2016 8th International Conference on Cyber Conflict
Cyber Power
N.Pissanidis, H.Rõigas, M.Veenendaal (Eds.)
2016 © NATO CCD COE Publications, Tallinn

Anonymity Networks and Access to Information During Conflicts: Towards a Distributed Network Organisation

Paolo Palmieri
Department of Computing and Informatics
Bournemouth University
Poole, United Kingdom
ppalmieri@bournemouth.ac.uk

Abstract: Access to information is crucial during conflicts and other critical events such as population uprisings. An increasing number of social interactions happen in the cyberspace, while information exchanges at the infrastructural level (monitoring systems, sensor networks, etc.) are now also based on Internet and wireless links rather than ad hoc, isolated wired networks. However, the nature of the Internet allows powerful hostile actors to block, censor, or redirect communication to and from specific Internet services, through a number of available techniques.

Anonymity networks such as Tor provide a way to circumvent traditional strategies for restricting access to online resources, and make communication harder to trace and identify. Tor, in particular, has been successfully used in past crises to evade censorship and Internet blockades (Egypt in 2011, and Iran in 2012). Anonymity networks can provide essential communication tools during conflicts, allowing information exchanges to be concealed from external observers, anonymised, and made resilient to imposed traffic controls and geographical restrictions. However, the design of networks such as Tor makes them vulnerable to large-scale denial of service attacks, as shown by the DDoS targeted at Tor hidden services in March 2015.

In this paper, we analyse the structural weaknesses of Tor with regard to denial of service attacks, and propose a number of modifications to the structure of the Tor network aimed at improving its resilience to a large coordinated offensive run by a hostile actor in a conflict scenario. In particular, we introduce novel mechanisms that allow relay information to be propagated in a distributed and peer-to-peer manner. This eliminates the need for directory services, and

allows the deployment of Tor-like networks in hostile environments, where centralised control is impossible. The proposed improvements concern the network organisation, but preserve the underlying onion routing mechanism that is at the base of Tor's anonymity.

Keywords: *Tor, anonymous networks, peer-to-peer, denial of service, DDoS*

1. INTRODUCTION

The nature of computer network protocols allows, in principle, a fairly straightforward geographical and organisational mapping of senders and receivers. This can be done both for more restricted local or wireless networks, as well as for the whole Internet. On a large scale, it is thus possible for a government or an Internet service provider to localise, filter, and monitor data streams directed to a specific web service or to a specific geographical region. Several governments effectively control, monitor, or censor Internet traffic, either during crises or permanently. Internet protocols have not been designed for privacy and anonymity, and therefore Internet users can also be easily traced and identified.

This reality has prompted researchers to develop privacy enhancing technologies and anonymity networks, which allow communication to be concealed from external observers, anonymised, and made resilient to control and restrictions. Tor (the 'onion router') is arguably the most successful and widespread anonymity network, counting millions of users [10]. The Tor network is independently developed, and runs on a number of volunteer-operated servers. However, development of the Tor software has been funded by a number of governmental organisations, including US Department of State, DARPA, and the Naval Research Laboratory, as well as the Federal Foreign Office of Germany.[1] This reflects the interest governments around the world have in anonymity networks, which are often seen as both a useful tool and a potential threat [2]. These conflicting sentiments are well exemplified by the discovery in 2007 by security researcher Dan Egerstad that a number of embassies around the world used Tor for delivering private messages, in order not to rely on the hosting country network infrastructure, while the same governments restricted use of Tor by their own citizens [10].

Access to information is critical during conflicts and crises, when controls and restrictions on the flow of information over the Internet are more likely to be imposed [1]. In particular, the ability to use and deploy anonymity networks can be crucial for enabling communication, especially in hostile settings.

A. Onion routing

The Tor anonymity network is built on the concept of *onion routing*. Onion routing was first proposed in 1997 [8], but found widespread use only when implemented in the Tor software in 2002 [5]. The main aim of an onion routing network is to protect users from surveillance and traffic analysis. The identity, location, and network activity of the user are protected by concealing from external observers both the content and routing information of the communication. An onion routing network is therefore a general purpose infrastructure allowing

1 The full list of funders of the Tor project is available on the project's web page: https://www.torproject.org/about/sponsors.html.en [January 4, 2016]

private communications over a public network. It provides anonymous connections that are strongly resistant to both eavesdropping and traffic analysis. This is achieved by relaying the user's traffic, including information about its destination, through virtual *circuits* composed of three successive *relays*. In particular, each relay in the circuit only learns the preceding and following step in the path from the user to the destination: because of this, the user remains anonymous to all relays except the first as well as the destination, while the destination of the communication remains secret to all relays except the last. Messages are also repeatedly encrypted in a layered manner, in order to protect the actual content of the communication: a public key encryption scheme is used to encrypt the communication multiple times, using in inverse order the public keys of all the relays in the circuit. Traffic going back from the destination to the user is similarly encrypted, and routed back from the last relay to the first one.

B. Onion routing during crises

Social media have become an increasingly important mean of communication during crises. In recent years, social media were used extensively in a number of crises, conflicts and uprisings, including the 'April 6 Youth Movement' in Egypt in 2008, the post-election crisis in Iran in 2009, the student protests in Austria in 2009, and the uprisings in Tunisia and Egypt in 2011 and subsequent years part of the larger Arab Spring phenomenon [11]. In all these crises, internet censorship was deployed to prevent access to social media, and anonymity networks played a role in re-enabling access to censored resources, influencing how people and other actors organised online, and ultimately behaved on the streets.

In particular, Tor was used extensively, showing both its strengths and weaknesses. The Tor network, and consequently the onion routing mechanism, proved to be an effective way of circumventing restrictions and internet blockades, while protecting the identity of its users and the secrecy of the communication. However, the open nature of relay servers, which are publicly advertised, make them vulnerable to targeted attacks that can only be partially mitigated by using *bridges* (servers allowing access to the Tor network when direct communication with a relay is impossible). The Tor infrastructure is also limited by the relatively small number of active relays and its semi-centralised structure, which promotes running relays as dedicated servers as opposed to a more distributed, peer-to-peer network organisation [9]. This makes it impossible to use onion routing over local (wireless) networks, which can be potentially deployed on the spot during a crisis using low-cost, low power devices.

Recent events demonstrated that a different implementation of onion routing based on a decentralised network structure might be more suited for crises and conflict areas, where Tor-like networks need to be deployed in hostile environments, and where centralised control is impossible.

C. Onion routing in Wireless Sensor Networks

Another field of application for onion routing is Wireless Sensor Networks (WSN). A WSN is composed of a number of sensors, communicating with each other through a wireless channel, and deployed in an environment where they observe physical phenomena [6]. They are being used in a number of military application scenarios, for purposes including monitoring

and intelligence. WSN are therefore often deployed in hostile or difficult settings, such as battlefields or conflict areas, and are therefore required to be highly fault tolerant, scalable and decentralised. Because of this, WSN are increasingly designed around a distributed network structure and peer-to-peer primitives, to enhance scalability and resilience [7].

Particularly in hostile settings, the security of the communication within the WSN is of paramount importance. Base stations, which are special network nodes that collect data gathered by the other sensors nodes in the WSN, are a central point of failure. It is therefore crucial to make them hard to distinguish from regular nodes. This can be achieved by hiding information on their location and identity (known as context information) within the network [3]. Context information can be protected by employing anonymous routing, and encrypting the communication. In particular, onion routing can be used in a WSN to prevent adversaries from learning the network topology using traffic analysis, and therefore preserve context privacy [4]. However, this requires protocol and mechanisms allowing the deployment of onion routing over the decentralised, peer-to-peer network structures at the base of current WSN.

D. Outline of the paper

The paper is organised as follows. In Section 2.A, we present the challenges of implementing a full onion routing mechanism in distributed networks, and in particular the fact that nodes in a network have limited visibility of the network itself. In Section 2.B, we introduce a novel data structure, called *visibility filter* that enables the sharing of information regarding node visibility across the network in a secure and distributed manner. We present the strategy used to distribute the filters in Section 2.C. Based on the visibility filter structure, we propose an onion routing circuit selection mechanism in Section 2.D. Finally, in Section 2.E we analyse the security of the proposed construction, and in Section 2.F we discuss the communication overhead of the scheme.

2. ONION ROUTING OVER DISTRIBUTED NETWORKS

In the Tor network, clients learn about the currently available relays by downloading the list of running relays from *directory authorities*. Directory authorities are a small subset of relays that collect and distribute routing information in the network [5]. This information is used by the clients when building a circuit, in order to decide which relays to select. Directory information, however, also allows any party (including an attacker) to learn the complete list of relays. The Tor network is based on distributed trust: as the network is open (in the sense that anybody with a sufficiently fast Internet connection can run a relay), it should be hard for a single person or organisation to control large parts of the network. For this reason, directory authorities are selected among long-running, established network nodes. Similarly, the first hop in any circuit built over the Tor network is restricted to being selected among the list of *entry guards* [5]. Both flags (directory authority and entry guard) can be earned by relays after a certain time of continuous operation, proving their stability. This design serves two main purposes: reducing the risk of end-to-end correlation for any given circuit, that is, the chance that both the first and last hop in a circuit are controlled by an adversary; and raising the start-up cost for the adversary.

Without entry guards, the attacker could introduce relays into the network and immediately start having chances to act as first hop. With entry guards, new adversarial relays need to earn the guard flag before they can act as first hop, and the limited number of selected entry guards can prevent attackers from gaining a guard flag for a significant number of relays [12],[13].

While the directory structure and entry guards help protect the privacy of the users, they also expose the Tor network to (distributed) denial of service (DDoS) attacks. As the list of relays and their role is publicly available, an attacker with sufficient resources can target enough relays to entirely disrupt network operation [15], as shown by the large scale DDoS attack targeted at Tor hidden services in March 2015. Worse still, instead of a blanket denial of service attack, an adversary may decide to selectively target relays that are not compromised (or under their control) in order to redirect users to systems one has access to, thus increasing the probability of compromising anonymity [14]. While detection of denial of service attacks is possible [16], and some basic countermeasures such as using client puzzles to mitigate their effect on the network exist [17], DDoS still pose a major threat to Tor, and any other onion routing network based on a similar semi-centralised structure.

A. Challenges of distributed networks

In this paper, we propose a general mechanism for achieving onion routing over a distributed and decentralised network, including network structures organised according to a peer to peer paradigm. Peer to peer (P2P) networks are in fact large decentralised and distributed systems. The design of distributed networks generally follows three main principles: *decentralisation*, in that peers operate without any central coordination; *fault tolerance*, or being able to accommodate nodes joining, leaving, or failing during network operation without a disruption of service; and *scalability*, the property of functioning efficiently for any given number of nodes. Distributed networks are the most efficient and reliable network organisation where network access is limited or restricted. They can allow local nodes to connect to each other, and can be deployed in hostile environments where centralised control is impossible. They are also inherently better suited to coping with denial of service attacks. For this reason, they can be effectively used during conflicts and crises, whether for dedicated networks such as wireless sensor networks or local wireless communication, or overlay networks providing a secure layer over an insecure or openly adversarial network, including the Internet.

However, onion routing cannot be directly implemented over distributed and peer to peer networks without modification. In fact, onion routing makes two important assumptions about the organisation of the underlying network: first, it is assumed that all relays are able to communicate with each other; and second, the Tor directory structure requires that a list of all relays active across the network can be created and maintained. Neither of those assumptions can be satisfied in a distributed network. In a peer to peer setting, nodes do not generally have a full view of the network; that is, they only know a subset of other nodes in the network, called *neighbours*. In fact, nodes are often unable to connect to most other nodes due, for instance, to NATs, firewalls, or, in the case of wireless networks, signal reach). The reduced network visibility means that no node or set of nodes can create with certainty a list of all nodes participating in the network at any given time. This prevents the creation of a directory

structure, informing nodes of the potential relays with which to create a circuit. However, while in Tor the roles of client and relay are generally distinct, it is possible to assume that all nodes in the distributed networks can act as relay. In this setting, nodes can create circuits where the first relay is one of their neighbours. However, this poses a second problem: how can we select following relays, considering that the client node has no knowledge of the network outside its neighbours? This is especially important, as we want to avoid circuits that are too local – that is, that are entirely comprised of neighbours of the client node – in order to avoid partitioning the network. At the same time, we cannot trust relays to select the following steps in the circuit, as doing that would mean that a compromised first hop would be able to influence the creation of the whole circuit, and include only relays under the control of the attacker, thus breaking the user's security.

In order to address this issue, we introduce a mechanism based on controlled flooding and a Bloom filter based data structure, which allows the distribution of relay information among nodes (see Section 2.B). The Bloom filter structure, which we call *visibility filter*, stores information on the neighbours of each relay (and therefore node) in a privacy-preserving way. This information is spread across the local portion of the network through a depth-limited flooding mechanism, where each node transmits its own filter to the neighbours, and at the same time relays the filters received from the neighbours for a limited number of hops, corresponding to the expected length of the circuit (generally 3). The visibility filter structure addresses both assumptions necessary to achieve onion routing: first, it allows relays to learn which other relays they can build a circuit with, without having to learn the relay identities; and second, it supersedes the directory structure and therefore the necessity to compile a list of all relays. The mechanism is introduced in the following section.

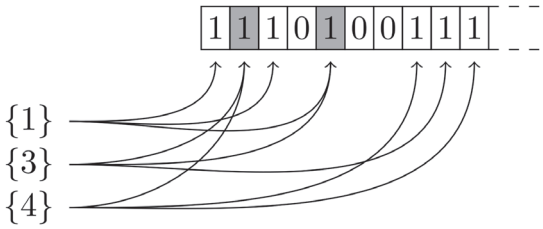
Several attempts have been made to combine mix-net based anonymity networks and P2P. Morphix [22], proposed in 2002, is a peer-to-peer based anonymity network that provides a collusion detection mechanism against an adversary running multiple nodes. However, since the mechanism is based solely on a node's local knowledge, the collusion detection mechanism can be broken, rendering the network insecure [23]. ShadowWalker, proposed in 2009, is a P2P anonymous communication system that aims at being low-latency in nature [24]. However, ShadowWalker circuits are constructed following a traditional random walk strategy that does not address the problem of making sure that not all nodes are in close proximity to the originating peer. Other anonymous routing protocols have been designed for specific P2P network topologies. Salsa [21], NISAN [25] and Torks [26] are all based on the common Distributed Hash Table (DHT) topology. However, the lack of anonymity in their lookup mechanisms may enable an adversary to infer the path structure [27]. In this paper, we propose a solution for implementing onion routing on a peer to peer network, independently of the network topology and structure. Our strategy doesn't require risky network lookups, and ensures that the circuits are not local to the originating node.

B. Bloom filters and visibility filters

A Bloom filter (BF) is a space-efficient data structure representing a set [18]. A BF generated for a set allows the determination, without knowledge of the set itself, of whether an element is in

the set or not, with a probability of false positives p. A Bloom filter can be represented as a binary string of length n, initially all set to 0, and a set of hash functions whose outputs are uniformly chosen in $\{1, \ldots, n\}$. During the creation of the filter, all the elements in the originating set are given as input to each hash function recursively, and bits in the filter corresponding to the output of each hash are set to 1; that is, if for instance one of the hash functions returns the value 5 for an element of the set, the 5th bit of the Bloom filter string is set to 1. A Bloom filter can be queried in the same way; we determine whether an element is part of the originating set by passing its value to the hash functions and reading the bits corresponding to their outputs. If one or more bits have value 0, the element is not part of the set. If instead all bits have value 1, the element is part of the set, minus a false positive probability p. A false positive happens when all the values have been set to 1 during the filter creation by other elements (an event called collision), and not by the element of the query.

FIGURE 1: BLOOM FILTER CONSTRUCTION. IN THE PICTURE, THREE ELEMENTS {1,3,4} ARE ENCODED IN THE FILTER. THE FILTER HAS LENGTH n=10, AND THREE HASH FUNCTIONS ARE USED



We can construct a Bloom filter for each node, containing information on the neighbours of the node itself. The filter can be then used by a third node (that is, a node that is not a neighbour of the first node) to determine whether there is a 'route' between the two nodes: in other words, whether there is a node that is a neighbour of both nodes, allowing them to communicate. This filter, which we call *visibility filter*, can be used when building an onion routing circuit in order to verify that all hops in the circuits are able to communicate with the previous and next hop.

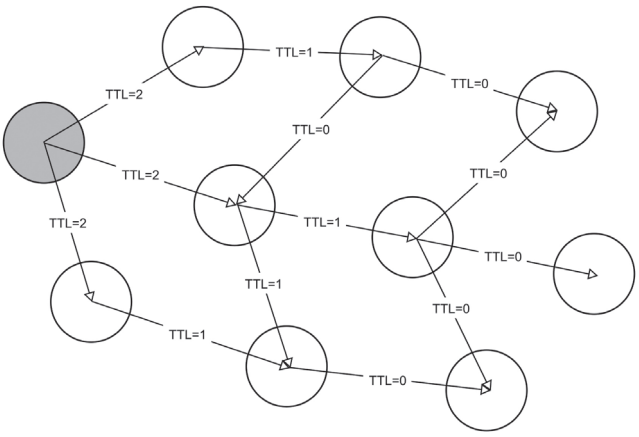
In practice, we assume that each node possesses an identifier unique over the network. The filter is then built over the set of all identifiers corresponding to the neighbours of the node for which the filter is being built. A node can verify the existence of a common neighbour with a node for which it has the visibility filter following a two-steps procedure. First, the node performs a XOR operation over the received filter and its own. If the resulting intersection filter has less than n bits of value 1, then no common neighbour is possible for the properties of Bloom filters. If the number of bits with value 1 is greater or equal to n, then the node proceeds to test each of its neighbours against the other node's filter. Neighbours satisfying the filter are common neighbours, minus a false positive probability p.

C. Distribution of visibility filters

In order to maintain their effectiveness in a constantly changing network, visibility filters need to be recreated and distributed at regular time intervals. We propose to distribute filters across the network using a controlled network flooding mechanism. Following this strategy, each node transmits its own visibility filter to all its neighbours. The identifier of the node is appended to the filter to allow identification of the filter origin (to guarantee integrity and prevent filter forging, the filter can be also signed using the node key pair), as well as a counter flag, called TTL (Time To Live) with value equal to the number of relays in an onion circuit (which defaults to 3). The neighbours will decrease the TTL flag by one, and further forward it to their own neighbours. This process is repeated by all nodes receiving the filter until the TTL reaches 0 (see Figure 2).

While this process involves a communication overhead on the network for filter distribution, it is important to notice that each node benefits from learning as many filters (and therefore potential relays) as possible, in order to enhance the variety of its relay list and therefore mitigate risks posed by adversarial relays. At the same time, limiting the distribution of the filters to the nodes within reach for circuit building reduces the network overhead. We discuss the network overhead in detail in Section 2.F.

FIGURE 2: A SCHEMATIC OF THE LIMITED FLOODING MECHANISM USED FOR DISTRIBUTING THE VISIBILITY FILTER. THE TTL VALUE LIMITS THE DISTRIBUTION OF THE FILTER ONLY TO THOSE NODES THAT CAN BE INCLUDED IN A CIRCUIT



D. Onion circuit creation

Once a node obtains a sufficient number of filters, it can start creating onion circuits for communication. A circuit is created using a trial and error strategy. The ‘client’ is the node (either a user’s device or an autonomous system such as a sensor) which wants to communicate in a secure and private manner; it is, in fact, a client connecting to the service provided by the onion routing network. Communication originating from the client should reach an intended

destination over an onion circuit, of which the client knows the identifier and visibility filter. We assume that the destination is a node of the network. The circuit is built as follows:

1. The client selects a first potential relay for the circuit among its neighbours.
2. Then, the client selects all filters received from neighbours other than the selected first relay. This ensures that the circuit will not be one artificially suggested by the relay itself through manipulation of the filter distribution.
3. Among the eligible filters, the client selects those that satisfy the identifier of the destination node as potential last relays. This step ensures that the eventual last relay will be able to communicate with the destination.
4. The client selects a random relay among the potential last relays. Then, it calculates the intersection between the filters of the potential first and last relay (using a XOR operation): if the number of resulting bits with value 1 is greater than n, a common neighbour exists (minus false probability p). Otherwise, the client selects a different relay and repeats the last step of the process. In practice, this step ensures that the first and last relay will be both able to communicate with at least one common third relay, which will be the middle relay of the circuit.
5. Once two compatible first and last relays are found, the client instructs the first relay to try to build a circuit by sending to it the intersection filter that will be used to identify the common middle relay.
6. The first relay builds the first part of the circuit by connecting to one of its neighbours which satisfies the received intersection filter (the middle relay).
7. Then, the client uses the incomplete circuit to communicate with the middle relay, and instructs it to complete the construction of the circuit by connecting to the last relay.

The circuit creation process enables nodes in a distributed network to reliably build onion circuits without relying on a centralised directory structure. This is made possible by the combined used of visibility filters and a decentralised mechanism for the distribution of relay information.

E. Sybil attacks and security considerations

In general, all distributed and peer to peer networks are vulnerable to Sybil attacks, where the adversary generates malicious nodes in the network in such a way that the target node in most network communication is in some way dependent on them [19]. Sybil attacks are the computer network equivalent of a siege; attack targets are generally surrounded by malicious entities, with the notable difference that there is generally no way to distinguish malicious nodes from honest ones, thus making detection of Sybil attacks more difficult. The general strategy for mitigating the effects of a Sybil attack is to limit the reliance of nodes on their neighbour for communication; if the attacker needs to control or deploy nodes across the whole network, the cost of the attack consequently increases [20].

The circuit creation protocol we propose in this paper achieves reduced locality of the circuit, thanks to the selection of potential relays; neighbours are excluded from acting as second

or third relay, and third relays are selected from those whose information was not originally received from the selected first relay (thus preventing the creation of circuits influenced by a single node). While perfect security against Sybil attacks is generally impossible to achieve, these measures mitigate the impact of such attacks, and therefore increase the privacy and security of the user.

The circuit building mechanism also uses an intelligent selection strategy for relays following the first in order to minimise the impact of a malicious first relay. In fact, at step 2, the client excludes from the set of potential second relays those nodes whose filter was received from the selected first relay. This ensures that a malicious first relay will not be able to influence the selection of the following nodes in the circuit.

F. Performance considerations

In terms of network performance and overhead, the main deviation of the proposed scheme from the classical onion routing implementation is the additional requirement of distributing the visibility filters among nodes. In this Section, we describe why the limited flooding strategy we propose is realistic and introduces only limited overhead.

In general, peer-to-peer networks adopt different strategies for distributing or searching for information among peers. The concept of *flooding* was introduced by the Gnutella 2000 search protocol [28]. In practice, a node looking for a specific resource over the network broadcasts its query to its neighbours. If a neighbour does not have the resource, it forwards the query to its own neighbours. This is repeated until the resource is found, or all the nodes have been contacted. This naïve approach scales very poorly in large networks. For this reason, several alternative approaches have been proposed and implemented in subsequent networks that modify the flooding behaviour [29]. The main issue with network flooding is the high network use it can generate. This, combined with the uncertainty of the timing of queries (especially when user generated) can result in a significant overhead. In the proposed scheme, we address this by adopting a smart flooding strategy that limits the impact of the visibility filter distribution over the network. In particular, we limit the depth of the flooding (that is, the number of times a filter is relayed). This restricts the communication to a small portion of the network, and consequently greatly reduces network use. We can safely do so because we know exactly how far the information should be transmitted: as relays will only communicate with neighbouring relays, the distance coincides with the length of a circuit. At the same time, we can control the timing of the flooding, thus preventing network overload. In fact, we can define regular interval at which the nodes in the network should transmit their filters, and can decide to limit this further by only transmitting filters if there is a significant change in the neighbour's set. New nodes entering the network can request a cached set of filters from their neighbouring nodes, which can easily keep them until a new transmission due to the very limited space requirements of Bloom filters.

We can estimate the network use of the limited flooding strategy. If we assume that each node has 100 neighbours, the size of the filer will be 839 bytes (for a false positive probability of less than one in a million, or e^{-14}). If the percentage of common neighbours between two

nodes is 20%, the overall network use for a single node participating in the distribution of the filter can be estimated to approximately 83KB. Please note, this is the size of the information being transferred (the payload): the actual network use depends on implementation details (and in particular, on the chosen communication protocol and its parameters). This overhead is perfectly compatible even with networks with limited bandwidth, such as WSNs.

3. CONCLUSIONS

In this paper, we discussed how a distributed network organisation could overcome many of the limitations and security challenges posed to networks deployed in difficult or hostile settings, such as during crises and conflicts. We analysed how onion routing can be used to secure communication under those circumstances. We also presented the limitations that the original design of onion routing and its most common implementation, Tor, have when used on distributed networks. In order to address these shortcomings, we proposed a number of modifications to the onion routing mechanism aimed at improving its resilience to a large coordinated offensive run by a hostile actor in a conflict scenario, such as a denial of service attack. In particular, we introduce novel mechanisms that allow relay information to be propagated across the network without requiring a directory structure, and we address the issue of the limited visibility among nodes of distributed networks. These results open the way to the deployment of onion-routing-enabled networks in hostile environments, where centralised control is impossible.

A natural direction for future research would be an implementation of the proposed scheme, to provide experimental results on the overhead incurred by the distributed network for the exchange of the relay visibility information.

ACKNOWLEDGMENTS

The author would like to thank Johan Pouwelse for interesting discussions on the topic of the paper, as well as for previous collaborations on peer-to-peer networks.

REFERENCES

[1] Mina Rady, Nazli Choucri. ‘Anonymity Networks: New Platforms for Conflict and Contention’. Available: http://web.mit.edu/minarady/www/papers/Tor Conflict -Mina Rady-v3.0 single .pdf (January 4, 2016).

[2] Emin Çalışkan, Tomáš Minárik, Anna-Maria Osula. ‘Technical and Legal Overview of the Tor Anonymity Network’. NATO Cooperative Cyber Defence Centre of Excellence. Available: https://ccdcoe.org/sites/default/files/multimedia/pdf/TOR_Anonymity_Network.pdf (January 4, 2016).

[3] Li, N., Zhang, N., Das, S.K., Thuraisingham, B.M. ‘Privacy preservation in wireless sensor networks: A state-of-the-art survey’. Ad Hoc Networks, vol. 7(8), pp.1501-1514, 2009.

[4] Paolo Palmieri. ‘Preserving Context Privacy in Distributed Hash Table Wireless Sensor Networks.’ In Information and Communications Security - 17th International Conference (ICICS 2015), Revised Selected Papers. Lecture Notes in Computer Science, Springer, vol. 9543, pp.436-444, 2016.

[5] Roger Dingledine, Nick Mathewson, Paul Syverson. ‘Tor: The Second-Generation Onion Router’. In Proceedings of the 13th Usenix Security Symposium (Usenix 2004), pp.303-320, 2004.

[6] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E. 'Wireless sensor networks: a survey'. Computer Networks, vol. 38(4), pp.393-422, 2002.
[7] McGoldrick, C., Clear, M., Carbajo, R.S., Fritsche, K., Huggard, M.. 'Tiny Torrents: Integrating peer-to-peer and wireless sensor networks'. In the Proceedings of the Sixth International Conference on Wireless On-Demand Network Systems and Services (WONS'09), IEEE Press, pp.109-116, 2009.
[8] Syverson, P.F., Goldschlag, D.M., Reed, M.G. 'Anonymous connections and onion routing.' In the Proceedings of the 1997 IEEE Symposium on Security and Privacy, IEEE Computer Society, pp.44-54. 1997.
[9] Paolo Palmieri, Johan Pouwelse. 'Key Management for Onion Routing in a True Peer to Peer Setting.' In Advances in Information and Computer Security - 9th International Workshop on Security (IWSEC 2014), Proceedings. Lecture Notes in Computer Science, vol. 8639, pp.62-71, Springer, 2014.
[10] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, Douglas Sicker. 'Shining Light in Dark Places: Understanding the Tor Network.' In the Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008). pp 63-76, 2008.
[11] Stefan Csizmazia. 'The Role of Online Social Networks in Political Uprisings.' Available: http://www.steviec.at/uni/bakk/csizmazia__role_of_osn_in_political_uprisings.pdf (January 4, 2016).
[12] Karsten Loesing. 'Measuring the Tor Network from Public Directory Information.' In the Proceedings of the 2nd Hot Topics in Privacy Enhancing Technologies (HotPets 2009), 2009.
[13] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, Douglas Sicker. 'Low-Resource Routing Attacks Against Tor.' In the Proceedings of the 2007 ACM workshop on Privacy in electronic society (WPES 2007), ACM, pp 11-20, 2007.
[14] Nikita Borisov, George Danezis, Prateek Mittal, Parisa Tabris. 'Denial of service or denial of security?' In Proceedings of the 14th ACM conference on Computer and communications security (CCS '07), ACM, pp.92-102.
[15] Norman Danner, Sam Defabbia-Kane, Danny Krisanc, Marc Liberatore. 'Effectiveness and detection of denial-of-service attacks in Tor'. ACM Transactions on Information and System Security, vol. 15(3), article 11, 25 pages, November 2012.
[16] Norman Danner, Danny Krisanc, Marc Liberatore. 'Detecting Denial of Service Attacks in Tor'. In the Proceedings of Financial Cryptography and Data Security, 13th International Conference (FC 2009), Lecture Notes in Computer Science, vol. 5628, pp.273-284, Springer, 2009.
[17] Fraser, N.A.; Kelly, D.J.; Raines, R.A.; Baldwin, R.O.; Mullins, B.E., 'Using Client Puzzles to Mitigate Distributed Denial of Service Attacks in the Tor Anonymous Routing Environment'. In the Proceedings of the 2007 IEEE International Conference on Communications (ICC 2007), pp.1197-1202, 2007.
[18] B.H. Bloom. 'Space/time trade-offs in hash coding with allowable errors'. In Communications to the ACM, vol 13(7), pp.422-426, 1970.
[19] John R. Douceur. 'The Sybil Attack'. In the Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS), pp.251-260, 2002.
[20] Rowaihy, H.; Enck, W.; McDaniel, P.; La Porta, T., 'Limiting Sybil Attacks in Structured P2P Networks.' In the Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007). IEEE, pp.2596-2600, 2007.
[21] Arjun Nambiar, Matthew Wright. 'Salsa: a structured approach to large-scale anonymity'. In Proceedings of the 13th ACM conference on Computer and communications security (CCS '06), ACM, 2006.
[22] Marc Rennhard, Bernhard Plattner. 'Introducing MorphMix: peer-to-peer based anonymous Internet use with collusion detection.' In Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society (WPES '02), ACM, 2002.
[23] Parisa Tabris, Nikita Borisov. 'Breaking the Collusion Detection Mechanism of MorphMix'. In Proceedings of the 6th International Workshop on Privacy Enhancing Technologies (PET 2006), Lecture Notes in Computer Science, vol. 4258, pp.368-383, Springer, 2006.
[24] Prateek Mittal, Nikita Borisov. 'ShadowWalker: peer-to-peer anonymous communication using redundant structured topologies'. In Proceedings of the 16th ACM conference on Computer and communications security (CCS '09), ACM, 2009.
[25] Andriy Panchenko, Stefan Richter, Arne Rache. 'NISAN: network information service for anonymisation networks.' In Proceedings of the 16th ACM conference on Computer and communications security (CCS '09), ACM, 2009.
[26] Jon McLachlan, Andrew Tran, Nicholas Hopper, Yongdae Kim. 'Scalable onion routing with torsk', In Proceedings of the 16th ACM conference on Computer and communications security (CCS '09). ACM, 2009.

[27] Qiyan Wang, Prateek Mittal, and Nikita Borisov. ‘In search of an anonymous and secure lookup: attacks on structured peer-to-peer anonymous communication systems’, In Proceedings of the 17th ACM conference on Computer and communications security (CCS ‘10), ACM, 2010.
[28] M. Ripeanu, ‘Peer-to-peer architecture case study: Gnutella network.’ In Proceedings of the First International Conference on Peer-to-Peer Computing, 2001, pp.99–100, 2001.
[29] Niels Zeilemaker, Zekeriya Erkin, Paolo Palmieri, Johan A. Pouwelse. ‘Building a privacy-preserving semantic overlay for Peer-to-Peer networks’. In 2013 IEEE International Workshop on Information Forensics and Security (WIFS 2013), pp.79-84, IEEE, 2013.