utilizing the network, and also for the administrators/owners of the hidden services.

A. Research Problem

The Tor system provides benefits to user and administrator with anonymity and privacy online, however, it does not claim to provide perfect protection. This was outlined in the original design paper [2], explaining that the system was developed with the intention of balancing anonymity, usability, and efficiency. The threat model outlined within this paper specifies that the system aims to protect against attacks from a non-global adversary (with the ability to control or observe a fraction of the network and its users). Despite the limited threat model, a variety of low-resource traffic analysis attacks have been outlined over its lifetime that fit within this threat model, such as the work of Murdoch & Danezi [3] and Bauer et al. [4]. The traffic analysis issue is acknowledged by developers as an open-research issue, with particular focus on *traffic correlation* and *webpage fingerprinting* attacks (WPF) [5], as previously outlined the issue has been proven across a variety of papers, illustrating that such attacks are possible within the constraints of the specified adversary threat model. Although much research has been made in this area, yet appears to lack a collated/collective evaluation of the countermeasures against such traffic analysis attacks, up to a more current date (May, 2017).

B. Contribution

We aim to outline some of the gaps and shortcomings within the current researched and implemented countermeasures against these attacks, with the intention of opening routes of research in this area. Additionally, we identified the countermeasures by a critical analysis that show potential for use within the Tor system, in terms of effectiveness and overall suitability to the original design goals.

The rest of the paper is organised as follows. Section II overviews background study, and Section III outlines the requirements and design goals of the system used to evaluate the suitability of the proposed countermeasures. Section IV consists of research findings overviewing the countermeasures, their individual attributes and actual functionality in order to provide protection. Section V follows with the analysis and evaluation of the countermeasures and Section VI concludes this work.

II. BACKGROUND STUDY

This section contains an overview of the vulnerable areas identified, through the study of the various traffic analysis attacks that have been researched, performed and simulated against the system lifetime.

A. Entry Relay (Guard) Selection & Rotation Algorithm

The most common area that facilitated the attacks researched was the relay selection algorithm. Almost all the attacks covered rely on the adversary being able to control and/or observe the entry relay (and entry point to the network). This position was later renamed and re-engineered as the *entry guard* relay (as of version 0.1.1.20) [4]. This position is focused upon mainly due to the fact that it is generally the closest relay to either the targeted client, and thus is the only relay that knows the IP address of the target.

Prior to and after the initial introduction of entry guards, the first relay was selected based upon bandwidth ratings and uptime on the network [2] (represented primarily using the "fast" and "stable" relay flags). Later selection was based upon whether a relay held the "guard" flag (determining the relay is suitable for use as an entry guard) [7]. The main issue with this selection mechanism is that initially resource costs were low to run and become an entry relay, even after the introduction of entry guards. Various attacks have reduced the resource requirements of gaining the position by reporting bandwidth and uptime information falsely in order to influence the algorithm, and thus increasing the probability of gaining this position [4], [8]. Øverlier & Syverson [8] revealed further issues with the entry relay selection process when performing attacks upon hidden services, since the attacker can induce circuit creation, allowing the attacker to force re-selection of relays to attempt to gain the position through probability alone.

B. Preserved Inter-Cell/Packet Transmission Timings

The design goals of balancing anonymity and usability, accompanied by a lack of batching strategies result in the retention of timing characteristics during the transmission of packets/cells across the network [3], [14]. These timing characteristics are the main factor facilitating all of the studied traffic correlation attacks (also referred to as timing-correlation attacks), the passive timing attacks studied utilised the natural timing characteristics in the targeted client traffic flows. An example of this is the attack presented by Bauer et al. [4] that exploited the natural timing pattern created upon circuit creation in order to de-anonymise newly created circuits.

Active attacks, however inject patterns (often referred to as active-watermarking) into a client's traffic in order to perform their correlation, the attack presented by Ling et al. [9] injected patterns via a controlled exit relay and correlated the patterns at the entry guard relay. Similar watermarking techniques were administered in the attack presented by Chakravarty et al. [14], however, the injection occurred by utilising a compromised web server, aiming to de-anonymize users accessing the compromised server. Additionally, the WPF attack presented by Panchenko et al. [11] utilises these timing characteristics in order to develop vectors/patterns, created based on the natural packet timings when loading a particular webpage or set of pages, and compared against a vector on target client's traffic.

C. Packet/Cell Order, Amount, Interval, Size and Direction

The main basis of WPF attacks is to create vectors or fingerprints based upon distinguishable traffic characteristics, such as packet/cell transmission timings.

The attack presented by Shi and Matsuura [10] focusses on what the author referred to as an "interval" in order to develop its vectors. The interval was said to be a recording of the number of inbound packets transmitted to the target client, each interval is separated by an outbound packet/flow. Panchenko et al. [11] created vectors based upon both inbound and outbound packets, recording the size of each packet, the direction of flow, the timings at which they are transmitted, as well as the observed packet sequence and total amount of packets transmitted. Cai et al. [12] extends the work of Panchenko et al. [11]. Their attack differs by developing their vectors based upon the observed sequence (including amount) and the observed direction of transmitted packets. The accuracy is improved with new techniques for vector production, to include website based vectors, as well as hot (cached page load) and cold (un-cached page load) vectors. Additionally, the authors simulate their attack, and the previously outlined attack by Panchenko et al. [11] with the addition of the newly implemented HTTPOS randomisation and pipelining countermeasure, claiming both attacks resist the countermeasure to a high degree. Wang & Goldberg [13] present further improvements to WPF attack accuracy by developing its vectors/fingerprints after extracting the fixed-size Tor cells from the TCP packets transmitted, instead recording the amount and direction of the extracted cells. In addition, new string-based vector and comparison algorithms are employed to perform the fingerprinting, offering further accuracy improvements and reduced resource costs.

III. MODELLING REQUIREMENTS

In order to evaluate the countermeasures, we determine the grounds at which they should be evaluated. A set of goals was developed based on the original goals of the Tor system, and the defined threat model. These are explained and elaborated below.

A. System Model/Goals

The original Tor design paper states that the main aim of the system is to frustrate attackers that are attempting to link communication partners (e.g., traffic from user to server/ hidden service, or instant messaging partners) or attempting to link multiple communications (or traffic) to a single user [2]. However, the paper continues to elaborate that this main aim has been met with the consideration of several other design goals, the most relevant to this research have been summarised below.

Deployability: The paper states that the system must not be expensive to run, must not place liability on relay operators, and the system must not be difficult or expensive to update or implement. This additionally covers compatibility across different systems and platforms [2].

Usability: As a mix-based system, Tor relies upon its popularity to ensure the strength of the anonymity provided, it essentially hides traffic alongside other users' traffic**.** Thus, it is important to ensure the system is both easily accessible and stress free in use, to ensure the growth and continued use of the system. This covers ease of installation, configuration, operation as well as the speed at which the service is delivered [2].

In terms of the evaluation, these design goals are useful for determining the actual suitability of the researched countermeasures.

B. Adversary Threat Model

The Tor Project does not claim to provide perfect anonymity to its users, the authors of the original design paper state that this is the case for all low-latency anonymity systems due to their inherent balance of security (anonymity & privacy), usability and efficiency [2]. The paper continues to specify the threat model the system aims to protect against, thereby defining the level of protection that should be provided by the system. The adversary is specified as a "non-global adversary", this is elaborated as an adversary possessing limited resources, whom can observe and/or control a subset of the relays and clients over the network [2]. In addition, the adversary is said to be capable of manipulating the traffic passing through his/her controlled resources (clients, relays, routers) to potentially inject a pattern and/or modify user traffic.

All of the attacks selected for this research fit within this limited adversary model, thus, it is of course important that the countermeasures provide protection against these threats.

C. Countermeasure Security and Suitability Goals

Each of the goals at which the countermeasures shall be evaluated has been listed and elaborated below. They have also been classified using the MoSCoW classification system in order to define their weightage and importance.

1.0 (Must) Protect against traffic correlation attacks within the specified threat model: This goal was established based upon the previously outlined threat model, it is classified as a "must" goal as it determines the minimum level of protection desired.

1.1 (Could) Prevent (all researched) traffic correlation attacks: This goal extends upon 1.0 to cover prevention, however, achieving complete prevention of this threat was still determined as a desirable, but perhaps unrealistic goal, and thus was classified as "could".

2.0 (Must) Protect against website/webpage fingerprinting attacks, within the specified threat model: This goal is derived from the threat model similar to goal 1.0, but covers WPF attacks.

2.1 (Could) Prevent (all researched) WPF attacks: Again, it is desirable but perhaps unrealistic goal similar to 1.1.

3.0 (Should) Not require prior configuration, decision making, or professional knowledge from the user: It is derived from "Deployability" design goal, system should be easy and inexpensive to set-up/install. However, as different users desire different levels of anonymity the classification of "should" was assigned, minor configuration may be necessary to set the desired level of anonymity.

4.0 (Should) Have the ability to withstand the potential growth of the network: This goal was elicited based upon the constant growth of Tor and its users. The classification of "should" has been assigned due to the dynamic nature of this area (i.e., constant progression and growth) [1]. If a countermeasure relies on there being a certain amount of bandwidth/processing power available, or is not effective under high load, then this goal has not been met.

5.0 Not impose excessive overheads: This goal has been based upon both the "Deployability" and "Usability" design goals of Tor, as well as the fact that it is intended to be a low-latency system. A countermeasure "must" not impose excessive overheads to the user, system or service/relay operators. This can be in the form of resource costs (monetary and bandwidth), as well as increases in latency upon the service provided.

6.0 Not require all clients to utilise the measure for the system to continue operating: Elicited upon the "Usability" and "Deployability" goals of the system, implemented countermeasures "must" be compatible with previous version of Tor to ensure that the current run relays continue to provide their resources to the network. A reduction in available resources could result in a slower service, conflicting with the usability design goal and coincidentally a slower service could result in less users, thus degrading the strength of the anonymity.

IV. RESEARCH FINDINGS

This section details the basic function of each of the researched countermeasures that aim to provide protection against the outlined traffic correlation and WPF threats. Additionally, the current status of each countermeasure is detailed (i.e., currently implemented, proposed/under-discussion in the Tor bug tracker, or suggested within relating research).

A. Entry Guard Selection and Rotation Algorithm Updates

As previously outlined the entry relay was initially selected based upon uptime and bandwidth ratings alone, in order to raise the resource cost of becoming such a powerful relay the entry guard concept was introduced. Initially 3 guard relays were selected by clients, keeping them for a period of 4-8 weeks before rotation, or if more than one guard becomes unreachable (0.1.1.11-alpha through 0.2.4.11-alpha) [15]. Elahi et al. [16] demonstrated that the rotation of guards increases the probability that a malicious relay shall be selected by a client, because of this the rotation period was increased to 2-3 months (as of 0.2.4.12) [15]. Further research into guard selection/rotation performed by Dingledine et al. [17] demonstrated that using a single guard for as long as 9 months would greatly decrease this probability even further. This generated the issues with load balancing across newer guards, because old guards are being rotated less, new guards are only likely to be selected by new clients (with a slim chance due to the uptime weighting) [18]. A new guard selection algorithm was introduced to adjust how guards are selected, the main basis is allowing a relay to become more "guardy" gradually, this means a relay that is only 50% guard will still be used in the middle and exit relay positions in order to retain the use of their resources (Tor 0.2.x-final) [19].

B. Multipath Routing Schemes – mTorHS and Conflux

mTorHS [21] and Conflux [22] are both multipath routing schemes suggested for the use in the Tor system. Both of these schemes feature flow splitting and merging functionalities, sending a single client traffic via multiple Tor circuits in order to both distort the flows (providing improved anonymity and distorting packet sequences and packet/cell timings) as well as balance the load across the network.

C. HTTPOS Randomised Pipelining

Currently implemented (as of TorBrowserBundle 2.2.x-stable) browser-based countermeasure provides its protection utilising traffic transformation techniques [23]. The countermeasure obfuscates identifiable traffic characteristics utilised within WPF attacks (packet count, order, size) by processing TCP requests whilst administering order randomisation [24], [12].

D. Improved Bandwidth Verification

To prevent malicious relay operators from making false bandwidth claims and influencing the relay/path selection algorithm, Perry [26] introduced "bandwidth authorities" (bwauths). Essentially the "bwauths" actively probe relays at regular intervals, logging their perceived bandwidth capabilities and comparing them to similar level relays, and then adjusting the weights accordingly. This results in a relay's bandwidth weight gradually increasing over time until it finally peaks and settles [18], [26].

E. Padding Schemes

A variety of padding schemes have been identified and detailed below, the general purpose of each scheme is to obscure the timings of packet/cell transmission.

1) Adaptive Padding (AP): The AP scheme is implemented and provided by the relays upon the network, timing patterns are reduced by this defence by inserting dummy packets into statistically unlikely gaps in inconsistent traffic flows [27]. The defence is suggested by Juarez et al. [28] and is also proposed as a possible WPF defence within the Tor bug Tracker. However, the proposal is still open to alternative schemes [29].

2) Dependent Link Padding (DLP): Wang et al. [25] proposed DLP, an alternative relay-based padding scheme. This scheme utilizes a combination of packet delays and dummy packets in order to obfuscate the timing patterns within packet flows. The delays are managed using a strict upper boundary to limit the length of the delays administered, similarly the insertion of dummy packets uses a minimum transmission rate [25]. Utilising packet delays alongside dummy packets remove timings more effectively than simply filling the gaps, making all flows over the network appear similar [25].

3) Defensive Dropping (DD): DD differs from AP and DLP. Instead of injecting dummy packets by relays, the injection is performed by the client/initiator. The packets are administered at a constant rate across the network removing all identifiable timings. Additionally, the scheme features packet dropping to obfuscate the observed number of packets being transmitted, dummy packets are selected to be "dropped" by individual relays along the circuit.

V. Analysis and Evaluation

A. Entry Guard Selection and Rotation Updates

The updates to the algorithm provide protection against 4 of the 5 studied timing/traffic correlation attacks, fulfilling goal 1.0. The countermeasure also fulfils goals 3.0 and 5.0, in that it requires no interaction from the user, nor configuration or knowledge in order to function. In addition, the refined load-balancing features of the algorithm ensure that there is no additional resource cost for the increased anonymity provided, each of these elements demonstrates good suitability for the design goals of the system. Despite refinement, it is of course still possible for an adversary to gain the entry guard position (with increased resource costs), this means it cannot fulfil goal 4.0, this combined with the fact that it is not a goal of all attack variations means it is not sufficient on its own. There are all remaining open issues, such as the case where an attacker attempts to perform a denial-of-service on the single guard node?, how long before reverting back?, etc., and thus, does not yet meet goal 4.0 [18].

B. Multipath Routing Schemes – mTorHS and Conflux

Multipath routing schemes offer the improved performance by load-balancing across the lesser used, lower bandwidth relays dynamically, and showing very good potential for withstanding the potential future increases in network load (meeting goal 4.0 and 5.0) [21], [22]. The flow splitting and merging properties result in the potential destruction of the packet transmission timings that many timing correlations and WPF attacks covered here exploit (meeting goal 1.0 and potentially 1.1 when configured aggressively). The scheme also distorts the packet ordering, making this a potential defence for 3/5 studied WPF attacks (mostly meeting goal 2.0). Goal 3.0 can also be met by this measure, as long as the configuration is predefined, the standard user needs not to understand the system to still be effective. There are still open issues with the scheme, first, is the fact that it requires each end of the circuit (relays or clients depending on configuration) to be up to date for the flow splitting/merging to be applied. Also, multiple entry guards and exit configuration conflicts with the single-entry guard update. All of these aspects conflict with goal 6.0, suggesting the need for further research on how to deploy and manage the countermeasure.

C. HTTPOS Randomised Pipelining

Cai et al. [12] simulated their (and Panchenko et al.'s [11]) WPF attack, claiming the defence had little effect on the accuracy of these attacks. However, further research into this area revealed that a bugged version of the Tor browser was tested (suffering from a lack of randomisation) [20]. The

measure has the potential to reduce the accuracy of all the WPF attacks covered (potentially satisfying goal 2.0), outlining the need for further research in order to establish whether the experimental measure is effective [24]. HTTPOS imposes little extra overhead, in keeping with the low-latency design of the system (satisfying goal 5.0). The countermeasure itself is part of the Tor Browser, defaulted to be activated, posing no compatibility issues, thus fulfilling goal 6.0. Unfortunately, in order for the countermeasure to work the web server being accessed must have the feature activated. Similarly, a malicious relay could deactivate the feature, this sensitivity conflicts with goal 4.0, and suggests the need for alternative measures that do not rely on certain conditions in order to be effective [24].

D. Improved Bandwidth Verification

The bandwidth verification algorithm is a supporting defence for the entry guard selection updates, the measure raises the resource requirement to become a guard relay. The measure requires no configuration as it runs in the background periodically across the network (meeting goal 3.0), as the "bwauths" are external machines, there is no observable issue with compatibility allowing relays to operate within the network as normal (satisfying goal 6.0) [15]. Additionally, measurements are generally a small transfer to ensure that the relay is not unnecessarily loaded, satisfying goal 5.0 [15], [26]. Although the bandwidth authorities frustrate an adversary from influencing the relay bandwidth weighting selection algorithm, it is still not impossible to cheat the authorities. Thill [20] details a method to fool the authorities. This can be achieved due to the fact that the "bwauths" have publicly available IP addresses, allowing an adversary to dynamically assign bandwidth during measurements. Because of this, the countermeasure does not satisfy goal 4.0, suggesting the need for further refinement to ensure that an adversary cannot deploy a relay with the appearance of a high resource relay.

E. Delay and Padding Schemes

1) Adaptive Padding: AP provides some resistance to the traffic/timing correlation attacks. Similarly, resistance is also provided to 3 of 4 WPF attacks that rely on counting the number and timing of packets, due to the dummy packets inserted (thereby mostly meeting goal 1.0 and goal 2.0) [27]. Adaptive padding is primarily administered by relays and should not require any configuration or interaction from the user or the relay operator, thus mostly satisfying goal 3.0. Compared to the other techniques studied, AP incurs the least bandwidth and latency costs, ensuring that it is in meeting with goal 5.0 [28]. An issue with this scheme is the requirement for relays to administer the technique, requiring the relay to be up-to-date, otherwise padding will only be applied from the outgoing flow. This is not ideal as most WPF attacks rely on fingerprinting incoming packet flows. A lack of significant delay means that actively injected timing patterns still remain hidden within the dummy traffic. This allows an attacker to potentially reverse engineer the padding in order to find the injected timing pattern, because of these reasons it cannot be said to meet goal 4.0, and most likely requires further research in order to be implemented more effectively [27], [28], [25].

2) Dependent Link Padding: DLP should prevent 4 of the studied timing correlation attacks and could potentially prevent 3 of the WPF attacks that rely on packet counting in order to correlate their fingerprints, making it the strongest studied padding scheme and thus meets goals 1.0, 1.1, 2.0 and 2.1 [25], [6]. The technique requires no set-up from the user providing the strict delay bound is predefined (meeting goal 3.0). The main downfall of this scheme is also the source of its strength, the volume of dummy traffic required to match the fastest traffic on the network, comes at a high bandwidth cost, conflicting with goals 4.0 and 5.0 [25], [6]. DLP is administered client-side, which means it does not rely on relays implementation of the measure to provide its benefits, nor would it stop the network functioning (meeting goal 6.0). However, this design has an issue, web servers or hidden services must administer the feature to ensure that return traffic is padded. This is not desired, especially, as WPF attacks focus on incoming traffic. The combination of these downfalls and conflicts illustrate that it is unsuitable for use on the Tor network, which aims at providing a usable and low latency service.

3) Defensive Dropping: DD is the final padding technique under evaluation, much like dependent link padding defensive dropping is administered by the client and relays independently. This means that a client can utilise its benefits immediately upon their outgoing traffic, satisfying goal 6.0. However, for the returned traffic to be padded, the relay must have the measure implemented for the benefits to be delivered to inbound traffic. In addition, these relays will not drop the dummy packets and continue to forward them introducing some excess traffic to the network, which is in conflict with goals 4.0 and 5.0. The countermeasure removes both natural and injected timing characteristics, thereby protecting against timing based correlation and WPF attacks (4/5, mostly meeting goal 1.0), the dropping of dummy packets also alters the number of packets that can be observed during transmission, thereby frustrating packet counting-based WPF attacks (satisfying goal 2.0).

VI. Conclusion and Future Work

This study establishes that the updated guard selection and rotation algorithm is a great improvement upon the original, increasing the resources required for an adversary relay to gain the guard status. However, the remaining open questions behind a single guard suggest the need for further research in this field. HTTPOS pipelining showed much promise with the updates to randomization, but a lack of literature utilising the updated version means that its effectiveness is inconclusive. Thus, it requires further research in order to perform a more

conclusive evaluation of the measure. The bandwidth probing improvements supersede previously unverified implementation, further increasing the resource requirements. Unfortunately, these authorities can still be fooled, albeit still requiring good burst bandwidth in order to influence the bandwidth weights. Multipath routing schemes offered great improvements to load-balancing and performance, meeting the design goals of the system. Unfortunately, due to the unclear deployment, further research in this field would be required. This is particularly true in case of its conflicts with the use of single entry guard relay. A variety of padding based schemes were analysed, finding that they generally pose excessive overheads in order to provide strong protection, the two most suited schemes are Adaptive Link Padding and Defensive Dropping. The common theme with these schemes was the requirement of relay co-operation for the protection to be provided in both traffic directions.

Future work in this area would benefit from testing the countermeasures, upon the system in its current state, as it was found much of the literature was based around previous versions of Tor (HTTPOS in particular). Simulating the studied attacks on the current system would enable a more thorough evaluation of their effectiveness and performance under different conditions. The research, however, outlined the potential of multipath routing schemes, enabling better utilization of the bandwidth being offered by smaller contributors to the network. The increased diversification of the relays used could potentially increase the strength of anonymity provided, as well as provide performance benefits. The main issue outlined with this proposal was the unclear deployment and management of the concept, potentially requiring network-wide co-operation in order for the measure to be applied. Furthermore, the multipath conflicts with a single guard suggest that the research around the guard selection algorithm should be performed in collaboration with such a scheme in order to address the conflicts in deployment.

REFERENCES

[1] The Tor Project, "Servers – Tor Metrics," 2017. [Online]. https://metrics.torproject.org/networksize.html.

[2] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation Onion router," *In Proc USENIX Security Sym.*, 2004.

[3] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," *In Proc. IEEE Symposium on Security and Privacy*, May 2005.

[4] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against ToR," *In Proc. Workshop WPES*, 2007.

[5] The Tor Project, "Tor Research: Research Ideas," 2008. [Online]. https://research.torproject.org/ideas.html.

[6] C. Diaz, S. Murdoch, and C. Troncoso, "Impact of network topology on anonymity and overhead in low-latency anonymity networks," *In Proc. PET Symposium*, pp. 184-201, 2010.

[7] The Tor Project, "torspec - Tor's protocol specifications," 2017. [Online]. https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt.

[8] L. Øverlier and P. F. Syverson, "Locating hidden servers," *In Proc. IEEE Symposium on Security and Privacy*, May 2006.

[9] Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia, "A new cell counter based attack against Tor," *In Proc. ACM Conference on Computer and Communications Security*, 2009.

[10] Y. Shi and K. Matsuura, "Fingerprinting attack on the Tor anonymity system," *In Proc. Information and Communications Security, LNCS 5927*, pp. 425-438, 2009.

[11] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website Fingerprinting in Onion Routing-based Anonymization Networks," *In Proc. Workshop on Privacy in the Electronic Society*, pp. 103-114, 2011.

[12] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a Distance: Website Fingerprinting Attacks and Defenses," *In Proc. ACM Conference on Computer and Communications Security (CCS)*, 2012.

[13] T. Wang and I. Goldberg, "Improved Website Fingerprinting on Tor," *In Proc. Workshop on Privacy in the Electronic Society*, pp. 201-212, 2013.

[14] S. Chakravarty, M. V. Barbera, G. Portokalidis, M. Polychronakis, A. D. Keromytis, "On the effectiveness of traffic analysis against anonymity networks using flow records," *LNCS* 8362, pp. 247-257, 2014.

[15] The Tor Project, "The lifecycle of a new relay | The Tor Blog," 2013. [Online]. https://blog.torproject.org/blog/lifecycle-of-a-new-relay.

[16] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg, "Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor," *In Proc. ACM Workshop WPES*, 2012.

[17] R. Dingledine, N. Hopper, G. Kadianakis, and N. Mathewson, "One fast guard for life (or 9 months)," *In Proc. HotPETs*, 2014.

[18] The Tor Project, "Improving Tor's anonymity by changing guard parameters | The Tor Blog," 2013. [Online]. https://blog.torproject.org/blog/improving-tors-anonymity-changing-guard-parameters.

[19] The Tor Project, #9321 (Load balance right when we have higher guard rotation periods) – Tor Bug Tracker & Wiki, *Trac.torproject.org*, 2013. [Online]. https://trac.torproject.org/projects/tor/ticket/9321.

[20] The Tor Project, "A Critique of Website Traffic Fingerprinting Attacks, The Tor Blog," 2013. [Online]. https://blog.torproject.org/blog/critique-website-traffic-fingerprinting-attacks.

[21] L. Yang, and F. Li, "Enhancing traffic analysis resistance for Tor hidden services with multipath routing," *In Proc. IEEE Conference on Communications and Network Security (CNS)*, 2015.

[22] M. Alsabah, K. Bauer, T. Elahi, and I. Goldberg, "The Path Less Travelled: Overcoming Tor's Bottlenecks with Traffic Splitting," *In Proc. PET Symposium*, pp. 143-163, 2013.

[23] The Tor Project, #3913 (Enable pipelining) – Tor Bug Tracker & Wiki, 2011. [Online]. https://trac.torproject.org/projects/tor/ticket/3913.

[24] M. Perry, "Experimental Defense for Website Traffic Fingerprinting | The Tor Blog," 2011. [Online]. https://blog.torproject.org/blog/experimental-defense-website-traffic-fingerprinting.

[25] Wei Wang, Mehul Motani, and Vikram Srinivasan, "Dependent Link Padding Algorithms for Low Latency Anonymity Systems," *In Proc. ACM CCS*, pp. 323-332, Nov. 2008.

[26] M. Perry, "TorFlow: Tor network analysis," *In Proc. HotPETs*, 2009, pp. 1-14.

[27] V. Shmatikov and M. Wang, "Timing analysis in low-latency mix networks: Attacks and defences," *In Proc. ESRCS*, 2006.

[28] M. Juarez, M. Imani, M. Perry, and M. Wright, "Toward an Efficient Website Fingerprinting Defense," *In Proc. ESRCS*, pp. 27-46, 2016.

[29] The Tor Project, #7028 (Implement Adaptive Padding or some variant and measure overhead vs accuracy) – Tor Bug Tracker & Wiki, 2012. [Online]. https://trac.torproject.org/projects/tor/ticket/7028.