A New Look at the TOR Anonymous Communication System

Imran Ahmad
Riphah International University Lahore
Pakistan
imran221975@yahoo.com

M. Saddique, U. Pirzada, M. Zohaib, A. Ali, M. Khan
Cecos University of IT & Emerging Sciences
Peshawar, Pakistan
icrg.csit@gmail.com

Journal of Digital Information Management

ABSTRACT: *Whenever a user tries communicating with another recipient on the Internet, vibrant information is sent over different networks until the information is intercepted or normally reaches the recipient. Precarious information crisscrossing networks is usually encrypted. In order to conceal the sender's identity, different implementations have proven successful - one of which is the invention of anonymous communication systems. There are many anonymous communication systems developed but, the Onion Router (Tor) is the greatest organized anonymous communication system, which offers online anonymity and privacy. There are a vast number of obstacles in security that have to be considered when deploying Tor. This paper thoroughly investigates and presents these security issues in Tor.*

Subject Categories and Descriptors
C.2: [Computer-Communication Networks]: Network Architecture and Design, Network Operations

General Terms: Security, Communication Systems

Keywords: Tor, Onion Routing, Design and Non-design Objectives, and Security Issues

Received: 5 April 2018, Revised 12 June 2018, Accepted 23 June 2018

DOI: 10.6025/jdim/2018/16/5/223-229

1. Introduction

Tor is a network of implicit channels that enables a user to connect to a manager with heightened confidentiality via the Internet [1]. Remote hosts can be introduced by using Tor from learning a user's location (IP address). The basic working of Tor is that it routes the outgoing connections from a client's computer via "onion routers". "To create a confidential structure - passageway - with Tor, the software of the user/client increasingly makes a circuit of networks, which are encrypted on the net through the servers. The circuit, which is created, is then lengthened through one jump at a time and each server only knows from where the data is coming from and to whom will it be transferred to. None of the servers ever knows the complete path. For each jump, the client uses a set of encryption keys, which are separate, so that each jump should not be traced as these links are passing through [2]. When a circuit is constructed, different forms of data can be traded and different types of software (applications) can be utilized over the Tor network [3].The use of traffic inquiry - to link the networks destination and source - cannot be done because in the circuit each server cannot see more than one hop, (neither by an adversary nor a malicious server).

To help protect everyday confidentiality by letting on the user to be anonymous, Tor acts as an excellent system for those who want to make outbound links that prohibit the use of certain protocols. Tor is one of the best services

which provide anonymity online [4]. It routes data, packed into equally sized frames, along a (cryptographically) secured path called onion routers. The routing follows the principles of CSN (circuit-switched networks), from where the terminology is provided to Tor. Each router only knows the predecessor and successor. This is achieved by limiting the perspective of onion routers on a circuit, which in return gives a high level of anonymity. In every jump a "coating" of cryptography is removed or added which depends on the direction of flow. A client, who wants to connect to a remote server anonymously, uses Tor as a proxy. All the connections and messages go through Tor first, then to the server. Thus, the client is hidden by the server because the server believes that the connection is coming from Tor. The Tor system is made up of a network of relays. Each relay is a volunteer machine. The client picks three relays from the network to form a circuit: the entry node, the middle node, and the exit node. The client establishes a connection with the entry node, then using the entry node as a proxy, extends that connection to the middle node, and finally, extends the same connection to the exit node. Currently, there are more than 500, 000 users in Tor and more than 6,000 relay nodes [5].

At first sight, the anonymity of navigating through the Internet may be used mainly by people with malicious intentions. However, the analysis shows that the real situation is much more complex.

The relative technical easiness of recording the navigation routes of numerous Internet users creates the preconditions of getting to know and accumulating their interests of various kinds. This information may be used by malicious people (probably, acting with the help of intelligent robots) for inventing the ways of making attractive for certain categories of Internet users some actions (purchases, donations, etc.) resulting in a considerable damage for these users.

That is why the use of anonymous communication systems like TOR protects very many Internet users from the attacks of people with malicious intentions. As a consequence, the usage of TOR by normal people contributes to their harmonic existence in knowledge society, and this corresponds quite well to basic objectives of cognitonics [6].

The premise of anonymity provided by Tor relies on the three relays used by the client to be non-colluding. Moreover, the identity of the three relays used by a client to connect to a server is hidden. If an adversary could somehow identify the three relays used by a client, this breaks some of the anonymity of the client as it reveals which three Tor relays the client chose, as after the Tor relays in the circuit have been identified, and the identity of the client is also leaked. Thus, de-anonymizing the three relays used by a client is the first step towards identifying which client is communicating with which server. This has a colossal tremble on Tor as the anonymity of any Tor user can be compromised [7].

This paper takes a deeper look at Tor, and highlights its security and other open issues. The remainder of the paper is organized as follows. Section 2 provides a background of Tor. Section 3 describes the components of Tor. In section 4, circuit creation and node selection is presented. Section 5 deals with transmission of data. The design and non-design goals are presented in section 6. Sections 7 and 8 provide the research and open issues in TOR. And section 9 concludes the paper.

2. Background

The Onion Router (Tor), as depicted in figure 1, is a circuit-based, low latency, overlay network which provides anonymity and privacy. It is the most deployed/available anonymous communication system in present era. Its users are in hundreds of thousands, e.g. military, intelligence agencies, journalists etc. and in more than 75 countries with over 6000, relays to provide online anonymity and privacy. The idea inside Tor is of "onion routing". David Goldschlag, Paul Syverson , Michael Reed developed it in the mid-1990s. It is funded by the U.S. Naval Research Laboratory.

For anonymous communication anonymity over computer/ internet is provided by Onion Routing (OR). Messages are encrypted and then forwarded to nodes known as onion routers. A header is peeled and the instructions for routing to next router are performed. This process occurs in repetition. No initial node or intermediate nodes know where the message is being passed send or received [8]. There are three nodes/relays in Tor as depicted in Figure 1: entry node, middle node, and exit node. As a communication system, there are four basic components in Tor: sender, receiver, onion routers and directory servers.

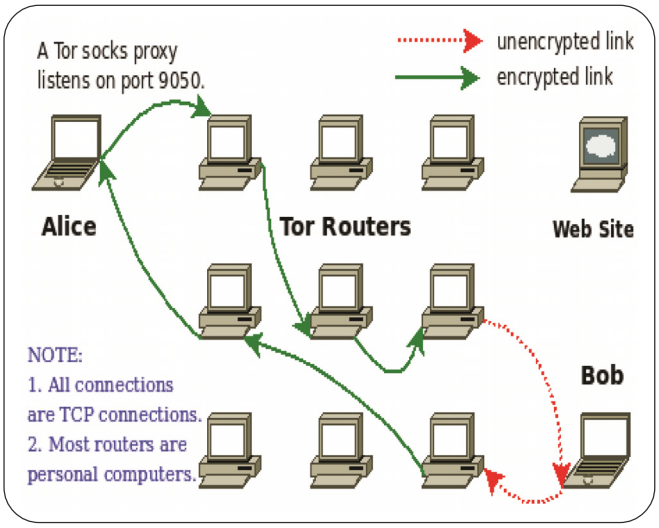

Figure 1. The Tor architecture [9]

3. Components of TOR

The Onion Router has the most number of users among all the anonymous communication systems and open-

source projects. It contributes anonymity for proxy awareness and TCP (Transmission Control Protocol) applications. As a communication system, there are four components of Tor as shown in figure 2.

Directory Servers: A Directory Server's authority holds related information about onion routers such as public keys for ORs and paths for onion relays, which are downloaded into directory caches. Tor clients download this information about onion routers, and these subsequently are used to construct circuits in the Tor network.

Client (Sender): The client uses Tor for anonymous communication. It fetches the information from directory servers and runs software known as Tor, Onion Proxy (OP) that fetches directories for consensus documents to build the required circuit for the network. For user applications, it handles connections, encrypts the data and sends through the Tor anonymous communication system.

Server (Receiver): It is the receiving party of the anonymous communication.

Onion Routers: They are Tor proxies, which forward/relay the data or messages from/to senders and receivers. Each router, without any specific advantage has to run a normal client/user-level process. There is a TLS connection between onion routers. The TCP stream accepts the OR and subsequently gets forwarded/multiplexed to the specified circuits. The last OR of the circuit, i.e., the exit destination, requests to relay data [10].

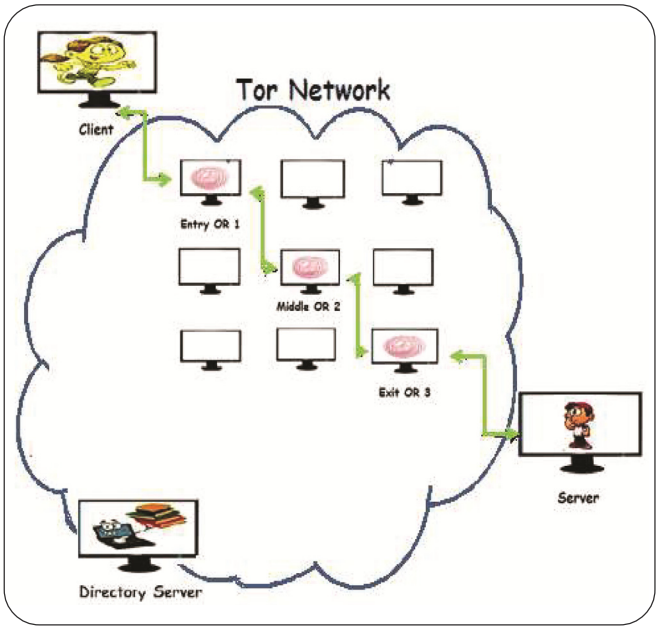

Figure 2. The components of Tor

4. Circuit Creation and Node Selection

There are three nodes/relays in Tor: The entry node, middle node, and exit node. Initially, nodes selections were random. This approach was good for anonymity but bandwidth/capacity of the node was not considered. To improve the performance of the Tor, selection criteria was changed and a node with high bandwidth is to be initially selected.

High bandwidth relays have a greater probability to pick than low bandwidth relays. Before building a new circuit, a path is generated. The exit node selects it first and it follows the desired nodes. The paths generation/selection is constructed under the following constraints:

- The same router is not selected twice for the same path.
- More than one router from the same family is not selected for the same path.
- More than one router is not selected in a given subnet for the same path.
- Non-running or non-valid routers are not selected unless configured.
- The first node must be a guard node [11].

Tor nodes are classified into four types based on flags assigned by directory servers: "Guard", "Middle", "Exit" and "Double" nodes (denoted as G, M, E and D respectively). Nodes with neither "Guard" nor "Exit" flag are considered Mnodes, while nodes with both "Guard" and "Exit" flags are considered Dnodes.

In the circuit construction phase, nodes selection is done according to their bandwidth. It is notable that in Tor, "Guard" and "Exit" flag nodes depend on the selection position and the total bandwidth they make. Let p_{en}, p_m and p_{ex} be the entry position, middle position and exit position in a circuit. The bandwidth weight for Tor nodes of type t serving at p position can be denoted by $W_{p,t}$ where $p = p_{en}, p_m$ or p_{ex} and $t = G, E, M$ or D. In practice, Tor directory servers calculate the values of $W_{p,t}$ based on a few rules and published them in the hourly-announced Tor network consensus file. Let $B(_{ni})$ and $T(_{ni})$ be the bandwidth and node type of Tor node n_i respectively. The approximate probability that node n_i is chosen to serve in a circuit at position p can be determined by:

$$P(_{p,n}) = W_{p,T(ni)} B(n_i) / \Sigma_j W_{p,T(nj)} B(n_j) \text{ [12].}$$

By default, each circuit is used for at least 10 minutes before it is recycled and a new one is created. The OP fetches the Tor node/relay information from Directory Server such as exit policies, IP address, bandwidth uptime and public key. The Tor nodes are known as relay nodes: OP chooses an entry node, a middle node and an exit node. The entry node is very important for protecting the client's anonymity. The middle node knows the identities of entry node and exit node, but is not aware of the client or server in the circuit. The exit node is crucial for making application-layer connections and serves as a gateway between the Internet (non-encrypted) and the Tor network (encrypted). The Onion Proxy that routes the session key

(shared) with each Tor node and relay node, sets up a circuit ID known as the cell header for the circuit. As an OP establishes/builds a circuit, the application message starts its transmission. The message received by the application layer (via SOCKS) to OP opens the first circuit.

The received message is separated into cells of fixed size (i.e., 512 bytes) with a payload and a header in each cell. The session keys (shared) decrypt/removes a cell of data routing through the circuit and a specified key removes one layer of encryption. The data or information emerged as plaintext on the exit node and that plaintext message is forwarded to the server/receiver. The cells are neither explicitly reordered nor delayed or dropped. Based on their commands, shown in tables 1 and 2, cells either are control cells (interpreted by the node that receives them) or relay cells (carrying end-to-end stream data) [1].

Control-cell command	Function
Padding	Use for link padding
Create	Setting up a new circuit
Created	An ACK of a circuit
Destroy	To destroy a circuit

Table 1. The control cell commands

Relay Command	Function
Relay data	To begin the flow of data
Relay begin	To open a stream
Relay end	To close a stream
Relay tear down	To stop a broken stream
Relay connected	An ACK for OP
Relay extend	To extend the circuit by a hop
Relay extended	An ACK that hop is extended
Relay truncated	An ACK that circuit is torn down
Relay send me	To control traffic entry
Relay drop	A relay is dropped

Table 2. The relay commands

5. Transmission of Data

When a client desires a TCP connection on a given port/address, it requests the OP, and the OP establishes a connection through the secure socket (SOCKS) server. The OP selects a new or a fresh circuit and picks an exit onion router for it. Ending a Tor network stream is similar to ending a TCP stream: one-step handshake is for errors and two-step for normal processes.

The Onion Proxy completes the process of establishing the circuit. Now it can communicate a message with the help of session keys with each OR on the circuit and the process occur on relay cell commands. To construct a relay cell addressed to a given OR, a client assigns the digest, and then iteratively encrypts the cell payload, which is the relay header and payload. At each step for the meaningful OR a digest encrypts. The exit point is the criteria selection for client with the required exit policy. Circuit building is incremental, so their tear or breaking down will be incremental.

When a cell arrives at an OR, the cell is decrypted. The necessary information is extracted from the header and the cell is relayed or forwarded to the output circuit queue (active circuit). Each circuit connection has an output buffer (data writes on that buffer) that transmits data to the next onion relay in the fashion of first in first out order (FIFO). Generally, a single connection is shared on multiple circuits. The circuit queues containing cells are multiplexed into the output buffer. If the cells are completely moved from the circuit queue to the output buffer, that circuit is checked as inactive.

At each hop, integrity check occurs. When a client communicates a key with a new hop, the concept of SHA-1 digest is used for verification of data and correctness of hashes. Computing the digest on each hop is much faster than doing the AES (advanced encryption standard) encryption for a cell, containing data that travels through the Tor circuit. The session key decrypts the cell from the Tor relay/node circuit (removes an encryption layer). That data or information remains as plaintext on the exit relay, which is forwarded as plain-text message to the destination/server. When an OR later replies to a client with a relay cell, it encrypts the cell's relay header and payload with the single key it shares with the client, and sends the cell back towards the circuit. Further layers of encryption are added by successive ORs back to the client by relaying the cell.

6. Design Goals/Non-goals of TOR

The following are the design goals/non-goals of Tor:

***Goals* (Table 1):** Anonymous systems are designed for low–latency. Tor defends against attackers from connecting as communication companions and its users from connecting to multiple communications.

7. Research Areas in TOR

The current areas of research in TOR include:

Security: A traffic confirmation attack [13] is possible when an attacker is controlling the relays on both ends of a Tor circuit and comparing traffic timing, volume, and

Design	Developments
Open/simple	Tor is open source; simple protocol; with security parameters paving the way for its defense
Accessibility on different platforms	Tor is easily accessible on different platforms (e.g., Windows, Linux, Mac, etc.)
Design is deployed	Tor is deployed in the real world and volunteers are willingly making it possible
Flexibility	Tor's flexible and well- identified protocol makes it a hotspot for future research

Table 1. Design goals of Tor

Non-goals (Table 2): Tor's deployable and simple design left unsolved questions, which need to be addressed.

Work required in its design	
Protocol	As like Privoxy or Anonymizer, Tor does not provide protocol normalization
End-to-end attacks	Traffic confirmation or end-to-end timing attack requires attention in the Tor community
Steganography	Steganography is not concealed in a Tor network
No peer-to-peer	A Tor network is non-peer-to-peer

Table 2. Non-design goals of Tor

other characteristics. This makes it possible to locate that the two relays are in on the same circuit. If the first (entry guard) and last relays (exit node) know the direction of the destination and the source in the circuit, then together they can de-anonymizeit, which demolishes the security of the data as well as the IP of the server and the destination. More work need to be done to avoid these types of attacks so that security is guaranteed.

Confidentiality: *Overall,* Tor networks are susceptible to numerous attacks. A path selection attack is an example of one such broad category of attacks. In Tor, the initiators choose the nodes on the circuit so the last nodes cannot be combined. The length of the circuit is three by default. Because of this, latency is kept to a minimum. This opens the door for connected attacks, which include congestion of the genuine Onion Routers to a point where they cannot require a fresh circuit to be built. The Tor last nodes pose a risk to confidentiality, since anybody can offer to route a Tor node. An assailant would have full admittance of data which is being routed if the assailant occurs to route the last node in a circuit. While in MITM (Man in the middle attack) [14] the exit nodes can similarly transmit attack by mentioning back a false text for the site the originator wants to join.

Authentication: Each onion router keeps Transport Layer Security (TLS) connection with all other onion routers. Tor uses TLS cipher suites with ephemeral keys. All TLS connections use short-term ephemeral keys. Short-term ephemeral keys are Onion encryption keys. Every onion router issues a router self-key. Moreover, directory servers keep a long term, authority self-key (stored offline) and a medium term authority signing key (3–12 months). The Onion Proxy does not have any identity keys. Tor uses a number of nodes located around the Internet to protect users' privacy. It is important that originator can guarantee that his/her communications with the many nodes is authenticated: If a malicious man-in-the-middle attack functioned or cooperated with one node to connect to the first node, then authentication is lost.

Performance: Many works have been done to improve the performance of Tor [15-16]. This led to improving the performance of Tor and moving it from a high latency to a low latency network. However, due to the constant processing of cryptographic modules, Tor is slow in performing these actions. For the improvement of Tor, more work needs to be done.

Anonymity: Tor is free of cost software for enabling online anonymity. This feature makes it viable for users to search and surf the Internet, making them untraceable (activity and location) by government agencies, corporations, or anyone else. However, more work is needed for improving anonymity (online) and defending against attacks.

Censorship Resistance: Censorship circumvention systems such as Tor are highly vulnerable to network-level filtering. Because the traffic generated by these

systems is disjoint from normal network traffic, it is easy to recognize and block; and once the censors identify network servers (e.g., Tor bridges) assisting in circumvention, they can locate all of their users. Due to this, Skype-morph [17] was introduced, but there is more work which is needed to avoid the blocking of Tor relays/ bridges.

Scalability: Tor's insistence on deploy ability and simplicity of design has led to the adoption of a clique topology and semi-centralized directory that made the network model completely visible to client knowledge. These properties cannot scale past a few hundred servers, but implementation experience will be useful to learn the relative importance of these bottlenecks.

Path Selection/Circuit Creation: A Tor client initially contacts Directory Authorities to fetch the consensus. As a Tor client gathers information about existing relays, it tries to build circuit paths. The paths are created according to the following rules [11]:

• The guard node should be the first node.

• For the same path, routers should not be identical in a Tor family.

• Un-valid or non-running router/relays are not selected. If their configuration is proper, then it is allowed to be connected in network.

8. Open Questions in TOR

The following is a list of open issues in Tor:

• The previously discussed issues are based on active and passive measurements (circuit latencies) as well as throughput estimations for improving the performance of anonymous communication channels provided by Tor. Work needs to be done on viable significance of new methods on security and anonymity of the system.

• The Circuit Clogging Attack [18] can be used to identify all the Tor relays used in a circuit; however, it is an open question to identify which of the relays are the entry, middle, and exit relays.

• There needs to be work done on the basis of onion proxies; i.e., to prevent compromised onion proxies to send false information so that they can obtain high scores. Also needed is the maintenance of Tor performance when the mechanism for optimizing Tor node store and output mode reduces the choice of relay nodes.

• Current algorithms may be modified to optimize performance by improving classification of the bulk traffic and considering alternative strategies for distinguishing web from bulk connections. Additional approaches to rate-tuning are also of interest. For example, it may be possible to further improve web client performance using proportional fairness to schedule traffic on circuits.

• Reliable relay of information is very important for building paths with better performance; therefore, a Relay Recommendation System (RRS) is needed for Tor to offer reliable relay material with better performance for building paths, ease low-resource attacks, and enable operators to explore the compromises among anonymity and performance based on their needs.

• The importance of the Tor network, as an online tool, is to safeguard the confidentiality and to try to improve the performance of applications for interactive users. To do this, researchers proposed Personal Computer Transmission control Protocol or PC/TCP (IPsec over TCP for the circuit), a new transport mechanism for Tor anonymous communication that allows you to design circuits protected by IP sec [19] TCP connection. There are some areas for improvement in this very aspect.

• The use of the path length is the key factor of path selection to provide flexible and easily deployed tunable options for users. It is an open research to design more options utilizing more potential factors in Tor to provide fine-grained tunable functions.

• Simple strategies are used to improve the selection method for relays with high bandwidth and TCP advertised window sizes. Bandwidth is a key factor in Tor design and path selection. An open research question is to work on this inadequate balance in the load distribution to enhance Tor circuit and the efficiency of performance.

In (LASTor): A Low-Latency AS-Aware Tor Client, a technique is used by agreeing on a value of 0 for low-latency and 1 for high-anonymity for parameter selection. An operator can select a suitable trade-off among anonymity and latency. An open research question that needs to be further investigated.

• The nodes which are under the same person/organization are called family nodes. There are many open research questions regarding family nodes: examining Tor family's influence, or Tor performance, availability and anonymity especially when family nodes are under attack.

Moreover, one needs to look deeper into Tor's family mechanism and discovering potential family misconfigurations in the Tor network.

9. Conclusion

Tor is free software which empowers censorship resistance and provides online anonymity. In this paper, many research areas in Tor were analyzed and described: Performance, security, anonymity, censorship resistance, scalability, and circuit creation/path selection. In recent years, Tor has become a research hotspot in the anonymous communication systems research community. Future work involves conducting a detailed study to compare Tor with other anonymous communication

systems.

References

[1] Dingledine, R., Mathewson, N., Syverson, P. (2004). Tor: the second-generation onion router. *In*: *Proceedings of the 13th Conference on USENIX Security Symposium-Volume 13*.

[2] Reed, M., Syverson, P., Goldschlag, D. (1998). Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*– 16(4), *p*. 482–494.

[3] Syverson, P., Reed, M., Goldschlag, D. (2000). Onion routing access configurations. *In*: *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX 2000)- Volume 1*, IEEE/CS Press, *p*. 3–40.

[4] Acquisti, P., Dingledine, R., Syverson, P. (2003). On the economics of anonymity, in Springer-Verlag, LNCS 2742, *p*. 1-19.

[5] The TOR Project. Retrieved from http://tor.eff.org/on July 21, 2017.

[6] Fomichov, V. A., Fomichova, O. S. (2012). A Contribution of Cognitonics to Secure Living in Information Society. Informatica. *An International Journal of Computing and Informatics* (Slovenia). 36(2). *p*. 121-130.

[7] Haraty, R., Zantout, B. (2014). The TOR data communication system. *Journal of Communications and Networks*. ISSN 1229-2370. 16(4), *p*. 415-420.

[8] Haraty, R., Zantout, B. (2014). The TOR data communication system – A Survey. *In*: *Proceedings of the Sixth IEEE International Workshop on Performance Evaluation of Communications in Distributed Systems and Web based Service Architectures (PEDISWESA'2014)*. Madeira, Portugal.

[9] Tor - The Onion HTTP Router. Retrieved from http://tohr.sourceforge.net/ on July 21, 2017.

[10] Ling, Zhen., JunzhouLuo, Yu, Wei., Fu, Xinwen., Xuan, Dong., WeijiaJia. (2012). A New Cell-Counting-Based Attack Against Tor, *In: IEEE/ACM TRANSACTIONS ON NETWORKING*, 20(4), *p*. 1245-1261.

[11] The Tor System. Retrieved from http://tor.stackexchange.com/questions/113/how-does-a-tor-client-pick-tor-nodes-for-circuit-creation on July 21, 2018.

[12] Wan, Xiao., Shi, Jinqiao., BinxingFanand, Li Guo, (2013). An Empirical Analysis of Family in the Tor, in *Communication and Information Systems Security Symposium, IEEE ICC*, *p*. 1995-2000.

[13] Murdoch, S., Danezis, G. (2006). Low-cost traffic analysis of Tor. *In*: *Proceedings of the IEEE S&P*, p. 183–195.

[14] Jansen, R., Syverson, P., Hopper, N. (2012). Throttling Tor bandwidth parasites. *In*: *Security'12 Proceedings of the 21st USENIX Conference on Security Symposium*, CA, USA, *p* 10-18.

[15] Haraty, R., Zantout, B. (2015). A collaborative-based approach to avoiding traffic analysis and assuring data integrity in anonymous systems.*Computers in Human Behavior Journal.* Volume 51, Part B, *p*. 780–791.

[16] Haraty, R., Assi, M., Rahal, I. (2017). A systematic review of anonymous communication systems. *In*: *Proceedings of the 19th International Conference on Enterprise Information Systems*, Porto, Portugal.

[17] Moghaddam, H. M., Li, B., Derakhshani, M., Goldberg, I. (2012). Skype Morph: Protocol obfuscation for Tor bridges. *In*: *Proceedings of the ACM Conference on Computer and Communications Security, CCS'12*, pp. 97-108.

[18] Tin, C., Shin, J., Yu, J. (2013). Revisiting circuit clogging attacks on Tor. *In*: *Proceedings of the Availability, Reliability and Security (ARES) eighth International Conference*, *p*. 131 – 140.

[19] Back, A., Moller, U., Stiglic, A. (2001). Traffic analysis attacks and trade-offs in anonymity providing systems. In *IH 2001*, Springer-Verlag, LNCS 2137, *p*. 245– 257.