# Anonymous Communications : A survey on I2P

Gildas Nya Tchabe and Yinhua Xu

Technische University of Darmstadt, Theoretische Informatik ,Kryptographie and Computeralgebra Cryptography, Privacy and Security
Karolinenplatz 5, 64289 Darmstadt, Germany
https://www.cdc.informatik.tu-darmstadt.de/cdc/

**Abstract.** Since some latest revelation from some secret organizations, the interest in anonymous communication have been gaining more and more attention from internet users as privacy and anonymity problem have come out. to achieve online anonymity, several software or platform such as Invisible Internet project(I2P), Tor, and many others have been developed to help internet users to protect their anonymity. In this paper we provide a brief overview of I2P, then gives some weaknesses and advantages and finally gives a brief comparison between I2P and Tor network.

## 1 Introduction

The rapid growth of the Internet applications has made communication privacy an increasingly important security requirement. Although encryption aims at preserving some modification of data, it is still possible for the adversaries to get significant information about the traffic carried on the network packets and the physical entities, such as the network addresses of the sender and the receiver of the message.

One of the main problem is the exposure of the network address which might result in severe consequences: Adversaries can easily overhear all the messages and perform traffic analysis; even if the communication content is encrypted, routing information is still sent in the clear because routers need packets destinations in order to route them in the right direction.

Several platforms were developed to help increase anonymity in Internet, as an example, we have Tor network which is one of the first solution to provide anonymous communication[1]. But Tor network later came up to have some limitations in a sense that it is built on a centralized system. To resolve those constraints, developers have came up with a distributed alternative for file-sharing which was then followed by the appearance of P2P networking. And then later followed by a new project called the Invisible Internet Project (I2P)[2].

I2P has been developed by a group of anonymous people and supporters, whereby the main developer and person responsible for this project is known by the nickname jrandom. The I2P developers' concept was to implement a great and unique idea for distributed P2P anonymous systems, which provide its users a better anonymity and security[3].

## 2 Description of I2P Network

I2P is described as a self-organizing network, providing better scalability, and has a resilient packet switched anonymous network overlay over which a certain number of anonymity and security can operate. I2P is based on what is called I2P-router. The router is a virtual software that runs on the host and provides a bridge to the local application. I2P application has two types of capability. It could either access the darknet services and playing the role of client, or access the host services and playing the role of the server. Both approaches are linked together using a peer-to-peer approach(completely decentralized). This link runs as overlay on top of IP. I2P to be able to exchanged messages on internet, needs protocols as well. There are two kind of protocols which I2P is able to use : either a TCP(NTCP) or UDP protocol(SSU) (see figure 1). The connection to the tunnels are mapped by the router[4].

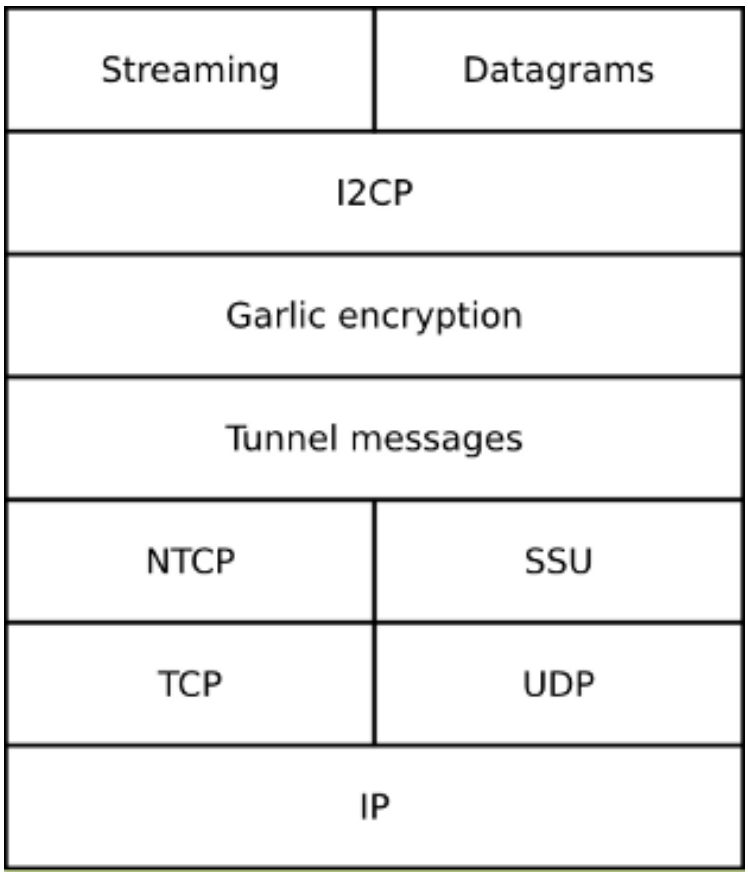


figure 1 : layer overview[12]

As many other software that are designed to protect user's anonymity, I2P also allows the implementation of an additional layer of encryption model and

a particular routing system known as " Garlic routing". Garlic routing is quite identical to Tor's onion routing. The difference with onion routing is that Garlic routing extends onion by bundling different messages together. All informations that are transmitted through the tunnels are fully encrypted and only the receiver's router is able to decrypt the message.

### 2.1 I2P Components

The fundamental components in I2P network are: router, tunnels, NetDB.

**Router**: As mentioned before, routers are just simple software that participate in the network.

**Tunnel**: They are unidirectional path through several routers, which means that the sending path and the receiving path are different. (example for bob and alice want to communicate via I2P, they actually require 4 tunnels). After a certain amount of time, tunnels get expired. Tunnels are checked every time to remove failing tunnels. The default lifetime is set to 10 minutes. The length of a tunnel hop usually varies. Tunnels could be exploratory or client. there are two type of tunnels (inbound and outbound tunnels) see figure 2 that is proposed by I2P[5].

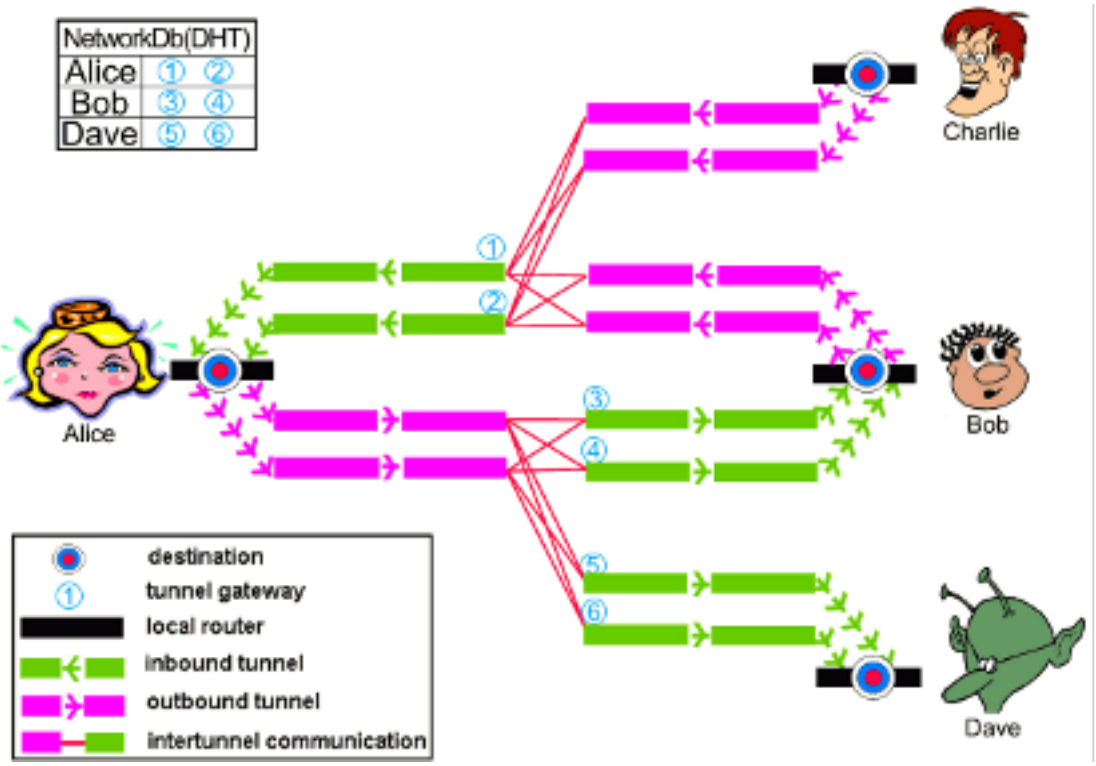


figure 2 : I2P tunnels overview

Inbound and outbound tunnels are automatically built when I2P is started. It is also important to notice that connections to tunnels are only valid for nodes(systems) over which I2P has an installed paths. As shown in figure 2, the first tunnel is a tunnel gateway and the last tunnel is the the tunnel endpoint.

**NetDB** : One of the key concept of I2P is the network database called NetDB. it is composed of a pair of algorithms(DHT Kademlia and floodfill) that make the sharing of metadata possible[13].

### 2.2 I2P routing

I2P practically faces the same problem as in some other peers-to-peers systems, which is to find available peers and the services offered by those peers. In I2P network, peers can only be identified by a data structure which is known as routerInfo. RouterInfo contains all the necessary information about the available peers. It also contains the encryption information such as 2048-Bit Elgamal encryption key, 1024-Bit DSA, AES 256/CBC Mode, MAC, (HMAC-MD5-128) and valid certificates as well. I2P to discover others peers, use a non-anonymous HTTP to download list of routerInfo of available peers from a fixed location.

I2P's DHT: After getting a list of peers, I2P uses a super-peer DHT to build the NetDB. the NetDB contains all informations about the peers and services that are available in the network. The super-peer in the DHT are peers having more responsibilities(exploratory) than the other peers. One of their main responsibility is to maintain the databases. There are called floodfill. Floodfill peers is also responsible for getting the informations about closest ID. In order to get the information the distance has to be calculated. It is calculated by using Kademlia's XOR distance metric[8]. For a peer to become a floodfill, some conditions have to be respected:

- Have a sufficient bandwidth
- Check its configuration

After completing those steps, the peer seeking to become a floodfill also has to check whether the active floodfill drops below a certain threshold before promoting itself to floodfill peers.

Information about how to contact peers are kept in the so-called leaseSet. The leaseSet are stored in the same NetDB that also contains routerInfos. But the LeaseSet and the routerInfo are completely independent entities. They are just stored in the same database. A leaseSet main job is to specify a set of entry points(called leases) to the service. As mentioned before the entry point of a tunnel is the peer currently serving as inbound gateway to the service[6,7].

For the LeaseSet and the RouterInfo to be stored in the NetDB, a request has to be sent to super-peer. After the floodfill peer receives the request, it forwards the request to seven other super-peers and then sends the confirmation to the initiator. To retrieve data from the NetDB, first we have to mention that the retrieving of Data into the NetDB are performed via tunnels. The retrieving procedure is done this way: firstly the request is transmitted to the (with respect to the destination address) two super-peers ( both super-peers have to been known by the requester). In case a floodfill does not have the

requested Information, a list of other close floodfill is sent back. The requesting peers will continue to query others floodfills peers until all known floodfills have been contacted. Then the floodfill sends the response to the initiator through the inbound tunnels[15] .

## 3 Thread Models and I2Ps Weaknesses

Although a detailed introduction has been expanded about the property of protecting users anonymity, it is still a hard question to answer, whether I2P satisfies our particular or special demands on anonymity appropriately. The website[16] has mentioned the 18 thread models which will lead to various results and we will find the most effective ways from them to show the shortcomings of I2P. In reference to [18], we will unfold the DoS and deanonymization attacks on the basis of [16].

### 3.1 Brute Force Attack

The firstly considered attack always goes in a brute force way, in which the attacker tries to observe all data streams and associate the possible sending paths with possible endpoints. If such a try against I2P is trivial, the attacker will be confused by the behaviour of frequent data transmission by all peers of the network and the changing size of content-varying traffics. In addition, it is difficult for the external observer to access the messages because of encoded and streamed communication among the intern routers. However, an attacker, with plentiful resource and also occupying a large ISP (Internet Service Provider) or an Internet exchange point, concentrates a huge amount of data, like 10 GB, on one inbound endpoint in a short period and excludes those peers not receiving the 10 GB data, which guides the attacker to find possible routes to the real destination.

In order to fight against the attack, one will on one hand set lower bandwidth threshold, nontrivial latency and restricted routes (not yet implemented) as some countermeasures, but at the same time the quality of service cannot be guaranteed. On the other hand one can think about unpublished or encrypted leaseSets for eepsites or just handle a peer selection algorithm with high reliability.

### 3.2 Intersection Attack

We give two assumptions, one is that the attacker takes control of both ends of the tunnel simultaneously, and the other that the size of the network is not large and expands only slightly, so that the monitor can regularly go in touch with his quarries and keep track of as many peers of the network as possible. Over quite a long period, during which node churn happens, the exact location of the quarry will be indirectly exposed by doing the intersection of the routes where data are successfully passing, i.e. ignoring the non-participants in contact with the target and narrowing down the number of candidate routes.

It is said in [17] such an attack would be more effective in combination with other types of attacks as Sybil attacks, which we will introduce later, and timing attacks. If the attack do not launch a combinatorial attack mentioned above, some succeeding steps are taken against the attack: as a partial defense, ordering the peers in a strict plan, altering the peer profiling and selection in the network in slow pace, decreasing the number of tunnels via each peer and supporting hosting on multiple routers at one time; for further fixing of the vulnerability, abandoning guard nodes with low bandwidth, traffic shifting among tunnels, tunnels with a short life cycle, configurable tunnel lengths will come to help as well.

### 3.3 Partitioning Attack

Just as the name implies, in a partitioning attack the peers of a network are to be divided into separate smaller zones. Here the partitioning attack is classified into two categories: the technical and the analytical one. In regard to the technical partitioning, the connections between peers are cut to fragment network into pieces with the help of the information I2Ps built-in network database, which stores peers statistics and records and hence decides whether an existent link to other sections will be utilized to heal the disrupted network. Under the terrible assumption that the attacker disconnects all contacts with peers, and as a result the target zone will be left alone, the network database will not help fix it, because the zone is unreachable in accordance with routing-table. The best choice for the router in this case is to alert the user that all connections are temporarily lost, if it can eventually notice the disappearance of a gigantic number of previously available peers. Of course, the network size is a key factor in protecting the anonymity of the target.

Another powerful kind of partitioning is characterized to be analytical. Under this circumstances one records different behaviours of routers and endpoints, and segregates them in terms of their behaviour, i.e. an attacker gets hold of a list of peers inside the I2P network by planting a peer in it and then harvesting information about the unknown neighbours, so that it will get well acquainted with the exact number of inbound tunnels extended to the target, and segment the peers in the light of criterion, e.g. number of tunnels. An attacker has to remember: before the trick he needs to make preparations, like booting its own routers into a separate network after taking over a significant part of the network, or cheating the developers to trust it.

### 3.4 Denial of Service Attack

Like brute force attack, the DoS will also produce a large amount of messages, but quite differently, the DoS degenerates or even terminates the network service in an abnormal way instead of identifying one single peer. In one case, the attacker overloads the peer processors by flooding a tremendous number of requests for some cryptographically complex operations. Attempting to mitigate the trouble, only one with good-formed network engineering skills can echo the

expensive requirements, nevertheless, the attacker will exploit some bugs in the implementation of countermeasures all the same. The bugs of the applications will not be discussed in this paper.

**Floodfill Takeover** One of the most efficient DoS attack is floodfill takeover, where an attacker creates multiple fraud identities as bases to control the total network. In fact, the participants will estimate the performance of the known peers and measure them in connection. It turns out definitely, that multiple identities on the same host limits the performance of each malicious peer, in another word, no sufficient resource will be distributed to those additionally created identities. Nonetheless, the attacker still has a opportunity to makes the participation of its peers in the NetDB possible by following the next instructions.

The availability of resources is weighted in regard to two available data rate, statically and administratively configured for peers, and job lag, the average delay between continuous scheduled tasks during operation. The attacker can generate more scheduled assignments to deploy the job lag, which is significantly dependent on the number of open tasks. For the network load changes and routers are possible to be rebooted from time to time, the attacker would like to wait for the decrement of the legal floodfill participants number so as to make additional malicious identities alive in the network, especially when a churn happens.

Furthermore, the attacker can even lengthen the job lag in cooperation with DoS attacks against a legal floodfill participant to raise update frequency of floodfill nodes. He not only turns many legitimate nodes down, but builds many new tunnels through them as well to order more searching jobs, which can overload the victims in the tunnel and build up a job lag. It should be noticed that the emission of large amounts of data through the victim will be avoided by the data limitation mechanism, which forces the victim to drop job requests and lower its burden, and a DoS attack is launched on one victim at a time only after the last victim quits from the floodfill set.

**Sybil Attack** A Sybil attack is doubtlessly an evolutional version from floodfill takeover, which allows the attacker to predominate just a limited part of the keyspace, consuming fewer resources than a complete takeover. Despite that the attacks on multiple I2P nodes cannot be performed in parallel due to the peer profiling, it is still possible to take down a large quantity of healthy identities and activate the malicious ones around the victim. In this condition an attacker will completely take over all intended positions in the keyspace within a couple of minutes.

It takes a long setup time, about an hour, for a new node to get access into the network, before the Sybil attack can really be in effe. The mapping from leaseSets and routerInfos to netDB keys, in which the attacker is actually interested, contains the current date with respect to UTC. As a result the keyspace is not kept stable, which results in that the attacking nodes have to hold flexible

positions in the keyspace to prevent to conceal themselves, whereas the substitute attacking nodes will be in right position thanks to the predictable behaviour and no random nonce before a new day begins and the attack will be run with merely few additional resources. In [10] the former attacks are described on occasion of full keyspace, and later attacks of partial keyspace.

**Eclipse Attack** The final goal of this attack is to make netDB unavailable to legitimate net- work participants like other DoS attacks. The attacker should isolate the victim to prevent it to find a key in the database by holding all its neighbours, otherwise a lookup-failing message from an uncontrolled neighbor will tell the victim that the attacker is intruding. If the attacker builds up a firewall in this region, he blocks all the access requests to items and responds to them with the answer "information not found . Gradually the legal ones will not be served and even not have no interaction with the nodes which should be visible to them.

### 3.5 Deanonymization Attack

This attack allows the attacker to correlate the service a legitimate participant uses to his IP address. For the sake of successful achievements, we will arrange malicious nodes in the netDB, as we have described earlier in the last section, to observe the relationship of the events in the network.

As peers store data they gather on the closest known floodfill node, they often verify the proper storage of these information by operating a lookup in other close floodfill nodes. If both storage node and verification node are controlled by the same person, the manipulator will set up connection between them and monitor both communications. For the peer information is stored without a tunnel, the peer will be exposed by the content of the netDB entry for all aspects; for the verification can be figured out through an exploratory tunnel, the tunnel endpoint will be exposed anyway. The attacker can take advantage of the combination of both to create a logical mapping from exploratory tunnel endpoints to the tunnel owners and for sure effectively deanonymizes the user. If service information expires some time later, every user has right to fetch it before starting a communication with a service and updates it regularly during the communication. Then the attacker gains the opportunity to identify which of the monitored peers communicate with each of the observed resources and its precise time point. The regular update of service information additionally reveals how long the service can be enjoyed by a certain peer. In this way the attacker is also able to deanonymize users concerning their utilization of certain services.

## 4 I2P vs Tor

As mentioned at the beginning of our report, there are a lot of software that are deployed to help to provide anonymity of internet users. One of the first system is Tor. Here we compare the Tor network to I2P network.

### 4.1 Tor

Tor network is a free software and open source that helps to provide anonymity of users. Tor is composed with 3 different types of node: directory servers, exit points, and internal relays. The directory server contains a list of relays which are to be obtained by the client. The directory servers do not only send the list of relays to the client but also his address and some basic configuration files. In order for the client to get valid relays, he needs to choose a trusted server. Once the client has received a list of valid and operational relays, he could choose an optional route for his traffic across the Tor network and later transmit it to the exit node. Tor components are basically the PC to which you are connected, relays, routers, and exit node[9,10].

In Tor network all traffic follow the same route and exit at the same point. This approach is quite different to way I2P operates. During the traffic all message are fully encrypted as in I2P network. The routing protocol in Tor is known as "Onion routing" which are used to repeatedly encrypt the original data including destination IP address, and send it through several network nodes called onion routers. It is also important to notice that exit node is only aware of it intermediate node to send/ receive data. He does not know what is actually in the message because it is fully encrypted . The difference between the exit and entry node is that: The entry node knows " who you are" but not "what you are doing " while the exit node knows " what you are doing" but not "who you are". The relays in between only forward encrypted messages.

In addition to Tor capability, it is possible for the users to access what it called "hidden services" such as emails servers, forums etc.

### 4.2 I2P

Tor and I2P both have a lot of benefits in common since they are both designed to help provide better anonymity. But from the beginning Tor was designed to help users to access public network anonymously . while I2P first function was to be a network between the network "being a true darknet"[10].

In contrast to Tor, all nodes in I2P act as routers. As mentioned before I2P uses different route than Tor network: I2P traffic are unidirectional (sending and receiving route are different) which help to increase reliability and redundancy to the network while Tor uses a simple duplex circuit (sending and receiving messages go through the same path).

In I2P, informations about peers and en/decryption keys are stored in the NetDB while in Tor those informations are stored in a central directory servers. Concerning the application-level, The users to be able to use Tor, the must configure the proxy services which are stored on his computer. While in I2P, special applications have already been written and must only be used in I2P network (i2pSnark, eepsites, and i2p-messenger ).

Except that I2P and Tor distinguish themselves from each other in some other details, which will not be discussed in this paper, the main part of which was summarized briefly by in a table[11]. Most of them have already been mentioned earlier.

| | I2P | Tor |
|---|---|---|
| Communication Method | Packet-Switched | Circuit-Switched |
| Communication Protocol | TCP or UDP | TCP |
| Type of Tunnels | Unidirectional | Biddrectional |
| Routing Algorithm | Garlic | Onion |
| Lifetime of Tunnels | Short-Lived | Long-Lived |
| Encryption Type | Link-, Layered- and End to End encryption | Link -and Layer Encryption |
| Storage of Peer Info | Floodfill Peers | 7 Directory Servers |
| Service Provider | Build-In Servers | External TCP Servers |
| Number of Services | Many Integrated Services | Few Hidden Services |
| Peer Selection | Performance-Based | Bandwidth-Based |
| Hop Number in the Tunnel | User-Configurable, Random Number of Hops | 3-Hop |
| Number of Exit Nodes | One | A large Number |
| Network Dimension | SMall | Large |
| Implementation Code | Java | C |

Table 1. Main Differences between I2P and Tor Network

Some advantages of Tor Over I2P: Tor offers better resistance, performance[14], documentations, low bandwidth overhead, more translations. While I2P on the other side is faster, completely distributed, self organized, selection of peers through ranking performance[12].

## 5 Conclusion

In this paper we firstly presented a brief description of I2Ps components then we commented on some weaknesses such as DoS etc. We compared the routing aspects of I2P system to Tor and saw that both Tor and I2P provide cryptographically methods to anonymously access information and communicate online. Tor provides service with better anonymity for Internet and higher QoS for any network, while I2P provides a more robust and reliable network within the network. Of course, when implementing either of these two tools, one must always be aware of that ones ISP can see that he or she is using Tor or I2P (though they cannot determine the content of the traffic itself). In order to hide this knowledge from ones ISP, one should make use of a high-quality VPN service to act as an entry point to either ones anonymous network arbitrarily or to the Internet at large. One of the key elements that annoy anonymous system researchers is QoS for the bandwidth utilized by peers on the systems and the overall network performance. Although this has been slightly commented on, more research in QoS and a bandwidth-choking approach is required while concentrating on security and functionality implications.

## References

1. Abdelberi Chaabane, Pere Manils, and Mohamed Ali Kaafar. Digging into Anonymous Traffic: A Deep Analysis of the Tor Anonymous Network. In Proceedings of

the 2010 4th International Conference on Network and System Security, NSS '10, Washington, DC, USA, September 2010. IEEE Computer Society.
2. J. Jrandom, I2P Anonymous Network: Technical Introduction, Retrievedon December 13, 2010, from Anonymous Network.,http://www.i2p2.de/techintro.html
3. Privacy-Implications of Performance-Based Peer Selection by Onion-Routers:A Real-World Case Study using I2P. Michael Herrmann and Christian Grothoff. http://grothoff.org/christian/i2p.pdf
4. "Introducing I2P"., http://geti2p.net/en/docs/how/tech-intro
5. wiki.ubuntuusers.de/i2p
6. Goldschlag, D. M., Reed, M. G. and Syverson, P. F.: Hiding Routing Information. In: Information Hiding. Lecture Notes in Computer Science. Cambridge, UK: Springer-Verlag. May 30 - June 1, 1996: 137 - 150.
7. Maymounkov, P, Mazieres, D.: Kademlia: A peer-to-peer information system based on the xor metric. p. 53-65 (2002)
8. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium (August 2004)
9. Ehlert M.: I2P Usability vs. Tor Usability: A Bandwidth and Latency Comparison. Seminar. Humboldt University of Berlin. Berlin, Germany, November 2011.
10. Herrmann, M. and Grothoff, C.: Privacy Implications of Performance-Based Peer Selection by Onion Routers: A Real-World Case Study Using I2P. In the Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 11). Waterloo, ON, Canada, July 27 - 29, 2011.
11. http://i2hq.srv.i2p2.de/de/docs/protocol
12. I2P Compared to Tor., https://geti2p.net/de/comparison/tor
13. TMA2012-LNCS.pdf. http://hal.archives-ouvertes.fr/docs/00/63/22/59/PDF/TMA2012-LNCS.pdf
14. Panchenko, A.; Lanze, F.; Engel, T. "Improving performance and anonymity in the Tor network", Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International, On page(s): 1 - 10
15. B. Zantout and R. Haraty, I2P data communication system, in ICN 2011, The Tenth International Conference on Networks, 2011, pp. 401409.
16. I2Ps Threat Model.https://geti2p.net/en/docs/how/threat-model.
17. Zantout, B.C. and Haraty, R.A.: I2P Communication System. In the Proceedings of 10th International Conference on Networks (ICN 11). Saint Maarten, The Netherlands Antilles, January 23 - 28, 2011: 401 - 409.
18. Egger, C., Schlumberger, J., Kr gel, C. and Vigna, G.: Practical Attacks against the I2P Network. In the Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 13). Rodney Bay, Saint Lucia, October 23 - 25, 2013