Mitigating Fingerprinting Attacks Against Location Based Path Selection in Tor

Avishek Mondal
mondal@princeton.edu

I. Problem Motivation

Tor is a deployed system that is widely used for anonymous communication [1]. However, there are numerous known attacks that can deanonymise users of Tor, including end-to-end attacks as outlined in the original design paper [1] and the RAPTOR suite of attacks [2], wherein an adversary utilises BGP prefix hijacks against the Tor network. Changing the guard selection algorithm in the Tor protocol, which is currently based solely on bandwidth, has been a proposed defence method against a number of proposed attacks, for example Counter-RAPTOR [3], LASTor [4] and DeNASA [5]. However, these path selection defences could potentially increase the vulnerability of the Tor network against a suite of temporal fingerprinting attacks where an adversary can deanonymise a client using the Tor network by observing his choice of guard relay over a period of time (no citation available at time of writing for the paper describing this kind of attack, as it has not yet been published). In this project, we seek to defend the Tor network and the anonymity of its users against such fingerprinting attacks.

II. Novel Contribution

The main contribution of this project would be the concept of clustering clients. This means that several clients at once will exhibit the same behaviour with regards to guard selection and thus, fingerprinting attacks will not be able to uncover the identity of the true client. This ensures a minimum anonymity set. It is expected that effectiveness of this defence will vary based on the guard selection algorithm implemented (for example, vanilla Tor vs Counter-RAPTOR vs DeNASA etc.). Thus, the contributions of this project can be summarised by the following -

1) **Defining "effectiveness" -** It is expected that future work will describe further attacks and defences of the Tor network. It is useful to have metric to compare how "effective" such ideas are compared to each other. Thus, metrics will be developed to compare the following -
   a) Security - How secure the data is through the Tor network, i.e. integrity and correctness of the data through the Tor network and whether or not it is received and can decrypted only by the intended recipient
   b) Privacy - How well the identity of the user and recipient are hidden
   c) Latency - How do the measures affect the latency, and hence utility of an anonymity system
2) **Clustering algorithms -** The project also hopes to contribute clustering algorithms that will implement client aliasing, will perform satisfactorily on the metrics defined above on at least some of the proposed path selection protocols. The main path selection protocols under consideration would be vanilla Tor (i.e. the existing path selection protocol already implemented in Tor), Counter-RAPTOR and DeNASA.

III. Background

A. Tor network

The Onion Router (Tor) is a popular system that enables anonymous communication, and the system described in [1]. Tor routes traffic from a client to a destination through 3 relays - commonly referred to as guard, middle and exit relays. To assure perfect forward secrecy, Tor uses an incremental (often described as "telescoping") path building design, where each relay only knows the address of the previous and next hop.

1) Guard selection in Tor: One of the goals of the creators of Tor was low-latency. [1]. To achieve this, the Tor network attempts to balance traffic over available bandwidth during the guard relay selection for each client [5]. More trusted Tor nodes, called *directory authorities* (or DAs) maintain the state of the network and prepare something called the *consensus* (this can be found in [6]) and distribute it to all clients.

Guard selection is then carried based on the bandwidth available to each possible guard selection [7]. Guards are rotated every 60 days to 9 months [5][8]. The default value of 2 months is in the Tor source code [9]. A new guard is chosen if the old guard becomes unavailable, or if it is at the end of this rotation period.

2) Possible attacks on the Tor network: The original authors of the Tor project conceded in [1] that the Tor network will not be able to ensure anonymity if the adversary has compromised both the client-guard link and the exit-destination link, as the adversary can then carry out traffic co-relation and match the client to the destination. There are several other attacks that are possible on the Tor network that can deanonymise its users. [2] has shown that given the natural network churn of the Internet topology, an autonomous system (AS) level adversary does not need to always have visibility of the traffic between both the client-guard and exit-destination links to be able to carry out traffic co-relation and deanonymise users. It also shows how the Tor network is also vulnerable to active attacks by AS level adversaries. [10] further shows how the effects of temporal dynamics degrade user anonymity in Tor. The rest of this section will go through these proposed attacks and defences against them.

B. RAPTOR

The Raptor (or "Routing attacks against privacy in Tor") suite of attacks are explored in [2]. The paper shows 3 main ways user anonymity is decreased on the Tor network:

1) A new, asymmetric traffic analysis method - TCP headers and ACKs are monitored by a compromising AS (i.e. an AS that is both in the path of client-guard and exit-destination) and correlation is computed (i.e. Spearman's rank correlation coefficient is computed) and the server trace with the highest correlation is matched to the client traffic.
2) Natural churn - Analysis of how path changes due to the BGP protocol increases the chances of an AS becoming a compromising AS.
3) Active attacks - Both hijacks, where packets are dropped, and interception, where path vector is modified so that traffic passes through an adversarial AS before going through the guard relay so that attack 1 can be carried out, are shown.

.

What this attack demonstrates, is that an adversary that is weaker than that conceded by the designers of the Tor system too can successfully compromise user privacy in Tor, by either passively collecting metadata on the user, or by actively targeting the Tor network through BGP attacks.

C. Counter-RAPTOR

In response to the Raptor suite of attacks, the Counter-Raptor was a suite of defence mechanisms proposed to make the Tor network more resilient, outlined in [3]. There are three main contributions of [3]:

1) A new measurement study on the resilience of the Tor network to active BGP prefix attacks
2) A new algorithm for guard relay selection that incorporates resilience to active BGP attacks in its selection
3) BGP monitoring system that can detect routing anomalies on the Tor network in real time

The guard selection algorithm is defending is against a BGP hijack attack. For each guard relay that a client can choose, the client calculates a quantity *resilience*, i.e. the probability that an adversarial AS will succeed in hijacking the client's traffic to the guard relay by falsely announcing itself to be the origin AS for the prefix that contains the guard relay. A linear combination of *resilience* and *bandwidth* using the Tor consensus from the DAs is calculated and assigned as the weight of the guard relay. The relay with the highest weight is then selected as the guard relay. The key difference of this guard selection algorithm from the one implemented in vanilla Tor is the fact that the algorithm depends on the client's location.

D. DeNASA

DeNASA (or "Destination-Naive AS-Awareness") is another location based path selection scheme, proposed in [5]. DeNASA uses the method outlined in [11], to infer ASes that could be on a network path. [11] does this by examining the business relationships between ASes and how this would affect the BGP routing. Based on this, the ASes most likely to be able to carry out traffic co-relation are identified as *Suspect ASes*. Following that, an AS-Aware path selection algorithm is implemented. The algorithm is made of two parts:

1) e-select: This method determines how the traffic exits the exit relay. The client chooses a guard relay from a pre-specified list (called "guard list") first, and then, by bandwidth weighting, an exit relay is chosen. For all suspect ASes that are on the client-guard link, the probability of that guard also being traversed between the exit relay

and the final destination is calculated (the final destination need not be known). If this probability is lower than a certain value, the chosen exit relay is accepted.

2) g-select: This is the method that is used by the client to choose guards to add guard relays to the guard list. Clients choose guards using the bandwidth weighted way vanilla Tor chooses guards, on the condition that there is no suspect AS on any part of the client-gaurd link.

In [5], g-select and e-select can be used independently of each other, or together. e-select is destination naive and not a location based path selection algorithm. g-select on the other hand, depends on the client's location and is thus a location based path selection algorithm. The authors of [5] conclude that such an algorithm reduces the vulnerability of the Tor stream by 74%.

E. Tempest

Tempest (or "Temporal Dynamics in Anonymity Systems") is a suite of attacks proposed in [10] against vanilla Tor, newer location based path selection protocols proposed for Tor, and proposed new Internet infrastructures. It uses a utilises the information leaks due to the following to carry out such attacks:

1) Client mobility
2) Usage patterns
3) Changes in the underlying network routing

Depending on the defence in place, [10] shows how attacks that leverage the above 3 can reduce user privacy. We summarise their findings on attacks against Counter-RAPTOR and DeNASA below.

1) Against Counter-RAPTOR: [10] utilises the fact that Counter-RAPTOR's guard selection algorithm chooses a guard based on the client's initial location, and does not take into account that a client can move. It goes on to show that if a client moves more than 10 times (to different countries) the probability that a hijack attack by an adversarial AS client will succeed (and thus allow an adversarial AS to carry out traffic co-relation of both the client-guard link and the exit-destination link) increases to over 90%.

2) Against DeNASA: For DeNASA, [10] utilises the fact that the suspect-free ASes that a client can choose for the guard list in the g-select method is dependent on a client's local network characteristics. Thus, the guard selection itself leaks information, because guard selection likelihoods from different locations can be aggregated over time. These information leaks can be combined with other known guard discovery attacks for the adversary to create a set of possible client locations. [10] shows how multiple guard selections by the client that are observed by an adversary results in a sharp drop in entropy in the adversary's set of possible client locations.

The biggest implication of the Tempest suite of attacks is that even with the state-of-the art defences against possible attacks on Tor, over time, there will be a reduction in user privacy over time. This inspires two lines of counter-measures -

1) Reducing the rate at which the measure of "user privacy" degrades - Given that a reduction in privacy is inevitable over time, the first defence is to naturally consider ways which would slow this down as much as possible. This means that the time frame for a user's privacy to be significantly eroded would increase, to the order of months and years and thus, raise the bar for the type of adversary that can significantly reduce anonymity on the Tor network
2) Ensure a minimum notion of "user privacy" - This entails ensuring that regardless of the time frame in which the adversary carries out passive temporal attacks like Tempest or the network state, Tor clients can be assured of some minimum guarantee of "privacy"

These two counter-measures are complementary to each other. This thesis will focus on developing the latter counter-measure against temporal attacks on the Tor network.

IV. Client Aliasing

To ensure the minimum notion of user privacy mentioned in the second part of III-E.2, client aliasing is a scheme that can be pursued. This is applicable to any location based path algorithm. Clients are clustered based on how similar they are choosing a guard relay on the Tor network, and subsequently, all members of a cluster follow the guard choice of a representative AS of that cluster.

A. Client Aliasing in DeNASA

For client aliasing in DeNASA, client aliasing is done by clustering client ASes. The "similarity" metric that will be used in DeNASA will be the probability of a client in a particular AS choosing a guard relay, based on DeNASA's g-select algorithm. The code base used in [10] for temporal analysis of DeNASA is relied on.It is assumed that all clients in a particular AS will choose

a particular guard relay with the same probability. The steps in the analysis are outlined in this section.

1) Calculating probabilities of clients within an AS to choose particular guards under DeNASA: [1] The first step would be to calculate the probabilities of of clients within an AS choosing particular guard relays. The steps for this is similar to [10]'s analysis of DeNASA, and described here.

a) Predicting the path between any two ASes: The first step would be to predict the path between any two ASes on the internet. The inputs required would be -

i CAIDA's internet topology data with inferred paths between every AS on the Internet, from [12].
ii A hash table and function that indexes each path

Using these inputs, AS the paths between any two ASes can now be predicted. This is then applied to the Tor network. The top 95 ASes containing Tor clients, similar to Counter-Raptor [3] are first collated. The ASes that Tor's guard relays are in must then be found. The Tor project provides the latest network state consensus in [6] (which is voted on by a majority of DAs on the Tor network), which allows Tor clients to have an up-to-date file of the possible guards they can choose for their path. Using the IP addresses provided there, the AS numbers on which the Tor guard relays lie can then be ascertained.

b) Probability of guard selection in DeNASA: DeNASA has 2 methods of altering the guard selection of the Tor network. One is e-select which does not depend on the path that the Tor traffic traverses, and the other one is g-select. In g-select, a Tor client cycles through all potential guard choices until it reaches a guard that will allow the traffic to traverse a path that is Suspect AS free. The Suspect AS list is generated by collecting the list of ASes that are on both (a) the paths to the ASes of the guard relays from the ASes that are contain most of the Tor clients and (b) the possible path from the exit relay's AS. To generate the probability table of a client AS choosing a particular guard relay under g-select, the following steps are used -

i All guard relays that lie on a suspect AS are removed
ii The bandwidths of the remaining client-guard links are then normalised with respect to the total bandwidth of all the paths from the client to a Tor guard relay.

[1] script.py file

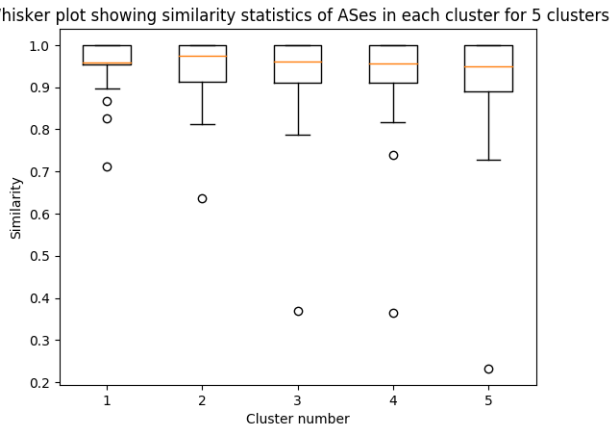

Fig. 1. Figure showing the whisker plot of the clusters when k = 5. The circles show the outliers

The result of the last step returns the probability of the client in that AS choosing a particular guard relay. This is computed and presented in a table so that client ASes can now be clustered.

2) Naive k-clustering based on probability table: [2] The probability table can now be used to cluster the top 95 ASes on which Tor clients lie mentioned in the previous section. Cluster representatives are first chosen by selecting the k most dissimilar metric. There are many choices on which similarity metric this is done. We first investigate the implications of choosing a naive implementation of the similarity metric by choosing the complement of the Euclidean norm of between the ASes (the features on which the norm is calculated would be the probability of clients in the AS choosing a guard relay). Thus the more similar an AS is, the higher this similarity metric is. The results of this clustering for k = 5 is shown in the figure 1. To get a better representation of the statistics in each cluster, the cumulative distribution can be plotted, and is shown in figure 2. Increasing the number of clusters will increase the similarity of member ASes within each cluster to the AS that is the cluster representative. This is demonstrated when k is increased from 5 to 10. The whisker plot is in figure 3 and the corresponding CDF is plotted in figure 4.

3) Security implications of naive k-clustering: What are the security implications of the clustering method described in the previous section? We have to define a metric, arbitarity called "risk". Risk would be defined as the probability of a member AS choosing a guard relay by imitating the cluster representative AS that

[2] script2.py

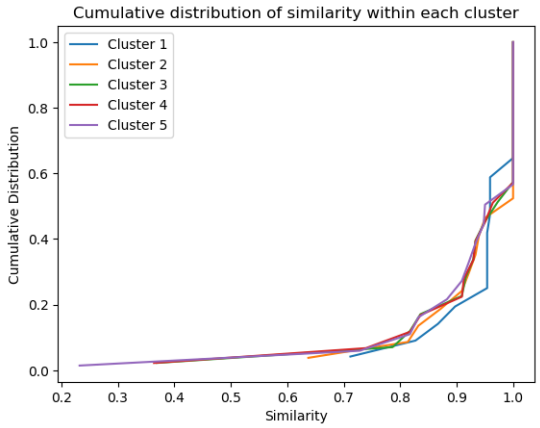

Fig. 2. Figure showing the cumulative distibution of the clusters when k = 5.

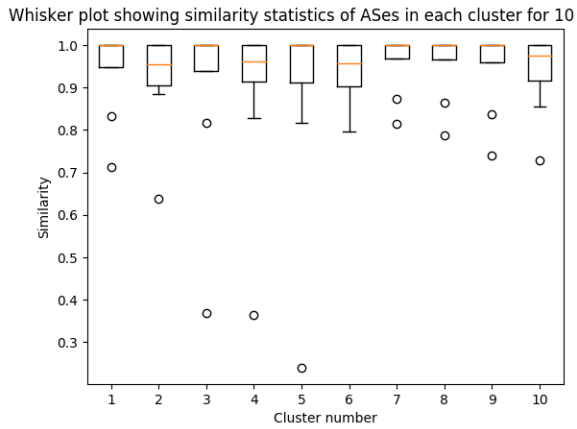

Fig. 3. Figure showing the whisker plot of the clusters when k = 10. The circles show the outliers

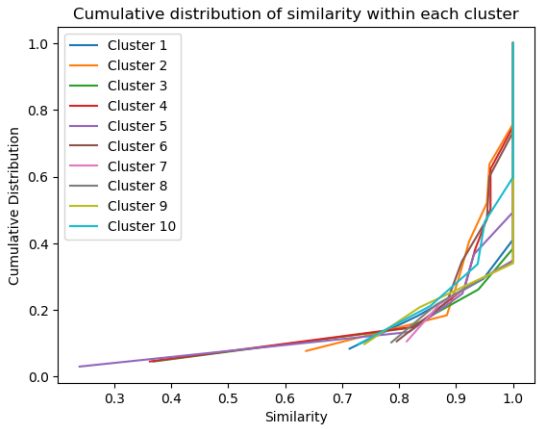

Fig. 4. Figure showing the whisker plot of the clusters when k = 5. The circles show the outliers

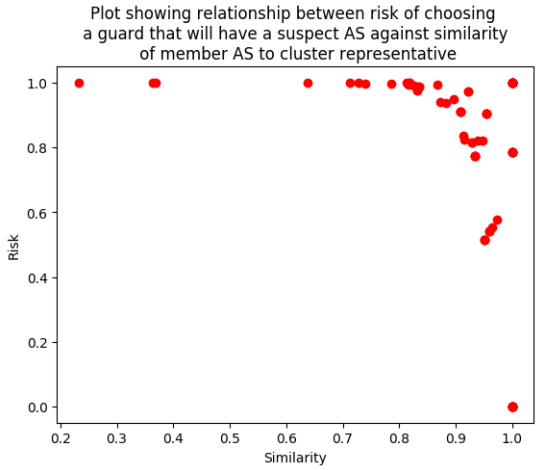

Fig. 5. Figure showing the relationship between risk and similarity (between member ASes and the cluster representative)

would cause Tor traffic to traverse a path with a suspect AS. To calculate this, the following steps were adopted:

1) For each representative AS, guard relays with a non-zero probability of being chosen are identified
2) For each of the member ASes, the path to this guard relay is analysed. If the path contains a suspect AS, the probability from the previous step is added to the risk metric

The risk would be expected to decrease as the similarity between the member AS and its cluster representative increases, because the more similar ASes are, the better the security properties of DeNASA are preserved. We can plot the risk vs similarity relationship in figure 5.

However, figure 5 shows the limitations of naive k-clustering. For a similarity value of 0.95, the risk of choosing a guard relay with a suspect AS on its path is 0.5. This does not preserve the security value of DeNASA, and a a better similarity metric for clustering must be implemented.

References

[1] Paul Syverson, R Dingledine, and N Mathewson. "Tor: The second generation onion router". In: *Usenix Security*. 2004.

[2] Yixin Sun et al. "RAPTOR: Routing Attacks on Privacy in Tor." In: *USENIX Security Symposium*. 2015, pp. 271–286.

[3] Y. Sun et al. "Counter-RAPTOR: Safeguarding Tor Against Active Routing Attacks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. May 2017, pp. 977–992. DOI: 10.1109/SP.2017.34.

[4] M. Akhoondi, C. Yu, and H. V. Madhyastha. "LASTor: A Low-Latency AS-Aware Tor Client". In: *2012 IEEE Symposium on Security and Privacy*. May 2012, pp. 476–490. DOI: 10.1109/SP.2012.35.

[5] Armon Barton and Matthew Wright. "DeNASA: Destination-naive as-awareness in anonymous communications". In: *Proceedings on Privacy Enhancing Technologies* 2016.4 (2016), pp. 356–372.

[6] The Tor Project. *Index of /archive/relay-descriptors/consensuses*. URL: https://collector.torproject.org/archive/relay-descriptors/consensuses/.

[7] Jordan Wright. *How Tor Works Part Three - The Consensus*. URL: https://jordan-wright.com/blog/2015/05/14/how-tor-works-part-three-the-consensus/.

[8] Roger Dingledine et al. "One fast guard for life (or 9 months)". In: *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*. 2014.

[9] The Tor Project. *Tor Source Code*. URL: https://gitweb.torproject.org/tor.git/tree/src/or/entrynodes.c?h=release-0.2.9#n546.

[10] Ryan Wails et al. "Tempest: Temporal Dynamics in Anonymity Systems". In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018), pp. 22–42.

[11] Lixin Gao. "On inferring autonomous system relationships in the Internet". In: *IEEE/ACM Transactions on networking* 9.6 (2001), pp. 733–745.

[12] CAIDA. *Routeviews Prefix to AS mappings Dataset (pfx2as) for IPv4 and IPv6*. URL: https://www.caida.org/data/routing/routeviews-prefix2as.xml.