Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Review Paper

A Comparative Study on Anonymizing Networks: TOR, I2P, and Riffle Networks Comparison

M. Hosseini Shirvani*, A. Akbarifar

Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran.

Article Info

Article History:
Received 25 July 2021
Reviewed 29 August 2021
Revised 01 November 2021
Accepted 01 November 2021

Keywords:
Tor project
I2P network
Riffle
Dark-web
Anonymity
Security

*Corresponding Author's Email Address:
mirsaeid_hosseini@iausari.ac.ir

Abstract

Background and Objectives: Among miscellaneous networks, onion-based routing network technologies such as The Onion-based Routing (ToR), Invisible Internet Project (I2P), and Riffle networks are used to communicate anonymously by different worldwide users for security, privacy, and safety requirements. Sometimes, these types of networks sacrifice anonymity for the sake of efficient communication or vice-versa. This paper surveys aforementioned networks for investigating their potential and challenges.
Methods: Onion-routing networks encapsulate messages in several layers of encryption similar to layers of an onion. The anonymous communication networks are involved dining cryptographers (DC) problem so-called DC-nets, which need sending anonymous message with unconditional sender and untraceable receipt. So, DC-nets must be resistant against traffic analysis attacks although they will attenuate the network bandwidth. In this line, ToR is a free software that provides anonymous communication, I2P networks are based on hidden internet service project which uses tunnelling for anonymous communications, and Riffle networks include a small set of camouflaging servers that provide anonymity for authorized users. This paper presents a comparative study on anonymizing ToR, I2P, and Riffle networks in terms of associated prominent parameters in this vein.
Results: The comparison is based on similarities, differences, and challenges in network behaviors. This comparison is beneficial for further researches and future improvements.
Conclusion: The review of the current paper reveals that the Riffle networks are more resilient and have great confidentiality and integrity against other onion-based routing networks.

Introduction

Mission critical applications associated to individuals and organizations need meeting security requirements for their business process fulfilment [1]-[4]. In their roles, stakeholders are also interested in privacy and protecting their identity on the web in some cases. In this regards, onion-based routing network technologies such as The Onion-based Routing (ToR), Invisible Internet Project (I2P), and Riffle networks are used to communicate anonymously by different worldwide users for security, privacy, and safety requirements on the web [5].

Sometimes, these types of networks sacrifice anonymity for the sake of efficient communication or vice-versa. Onion-routing networks encapsulate messages in several layers of encryption similar to layers of an onion [6].

Albeit, packet encryption in the network is aimed at confidentiality, integrity, and authentication goals but

tracking the packet will be possible because the routers need to know the source and the destination of data [5].

Anonymizing communication networks such as ToR, I2P, and Riffle are attempting to hide the identity of sender's information [5]. ToR networks provide an anonymous communication with hiding the real locations of the senders. Notwithstanding communication profits, hidden web (Dark-web) on ToR's network caused many problems for the government due to its concealment. The Dark-web that is non-registered domains has a specific 16-character address; so, it becomes a place for cyber criminals for some actions like arms trafficking, selling human organs, drug dealing, assassination missions, cyber-attacks and etc. [7]. On the other side, I2P network is a peer to peer and message-oriented anonymizing communication network. Basically, I2P has been developed in order to anonymizing a connection between two regions within the network. The I2P network was firstly introduced in 2003 and its origins could be founded in invisible internet project [7]. This network is built on top of the internet protocol [8].

The Riffle identity anonymizing network is providing safety and security requirements when the attackers in another anonymous system make counterfeit servers to traffic analysis attacks, but the ToR networks are vulnerable to these kinds of attacks. On the other side, a Riffle network utilizes hybrid networks approach as a defensive mechanism against these attack scenarios. Similar to other anonymizing networks, a Riffle network uses of onion routing (OR) protocol that is a method for anonymous information's exchange in computer networks. The packets are encrypted successively and then they will be sent to too many network nodes or ORs. Each OR decrypts a cipher layer to read the routing instruction and then will send it to the next router to do the same process. This method caused the network nodes to have nothing known about the nodes contents and the origin of the packets [9]. A Riffle network is a shuffle network which shuffles data streaming with different keys. So, in the mixed networks, a batch of incoming packets is transmitted between different safe servers, without the use of inefficient public keys. In fact, the traffic will change before and after the logging into the server. Instead of using inefficient public keys, a Riffle network verifies the encryption based on verifiable shuffle during the validation of incoming encrypted packets. However, even unsafe servers on the network could not access the data. To do so, the unsecured servers should shuffle the message correctly, since the input data could be accepted by secure servers which are nearly impossible [10].

Recently, numerous authors have compared low-latency anonymizing communication networks in literature such as ToR, invisible internet project (I2P), and Riffle networks [5], [7], [9]-[11]. In this regard, an overview on utilization of anonymity technologies has been presented [12]. This study discusses user's security perils and describes principal mechanisms to prevent the attacks in anonymizing communication networks. According to this overview, the protection of user's privacy and its violations by government agencies and information security organizations has been investigated. Also, Invisible internet project (I2P) is one of the ToR's capabilities in which many researchers have studied about it in [7], [12], [13]. The contribution of the current paper is to present a comparative study on three anonymity communication networks ToR, I2P, and Riffle networks in terms of advantages and disadvantages along with their commonalities, discrepancies, and possible challenges in their network behavior. This comparison is beneficial for further researches and future scheme improvements.

Related Works

Although the anonymity communication networks are not very novel and stems back to 1997 [14], there is a clear lack in literature to pay on these types of networks. Nevertheless, we bring some of literatures to introduce their behavior.

An efficient communication system with strong anonymity called Riffle has been presented in literature by Albert Know et al. [10]. The proposed Riffle network provides a bandwidth and computation efficient communication network with high anonymity guarantees. To do so, it used hybrid verifiable shuffle and private information retrieval techniques [10]. A novel algorithm called ToRank was proposed that ranks hidden services in ToR networks when the users surf on web; this is because to lessen the harm to ToR network related to suspicious activities. It had successful behavior on famous datasets [15]. A universal serial bus (USB) side-channel attack on ToR has been introduced where a malicious is allowed to reach a public USB charging station [16]. This type of attack depends on power measurements of attacker's device without observing network traffic analysis [16]. In this vein, fingerprint attack is a famous threat [17]. To obviate this problem, an adaptive online website fingerprint attack for ToR networks has been introduced by Attarian et al. [18] To recognize the attack, the authors applied machine learning techniques in dynamic fashion. Also, the round robin queuing process was done to defense against protocol level attack in onion-based routing networks [19]. This utilizes integrity checks and counterfeit traffic auditing in the middle layers of relay nodes to recognize protocol level attacks. The review of related works introduces how the anonymity communication networks

perform along with known and contingent attacks and the countermeasures in these fields.

Motivation and Research Plan

As mentioned earlier, users utilize different anonymizing networks to meet their security, privacy, and confidentiality requirements. Although it is clear-cut that these types of networks make fortunities and also challenges, there is a clear lack of a survey study on these kinds of networks. This is the reason of preparing the current review paper in comparative perspective. To do so, our subjective research plan which contains explanation on several sub sections is depicted by Fig. 1.

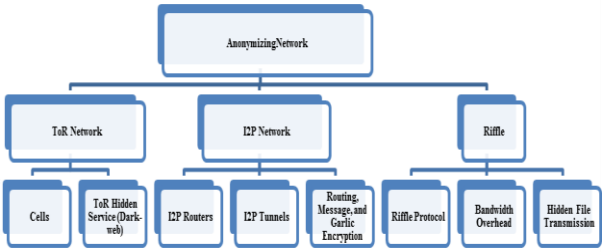

Fig. 1: Subjective research plan.

After introducing the main anonymizing networks along with their attributes, merits, and challenges, the comparative schemes in terms of prominent parameters are presented and discussed in forthcoming sections.

ToR Network

ToR is free network-based software which provides anonymous access to internet servers. There are two important factors in anonymity: firstly, the attacker should not be able to distinguish which IP address interacts with the servers and secondly, the server should not be able to recognize the IP address of data transmitter. Fig. 2 illustrates a typical ToR network structure. Note that the dark grey squares in Fig. 2 represent the core nodes; and the light grey squares represent periphery nodes.

The basic idea for ToR is to transmit traffic by cluster nodes in which they have no information about the source and destination of the transferred packets. This relies on the distributed system principle in which the whole system seems singularly and coherently from the user's point of view [21]-[23].

ToR network utilizes onion routing protocol in order to anonymize user's communication. In onion network, the packets would be encrypted into layers, so to this layered architecture, it is called onion configuration. An onion routing is applied to each node that it is responsible for encrypting each onion layers in order to discover data for the next node within given networks [5], [7], [9], [13], [21], [24]. In a ToR network, the user is required to create an orbital path to communicate with the server. The orbital path is created by using socks via onion proxy on the user's side.

Fig. 2: Core-periphery categorical model of ToR network [20].

A ToR has a directory of verified onion routers associated with their public keys.

The index is examined manually and prevented to make bogus ORs for controlling traffic; hence, it informs onion proxy (OP) from ORs and then onion proxies will fetch the sequence of OR from the intentioned routing list [7].

The first OR is called as a watchdog and the last node which is decrypting the last layer is called exit node. Note that in existing ToR network, there are 3 ORs such as gateway or input, middle, and exit nodes that communicate with OP via transport layer security (TLS) secure connection and disposable keys. This procedure is a development for anonymizing in ToR network.

In order to restore a list of all onion routers, the distributed valid server is used. These servers should be known or being published in particular websites and also be able to track topology changes in the network. Index servers combine network topology and establish subscriber identity form the whole network. These directories automatically will fetch by OPs. Also, user's software contains a default list of directory servers [5], [25]. Directory servers store all ORs information on a list.

The circuits are a virtual duplex communication in ToR network that is established between OPs and a category of ORs. Obviously, a single circuit path in OR's could use several TCP flows simultaneously. In order to prevent flows recognition by the attackers, the lifetime of these circuit paths would be 10 minutes. After spending the circuit path time, the circuit is eliminated and a new one would be used. Note that the new circuit would be made as operational background, so there will be no additional delay in the system [26].

Cells

Circuit path consists of numerous ORs. The client acquired traffic is placed in cells with constant size (512 Byte) to making traffic analysis more difficult. The cells within each OR are rearranged by a symmetric key. The cells within ToR network are considered as a unit which is composed of a header and the only response by a payload. These cells are able to control (build and destroy) or redistribution of the cells (end-to-end data streaming). For instance, if a user intends to create TCP flow at the first point, he should transmit the control cell to first relay station by determining the next OR address; then, the process begins with exchange the symmetric key to second relay station by Diffie-Hellman key exchange model [25]. This process is maintained as a similar model until the circuit configuration was created. Afterwards, the user transferred broadcasted cells, these cells will carry with an end-to-end data stream [5], [27].

ToR Hidden Service (Dark-web)

One of the most important features of ToR network is the Dark-web [7], [13]. In this section, the main focus is on finding pages that the others are not able to see that page, as well as user's identity. Hidden services are included some anonymous websites and servers that could be accessed only by onion routing based networks. The site addresses in ToR are represented by an onion domain that is registered nowhere. On the other hand, the addresses are like "name. Onion" in which the name is a 16-character string.

In computer networks, the Rendezvous protocol is used when two persons have no information for communicating with the second party. This protocol is caused the sources and also counterparts to find each other in peer to peer networks. The Rendezvous protocol, which uses handshake model for communicating, does not send the data before the preparation of the destination. The ToR network uses this protocol in order to build its hidden service.

The hidden service uses three network nodes and calls each of them a "Recommender Node". Then, the network sends a request to recommender nodes based on the "Recommender Node" placement. If recommender nodes receive a positive response, they will send their public key. Note that the recommender node is not aware of service location. In next step, the hidden service is provided "service descriptors". A service descriptor contains recommender nodes, their characteristics, and the public key. Service descriptor encrypts the data with the public key and puts it on the hash table. The distributed hash table is a key quantitative database that the value and the key are hidden service descriptor and 16-character address respectively. The 16-character address is generated from the public key of user's service. This address went to the distributed hash table and has no information about IP addresses. The user should have anonymous onion address to be connected. This process is done by using the hash table and users public key for value estimation. Hence, according to the Rendezvous protocol, the user chooses some points named Rendezvous and begins the key exchange. It is worth noting that the user has recommender key and server key in this step. After the intro messages are made by the user, the random point is encrypted by disposable hidden address via server public key and consequently, the only hidden server is able to read that. The 16-character onion address is composed secure hash algorithm 1 (SHA1) and the public key that encrypted with base32. Therefore, the production probability of acquired strings by another person via the same public key is very low and the address would be unique. In making onion address, 3 practical soft wares are used inducing Scallion, shallot, and Escholat [7], [13]. Shallot software is based on the hash structure by using GPU and Escholat which uses a list for a dictionary based search. Although hidden services have a beneficial application they have a wide range of misbehaviour in cyberspace. Creating and using these services as the Dark-web and generating anonymous pages are caused illegal services such as human organ trafficking, drugs, murder, kidnapping, cyber-attacks operations, hiring attackers, and other detractive functions which the government is looking to monitor this network [7], [13].

I2P Network

I2P is a message-oriented and P2P identity anonymizing network. This network is generated due to anonymous communication between two intermediate network sections. Today, various fields of I2P applications are available including unsigned web hosting, browsing web pages, file transfer, and email service.

Utilizing exterior service shows that the service which is not hosted in the I2P network requires external proxies. I2P is a cover network which allows the user to interact with the network anonymously. Technically, I2P has a framework based on java platform which is designed to provide peer to peer anonymous networks.

Each user is responsible for running I2P routers that establish I2P core software. All the packets in this network are transmitted by tunnels via I2P routers, as well as other counterparts. These tunnels would only be performed on route traffic. Therefore, internal and external tunnels are needed for input and output traffic [5]. Fig. 3 depicts a typical I2P network.

In I2P, peer selection is done by an executive row-based algorithm of each I2P router. After creating

internal and external tunnels, users will store their data connections in a global database which is called netDB.

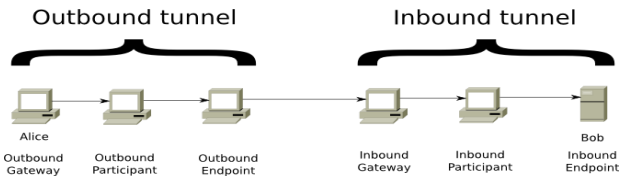

Fig. 3: I2P network with two types of tunnels inbound and outbound [28].

This database contains data connection of each I2P network and other public services within I2P network. The transmitted message is encrypted by Garlic encryption algorithm based on an end to end connection [29], [30].

Garlic encrypting is just same as onion encrypting with the exception that a single Garlic message may be contained within the different messages for various receivers [5], [3].

I2P Routers

I2P network is composed of users, nodes, or executive routers on I2P software which allows application communication to the mentioned network. I2P router is the core of this software. This router is responsible for keeping pairs and counterpart's statistics, which has been used in pair selection including encrypting operation, making the tunnel, providing service and sending the message. The aforementioned applications heavily depend on the tunnels that are created by I2P routers for being anonymous.

The netDB routing information and Lease set: super peers are utilized in creating and managing the database. The netDB is based on a distributed hash table which comprises all discovered information such as I2P pairs and I2P network-based services. Each super peer is only responsible for certain part of network information. Kademlia is a standard distance metric for XOR that determines which super peer is responsible for which part of the network depending on ID Couples [31], [32]. The peers with adequate bandwidth may be promoted to the super peer by decreasing the threshold. The netDB keeps the router infrastructure information as well as a Lease set for all known services.

I2P peers are recognized by a data structure called router information which includes all knowledge about the peers (IP address, port number, I2P stable version number, some of the statistical information, public key, and definitive 256-bit key of hash). In order to restore an available list of I2P peers, lists of routers information are able to be loaded from an identifiable system or a known web server. Restoring primary list from the router information is defined as reseeding operation.

A Lease set is a peer as the input gate to the internal tunnel of the corresponding service that is able to detect the habits. Both router information and Lease set are able to easily accumulated and restored by communicating with the closest super peer. For storing the super peer, the received router information and Lease set are transmitted to seven close super peers. About reseeding, two adjacent super peers communicate with each other. If the requested information did not exist, the super peer will provide a list of the closest super peers. On the other hand, the peers maintain the super peer information until the information restoring.

All routes in the I2P network are identifiable as a 256-bit encrypting key which is composed of a public key with 256-byte, an ID key with a 128 byte and a blank ID. Then, I2P refers to a primary service created by I2P router. Similar to domain name resolver (DNS), to mapping target names to cipher keys, three local host files are applied. In order to merge local and external host files, I2P is intended to make an index directory. Note that, this type of addressing will increase the anonymity.

I2P Tunnels

In I2P network, all the message are transmitted by tunnels. The tunnel is a virtual half-duplex encrypting connection in which two or three I2Ps use. In contrast with ToR, I2P router is going to create a tunnel which itself is contained another tunnel. Primitively, each I2P router makes various tunnels for input or output traffic.

The first I2P peer of a tunnel is called gate tunnel. The last I2P peer is called the end point tunnel. For output tunnel, the I2P router which is responsible for creating a tunnel will always be the gate tunnel and for input tunnel, this will be always the end point tunnel. The default value and tunnel length will be customized by the user in the setting. The tunnel length is always an evaluation of anonymity and functionality. The long tunnels increase the anonymity, whilst decrease the functionality and performance. An application doesn't belong to a particular tunnel and it may need different tunnels for message broadcast. Generally, there are two types of tunnels: exploratory and user tunnels. Exploratory tunnels are the ones with the limited bandwidth that would not be used in sensitive privacy operations. A router uses these tunnels for communicating with super peers and restoring the netDB database. On the other hand, the exploratory tunnels are used for certain, management, and destroying other tunnels. Normally, rebuilding the tunnels prevent data analysis attacks.

Creating new tunnels is performed by the first set of I2P peers. As mentioned in the last section, the peer's selection is based on raw and profile selective algorithm. An exploratory tunnel is used when encrypted tunnel creation requests are transmitted to the first I2P router simultaneously. Each layer includes some information for a single I2P, such as symmetric key and machine address. Similar to OR circuit design, the message is conducted until it reaches to the last I2P peer. The return response is to the successor that each I2P peer will add a layer of encryption. The I2P peer receivers are free to accept or reject the requests.

Routing, Message, and Garlic Encryption

I2P router can send and receive the message by this network while at least one output or input tunnel is created. In order to connect with an I2P service, firstly the router should restore service destination from the super peer. The destination determines a set of input gate tunnels from the counterpart service.

I2P uses the Garlic routing that is a kind of onion routing. This routing uses several Garlic message called Cloves. The Cloves are the data message with traditional routing instruction such as latency. On the other hand, it could be concluded that the Garlic message contains several practical messages. The real data message is encrypted by an end to end connection via a public receiver key. The Garlic message is encrypted multiple times via symmetric encryption by a public key exchange to tunnel peers. When the tunnel is scrolled, each I2P peer eliminates a layer of encryption until the Garlic message reaches the output end-point tunnel. The final point transfers the output message to the input gate. The input gate will transfer the Garlic message to the real receiver until each peer within the tunnel add encrypted layers by symmetric keys of those only the receiver is able to remove all encrypted layers from the Garlic message [5].

Riffle Network

The Riffle network includes small set of identity anonymizing servers that ensure anonymity between authorized users until at least one guaranteed server is presented [5].

The Riffle network is a new approach composed of hash verification, private data recovery for the bandwidth, and computing anonymous communication functionality. The attacker is able to target the Riffle by the ToR node via manipulating some servers and using malicious code.

To prevention of these attacks, the Riffle uses verification and authentication approaches which are located above the ToR stack. This approach presents a verifiable statistical report for the same received or sent the message.

When the secure connection is established between the servers, the system uses encryption and identity verification to confirm the encrypted message by a less computing power, and on the contrary it provides high transmission speed rather than ToR network. By using ID verification mechanism, even the malicious servers are not able to disrupt the network and communications. They need to disrupt message correctly, hence only verified servers are able to receive. Therefore, the network would be safe as long as a single and unique server is presented in the anonymizing network in Riffle [10].

File transfer in Riffle is $\frac{1}{10}$ of the time needed for the same operation in ToR and other anonymizing networks [10]. In addition to, the Riffle is able to access 100 kbps bandwidth per user in a set of 200 users and also is able to respond to 100,000 users in microblogs with less than 10-second latency [10]. Similar to described anonymizing networks, the Riffle is expressed to traffic analysis attacks. Two measures caused the traffic analysis attacks to be limited in Riffle network: firstly, DC-nets which technically suggest the information for users and servers safely; secondly, the verifiable mix-nets that are based on hybrid and complex patchworks. In this scheme, the mix sets use disruption for replacing the ciphertext.

Same as DC-nets, the verified mix-nets guarantee anonymity. A DC-net is overloaded by many processes and only scalable for a few thousand users. On the other hand, the verified mix-net allow the user to send a message according to the message size. Therefore, a significant improvement could be observed in bandwidth usage. Although high computation and disruption overhead are guaranteed, the verified mix-nets are prevented from high bandwidth connections. The Riffle considers the problems of mix-nets and DC-nets, while it suggests the same amount of anonymity. The high levels in Riffle are organized as user-server structure. This network has been focused on minimizing bandwidth interface like smartphone and reduces computation overhead of the server and provides support for more users. Obviously, the Riffle users have an appropriate bandwidth given to message size and the members, so the server computation requires symmetric key encryption in common cases. This process allows the user to exchange the message and makes the system suitable for the application of effective file transmission. It is noteworthy that so far, the identity anonymizing systems have not the expected support from anonymity for all users, as well as the servers [9], [10].

The effective bandwidth and efficient computations in Riffle are caused by two factors: first for the verified disruption and a new combination of upstream communication; and second for Private Information Recovery (PIR) for downstream communications.

The current identity anonymizing networks evaluate the computations and bandwidths by the broadcast of all messages to all users via limited computation-high bandwidth or by an expensive PIR computation, high commutation-limited bandwidth. The PIR-based model is able to minimize the download bandwidth by minimizing computation overhead.

Although most communication anonymizing system relies on protecting transmitter anonymity, PIR protects the receiver privacy. In PIR, the user has access to some data via management server or a set of servers which are intended to hide regulatory data. There have been various PIRs for different settings, but some of the PIR strategies have complicated method and rules. In this line, Table 1 show the abbreviations which have been used in Riffle network.

Table 1: Samples of Calibri sizes and styles used for formatting a pes technical work

Terminology	Description
C	Set of Riffle clients
n	The number of clients
C_i	The $i-th$ Riffle client
S	Set of Riffle servers
m	The Number of servers
P_j	Public keys where $j \epsilon [1, m]$
b	Size of a message
π_i	The $i-th$ permutation function
f	The file name
$H_f^{\rightarrow}$	Hashes of all blocks relevant to file f
$H_j \in H_f^{\rightarrow}$	Hash blocks of flie f requested by client C_j
$H_\pi^{\rightarrow}$	Permutated available hash blocks
$\mathrm{M_j}$	$j-th$ plaintext message
r	Number of round
λ	Security parameters

Also, the Fig. 4 illustrates the deployment model of Riffle network. As shown in Fig. 4, the Riffle system is composed of a user-server structure. Users are considered as a set of individuals who are intended to communicate anonymously and the servers are considered as anonymous service. For replacement operations, each server is considered as a separate member or subject. Each user will communicate in a Riffle with a priority service, which is the main server based on the parameters such as host organization and location. The most important and valuable source in Riffle configuration is the bandwidth between user and server, supplying a high bandwidth network between a few servers is a common and feasible process. However, the high bandwidth could not be expected by all users due to their connection mode and the network infrastructure. Therefore, a Riffle has been focused on minimizing the requirements and the bandwidth between users and servers.

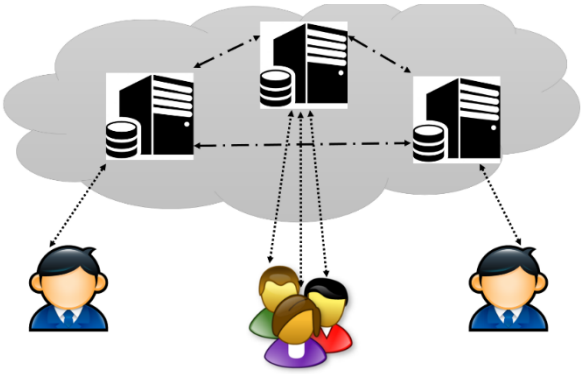

Fig. 4: Deployment model of Riffle [10].

Similar to I2P, a Riffle network tries to prevent traffic analysis attacks. Thus, the Riffle connections are dismissed in a cycle alternately. In both cases, each user receives and sends a message even the non-accession connection existed.

Riffle Protocol

In the following, the Riffle protocol is discussed. During the installation phase, the users establish three sets of ciphers which are coupled with the servers as follows: the $\{K_{ij}\}$ key uses the guaranteed disruption, $\{S_{ij}\}$ and $\{m_{ij}\}$ use a simpler approach such as Diffie-Hellman in PIR [33].

Each server generates π_i permutation for a guaranteed disruption and keeps them for the next use according to connection phase. The $\{K_{ij}\}$ key would be placed in S_i at $\pi_{i-1}\left(\dots\left(\pi_1(j)\right)\right)$ when the installation was done.

In the *r* cycles of connection phase, the protocol uses hybrid shuffle and PIR or distribution for loading and downloads respectively. In loading step, each user C_j encrypts a message by $\{K_{ij}\}$ key where $i \in [m]$; and loading the encrypted cipher text is done in $S_1$by the main server C_j. At disruption step which is began by S_1, each server S_1 verifies the cipher text and encrypts them by $\{K_{ij}\}$ key where $i \in [n]$; and they are stored during the installation phase via π_i permutation; then, the results are transmitted to the server. This also means that the ciphertext C_j in S_i, $\left(\dots\left(AEnc_{kij,r}\left(m_j^r\right)\right)\dots\right)$ which is confirmed by K_{ij} key. The final server shows the ciphertext for all servers. the final permutation of the message is according to $\pi = \pi_m\left(\pi_{m-1}\left(\pi_2(\pi_1)\right)\dots\right)$.

Bandwidth Overhead

The Riffle has been achieved with the bandwidth optimization between the user and the server during

message transmission. The ciphertext from encryption architecture is loaded based on the verified ID layer with $b + m\lambda$ scale. The parameters of this equation are taken from Table 1. If the user is interested in a particular message and the indicator was known; then, the transmission overhead would be included coverage n and the number of users to the main server. The high bandwidth stream $b + m\lambda + m$ as well as low bandwidth stream would be b for each user and cycle.

It should be noted that also the high bandwidth is grown linearly to n; it needs only one bit per user. When a message was anonymous, the download bandwidth like nb for each user. While the loading bandwidth would be decreased by n, the bandwidth requirement between the servers is increased linearly given to the number of users. Each server has to download and a loading n ciphertext (with a crossed off layer) to the next server. The last server has to send an empty text to the others. In addition, although PIR decreases that download bandwidth overhead for the users, it will increase the server to server bandwidth.

Hidden File Transmission

The Riffle makes these systems suitable for the functions of each compressed bandwidth such as file transmission. The file transmission is similar to Bit-Torrent network with a few detailed differences. In a Riffle, the user-server model is presented, but the Bit-Torrent has a peer-to-peer structure. While a user is intended to share a file, he/she will generate torrent file by which is including mixed string off all the file blocks. Then, the user loads the torrent file in the server via Riffle. The servers play the role of the torrent tracker in a Bit-Torrent network and manage all available files on that group. In the simplest design, the file descriptor is propagated to all users and they were able to choose the download directly. Although the torrent files sharing have an expense at once, their distribution would not cost so much the users share the files by distributed torrent files via the Riffle anonymously. Therefore, there are three basic steps:

Firstly, it is Block Request: Each client C_j chooses the file f arbitrarily and disrupts the file block $H_f^{\rightarrow}$ by torrent file. Then, the C_j requests a block by using the Riffle and loading the file $H_j \in H_f^{\rightarrow}$ to S_{pj}. When the user has no block to request, the user sends a dummy non-request message to retain traffic analysis resistant. In this case, all requests in $H_{\pi}^{\rightarrow}$ are dispersed to the users at the end of each round.

Secondly, it is Block Loading: Each client C_j investigates whether the requested block bt disrupted files via $H_{\pi}^{\rightarrow}$ or not. If a matched block such as M_j was found, then, C_j loads the M_j by the Riffle. While the blocks with plaintext were available for the servers, each server will propagate the disrupted file of the existed block $H_{\pi}^{\prime\rightarrow}$.

Thirdly, it is Block Download: Each client C_j uses the H_j to find the $H_{\pi}^{\prime\rightarrow}$, which is the I_j index of the requested C_j block. Then, client C_j downloads the blocks by PIR. Fig. 5 shows the model of anonymous file transmission protocol in Riffle [9].

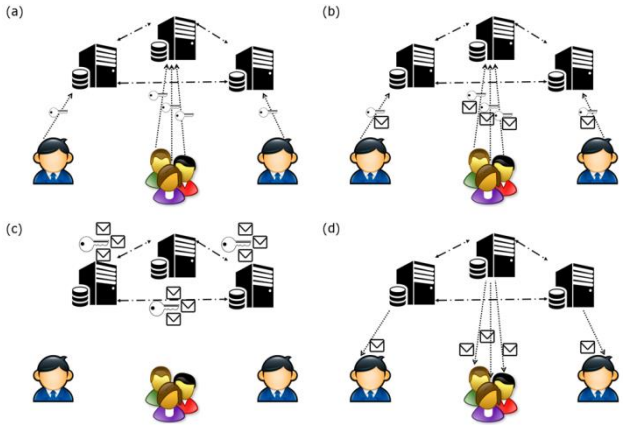

Fig. 5: Hidden File Transfer Protocol [9].

Fig. 5-a, is associated to setup phase in which users share the torrent files anonymously. Fig. 5-b, is related to request phase when a user requests a file upon uploading the hash of the file by utilizing Riffle. Fig. 5-c, is for upload phase when a user uploads an encrypted file according to requests by applying of Riffle. The download phase is depicted in Fig. 5-d, where a user downloads the file that he/she requested via PIR protocol.

Comparison between Identity Anonymizing Networks

There are obvious and various differences between I2P and ToR networks. ToR is based on the services which are provided voluntarily for generating circuit modes while I2P uses the counterparts with a sufficient functionality profile. Also, ToR network has been designed by many routers for optimization in output traffic where the I2P network has been designed for providing intermediate network services and facilities of only one set of output proxies. However, both methods present stable and low latency anonymity. In the following a comparison is given for important aspects of identity anonymizing communication networks:

ToR network uses the safe port interface and socks. In this field, the network is functioned as a proxy server. This expresses that the practical applications are able to use socks without any changes. On the other side, I2P is a middleware that provides the practical applications of the network or communication. Utilizing socks for ToR has two negative aspects: first, the socks interface is able to measure the message over TCP connection while the

I2P has to choose between TCP and UDP. So, I2P has a better performance in particular applications. Second, the message sent by applications may have information which would recognize the transmitter. In order to prevent this process application-level proxies with filtering capability such as Privoxy, have been used.

I2P and ToR have various capabilities. In fact, most I2P applications are designed to access intermediate network services exclusively. On the other hand, ToR network is able to use any program which is configured by proxy socks. Different encryption layers have been used in these three aforementioned networks which begin with transmission layer encryption and require TLS connection maintained by onion routers or I2P counterparts. I2P has additional functions in tunnel encryption. The message which is transmitted by the networks may be encrypted by onions or Garlic encryption. This means that the user connection to tunnel or the circuit would be completely anonymous within every anonymizing network such as the end-to-end encryption in I2P. However, the end-to-end encryption in ToR and other networks has no warranty due to the transmission layer protocol. Therefore, insecure protocols would not be used in these networks whilst a malicious or manipulated output node may store the message in plaintext and restore usernames and even the passwords.

In ToR network, only the first OR in the circuit knows the IP addresses of the real user; all other onion routers only know that before and after routers. User anonymity in ToR is significantly depending on ToR nodes selection algorithm. About I2P, even the first peer has no information about its transmitted data to another peer. In contrast to ToR, I2P does not need an input protector. ToR browser has a better performance compared with I2P except for the HTTP-GET-Request request. Increasing of users in anonymizing networks influence directly on ToR, I2P, and Riffle networks. Conflict and latency effect on the user experience and network usability, but the self-cover traffic for anonymity would be stronger. ToR is not well distributed like the I2P network. The routing in ToR is circuit-based while the routing in I2P drives the stack by implicitly load balancing and prevents any crash or delay in the system. This case is appropriate for high volume file transmission and could highlight I2P network. Two-step identity verification in Riffle makes this network safer and faster than another identity anonymizing network such as I2P and ToR networks. On the other hand, Riffle guarantees the transmitter anonymity by using PIR and protects the receiver's privacy. The most important requirements in these three networks are that the bandwidth supplying between the server is growing linearly given to the number of users. From functionality and performance point of view, Riffle would be better than other identity anonymizing networks for applications with compressed the bandwidth such as file transmission.

Onion Routing (OR), I2P, and Riffle are the anonymizing networks for tunneling issues. Low privacy frameworks can tunnel their information exchange by aforesaid networks. The main difference among the I2P and OR networks is periphery threat specimen and the exterior body of the proxies design. ToR is a directory based approach. So, it has a centralized point to lead the general network with information gathering and report abilities. This case is opposite of the other anonymizing networks that worked based on distributed DBMS. On the Other side, I2P has vulnerabilities such as traffic analysis. The attackers can analysis the traffic when the data came out from the mixed networks. This issue can be done by WATERHOLE and man-in-the-middle (MITM) attacks that made a suitable background to sniff the user's real time communications [34]. In forthcoming subsection, comparison between networks are performed and tabulated.

Comparison of ToR and I2P Terminology

The differences between the ToR and I2P idioms are described in Table 2. The comparison is determined based on the type of network, data transmission policy, kind of routing, etc. [35].

Note that, both I2P and ToR proxy performance have some drawbacks versus especial types of attackers. The used proxies are vulnerable to misuse besides several security penetrates. Although both of them have the same similarity in some cases, their utilizing terminology are rather different in which the deference idioms are tabulated.

Table 2: Differences between the ToR and I2P idioms [35].

Tor	I2P
Cellule	Packet
User	Customer
Circuit	Tunneling
Index	NetDb
Index Server	Router flood fill
Input Supervisor	counterpart
Entrance Point	Entrance Proxy
Egress Point	Egress Proxy
Hidden Service	Lease-Set
Primitive Point	Entrance Gate
Volunteer Users	Router
Onion Proxy	Gate
Point of assignation	Entrance Gateway and Egress point
Onion Service	Conceal the service

Comparison of ToR, I2P, and Riffle Terminology in Terms of Network Requirement

In addition to, a comprehensive comparison of requirement analysis in ToR, I2P, and Riffle networks is proposed on Table 3. The comparison is based on several network and distribution requirements.

As Table 3 shows, the Riffle outperforms against other two networks in terms of reliability, confidentiality, response time, and other network and distribution systems' requirement features.

Table 3: Comparison between ToR, I2P and Riffle in terms of analytical aspects

Network Type / Requirements	ToR	I2P	Riffle
Reliability	safe	Insecure	more safe
Confidentiality	Confident	Unconfident	more confidently
Availability	accessible	accessible	inaccessible
Usability	Easy	Medium to hard	Normal
Reusability	Yes	Yes	Yes
Cost	free	Non-free	Non-free
Response Time	Medium	Faster	Too Fast
Functionality	Poor	Fast	Faster
Security	Confident	Low Security	Safe
Reputation	Popular	Less popular	Less popular
Performance	efficient	efficient	efficient
Accessibility	available	attainable	attainable
Scalability	unchangeable	changeable	changeable
Adaptability	Inconsistent	Inconsistent	consistent

Comparison of ToR, I2P, and Riffle Terminology in Terms of Analytical Aspects

Also, Table 4 is dedicated to a comparison between anonymizing ToR, I2P, and Riffle networks in terms of analytical aspects and each of literature which paid for.

In terms of analytical aspects, ToR, I2P, and Riffle have competition in some comparison parameters. For instance, in term of upload speed, Riffle is the best but in term of memory usage and utilization, the ToR beats other networks.

Table 4-a: Comparison between ToR, I2P and Riffle in terms of analytical aspects

Network Type	ToR	I2P	Riffle	Ref.
Operational infrastructure	User base	Server base	Server base	[9], [24], [26], [36]
Extra abilities	Hidden web	No extra abilities	Hidden web	[7], [9], [10], [12], [24], [25]
Funding	Considerable funding	Without funding	Limited	[9]-[11], [24], [26]
Developers	More Developers	Limited	Limited	[9]-[11], [24], [26]
transport layer	TLS and Bridge	TLS	TLS	[9]-[11], [24], [26]
Scalability	High	Low	Low	[9]-[11], [24]
Switching	Circuit switched	Packet switched	Packet switched	[9]-[11], [24]
circuits	bidirectional circuits	Unidirectional Tunnels	Unidirectional Tunnels	[9]-[11], [24]
Upload speed	Low	Medium	High	[9]-[11], [24], [26]
Prone to DoS attacks	Larger enough for prevention	Smaller enough to attack	Larger enough for prevention	[5], [6], [9], [10]
Documentation	efficient	inefficient	inefficient	[9]-[11], [3], [13]
Overhead	Low bandwidth	High bandwidth	Low bandwidth	[9]-[11], [24], [26]
Security Focus	Exit Node	Entire network	Sender node	[9], [10], [27], [36]

Table 4-b: Comparison between ToR, I2P and Riffle in terms of analytical aspects

memory usage	Efficient	Inefficient	Inefficient	[9]-[11], [24], [26]
Distributed	Decentralized	centralized	Decentralized	[9]-[11], [24], [26], [21], [36]
Complexity	Reduced	Increased	Increased	[9]-[11], [13], [36]
Latency	Lower	Higher	Lower	[9]-[11], [24], [27]
Throughput	Higher	Lower	Higher	[9]-[12], [24]
Programming platform	C	Java	Probably python	[9]-[10], [27], [36]
Transport protocols	TCP	TCP & UDP	TCP & UDP	[9],[10],[26],[13], [27]
organizing	Server- organizing	Self-organizing	Self-organizing	[9],[10], [24], [36]
Directory servers	Safe	Unsafe	Unsafe	[9],[10], [24], [27]
User friendly	Very satisfying	desirable	Desirable	[9]-[12], [24], [37]

In terms of analytical aspects, ToR, I2P, and Riffle have competition in some comparison parameters. For instance, in term of upload speed, Riffle is the best but in term of memory usage and utilization, the ToR beats other networks.

Table 5: Bilateral comparison between ToR and I2P

Benefits of ToR over I2P	The extremely great user base Answered some obstacle that I2P has not yet to address them Considerable funding More developers TLS transport layer and bridges High scalability and resistance to attacks Circuit switched bidirectional circuits planned and optimized for exit traffic Better documentation, specifications, better website, and translations efficient memory usage low bandwidth overhead Centralized control reduces complexity at each node that can efficiently address Sybil attacks high capacity nodes higher throughput lower latency C language platform
Benefits of I2P over ToR	speedy usage of hidden services than TOR Fully distributed self-organizing peers selection by continuously profiling and ranking performance unvarying and untrustable directory servers Small enough to prone the DOS attacks Peer-to-peer friendly Packet switched implicit transparent load balancing Resilience Unidirectional tunnels Protection against detecting client activity short-lived all peers participate in routing for others The bandwidth overhead of being a full peer is low Integrated automatic update mechanism Both TCP and UDP transports JAVA language platform

Although I2P has great features such as anonymity and speed up in file sharing, there are some problems with its application which cannot be neglected. The I2P is a good framework for companies and organizations' interaction, but this system will drop the packets in two cases: the former is according to technical test with making distances between the received and sent packets; the latter case happens once the length of message exceeds from a certain value. This can be considered an important issue in the above network. Table 6 compares ToR and I2P in implementation details such as in security, scalability, and interfaces.

Table 6: Bilateral comparison between ToR and I2P

Titles	TOR	I2P
Implementation	Easy	Difficult
Scalability	Medium	Suitable enough
Stability	Stable	Unstable
Server Expert (difficulty)	Easy	Medium
Client Expert (difficulty)	Easy	Medium
Security	Medium	Better (with considering attack prevention)
Startup Time	Less	More
GUI features	No	Yes
Speed	As Expected	As Expected
Error rate	Seen	Not Seen
URL Stability	Auto Changed	Not Changed

Investigation over three famous anonymizing networks ToR, I2P, and Riffle reveal that they make fortunities to hide network connections for the sake of security and confidentiality reasons. In some cases, they dissipate network bandwidth and sacrifice network performance to reach security objectives. Nevertheless, some proposed protocols and bandwidth optimization techniques can improve network quality of service (QoS) besides reaching concealing justifications and security objectives. Furthermore, since each anonymizing network has its own merits and demerits, the hybrid network within its protocols design which inherits all of plus points and excludes negative features is favorable.

Results and Discussion

This paper presented a profound comparison between three famous anonymizing ToR, I2P, and Riffle networks which tend to camouflage services from their users. To do so a subjective classification plan for comparison among anonymizing networks has been presented. The comparison has been done based on commonalities, discrepancies, and challenges in this field. This comparison is beneficial for further researches and future improvements to fill the existing gaps. However, by this review, this can be pointed out that the ToR is a popular network whilst other networks like Riffle and I2P are relatively considered novel alternatives. The mentioned systems are updated continuously for performance improvement and also providing more anonymity to protect the users. ToR and I2P differences are in preparation and using virtual communications. ToR network is a set of volunteer servers all over the world which are functioned for anonymous connection to browsing a web page or some particular operations. However, I2P provides anonymous file transmission between two peers. In data transmission, Riffle is faster than other identity anonymizing networks due to using two-step identity verification structure, and the traffic analysis is barely feasible. Also, Riffle has more efficiency because it provides a significant anonymity due to minimizing bandwidth and computation overhead. Totally, anonymizing networks can utilize efficient protocols and bandwidth optimization techniques that can potentially improve network QoS besides reaching concealing justifications and security objectives. Furthermore, as each anonymizing network has its own merits and demerits, the hybrid network within its protocols design which inherits all of plus points and excludes negative features is favorable.

Author Contributions

Dr. Mirsaeid Hosseini Shirvani was the supervisor of the current research plan. He sketched the research framework and the roadmap. Also, he analyzed the results and tabulated the outcome derived from excerpted literatures. In this line, Amir Akbarifar searched in authentic journals to gather all relevant papers. In addition to, he prepared the blueprint of the research plan. He and his supervisor cooperatively summed up the work.

Acknowledgment

This work is completely self-supporting, thereby no any financial agency's role is available.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviation

I2P	Invisible Internet Project
ToR	the Onion Router

OR	Onion Router
DC	Dining Cryptographers
OP	Onion Proxy
TLS	Transport Layer Security
TCP	Transmission Control Protocol
SHA1	Secure Hash Algorithm 1
P2P	Peer-to-Peer
DNS	Domain Name Resolver
DC-nets	Dining Cryptographers
PIR	Private Information Recovery
UDP	User Datagram protocol
MITIM	Man-in-the-Middle attack
QoS	Quality of Service

References

[1] M.S. Hosseini Shirvani, A.M. Rahmani, A. Sahafi, "An iterative mathematical decision model for cloud migration: a cost and security risk approach," Software Pract. Ex., 48(3): 449-485, 2018.

[2] M.S. Hosseini Shirvani, "To move or not to move: An iterative four-phase cloud adoption decision model for IT outsourcing based on TCO," J. Soft Comput. Inf. Technol., 9(1): 7-17, 2020.

[3] M.S. Hosseini Shirvani, "Web Service Composition in multi-cloud environment: A bi-objective genetic optimization algorithm," in Proc. 2018 Innovations in Intelligent Systems and Applications (INISTA): 1-6, 2018.

[4] The Center for Internet Security (CIS), "The CIS Security Metrics," v1.0.0, 2010.

[5] B. Conrad, F. Shirazi, "A survey on Tor and I2P," in Proc. ICIMP 2014: The Ninth International Conference on Internet Monitoring and Protection: 22, 2014.

[6] D. Goldschlag, M. Reed, P. Syverson, "Onion routing for anonymous and private internet connections," Commun. ACM, 42(2): 39-41, 1999.

[7] G.H. Owenson, N.J. Savage, "The Tor darknet," Global Commission on Internet Governance," paper series no. 20, 2015.

[8] http://www.geti2p.net

[9] https://github.com/kwonalbert/riffle

[10] A. Kwon, D. Lazar, S. Devadas, B. Ford, "Riffle: An efficient communication system with strong anonymity," Proceedings on Privacy Enhancing Technologies, 2: 115-134, 2016.

[11] F. Shirazi, M. Simeonovski, M.R. Asghar, "A survey on routing in anonymous communication protocol," ACM Comput. Surv. (CSUR), 51(3): 1-39, 2018.

[12] B. Li, E. Erdin, M.H. Gunes, G. Bebis, T. Shipley, "An overview of anonymity technology usage," Comput. Commun., 36(12): 1269-1283, 2014.

[13] E. Jardine, "The dark web dilemma: Tor, anonymity and online policing," Global Commission on Internet Governance Paper Series 21, 2015.

[14] P.F. Syverson, D.M. Goldschlag, M.G. Reed, "Anonymous connections and onion routing," in Proc. IEEE Symposium on Security and Privacy (Cat. No.97CB36097): 44–54, 1997.

[15] M.W. Al-Nabki, E. Fidalgo, E. Alegre, L. Fernández-Robles, "ToRank: Identifying the most influential suspicious domains in the Tor network," Expert Syst. Appl., 123: 212–226, 2019.

[16] Q. Yang, P. Gasti, K. Balagani, Y. Li, G. Zhou, "USB side-channel attack on Tor," Comput. Netw., 141: 57–66, 2018.

[17] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, S. Devadas, "Circuit fingerprinting attacks: Passive deanonymization of tor hidden services," in Proc. 24th USENIX Security Symposium (USENIX Security 15): 287–302, Washington, D.C., USENIX Association, 2015.

[18] R. Attarian, L. Abdi, S. Hashemi, "AdaWFPA: Adaptive online website fingerprinting attack for Tor anonymous network: A stream-wise paradigm," Comput. Commun., 148: 74–85, 2019.

[19] K. Sangeetha, K. Ravikumar, "Defense against protocol level attack in Tor network using deficit round robin queuing process," Egyp. Inf. J., 19 (3) 199–205, 2018.

[20] B. Monk, J. Mitchell, R. Frank, G. Davies, "Uncovering tor: An examination of the network structure," Secur. Commun. Netw., 2018: 1-13, 2018.

[21] A.S. Tanenbaum, "Distributed operating systems," Pearson Education India, 1995.

[22] M.S. Hosseini Shirvani, N. Amirsoleimani, S. Salimpour, A. Azab, "Multi-criteria task scheduling in distributed systems based on fuzzy TOPSIS," in Proc. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, Canada, 2017.

[23] Y. Ramzanpoor, M. Hosseini Shirvani, M. Golsorkhtabaramiri, "Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure," Complex Intell. Syst., 1-32, 2021.

[24] G. Danezis, C. Diaz, "A survey of anonymous communication channels," Technical Report MSR-TR-2008-35, Microsoft Research, Jan. 2008.

[25] R. Dingledine, N. Mathewson, P., Syverson, "ToR: The second-generation onion router," in Proc. 13th conference on USENIX Symposium., 13: 1-21, 2004.

[26] D. McCoy, k. Bauer, D. Grunwald, T. Kohno, D. Sicker, "Shining light in dark places: Understanding the Tor network," in Proc. the 8th International Symposium, PETS 2008 Leuven, Belgium, 2008.

[27] H.Y. Huang, M. Bashir, "Who is behind the Onion? Understanding Tor-Relay operators," CyLab Usable Privacy and Security Laboratory (CUPS), 2016.

[28] https://geti2p.net/en/docs/how/tech-intro.

[29] A. Crenshaw, "Common darknet weaknesses: An Overview of Attack Strategies," DEFCON 19, Las Vegas, August 6, 2011.

[30] M. Wahal, T. Choudhury "Anonymous network routing mechanism," in Proc. International Conference on Infocom Technologies and Unmanned Systems, Trends and Future Directions (ICTUS), 2017.

[31] P. Mayamounkov, D.M. Eres, "Kademlia: A Peer-to-peer information system based on the XOR Metric," International Workshop on Peer-to-Peer Systems. Springer, Berlin, Heidelberg, 2002.

[32] H. Niedermayer, "Architecture and components of secure and anonymous peer-to-peer systems," Ph.D. dissertation, Dept. Computer Science, Univ. Munich, 2010.

[33] W. Diffie, M. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theory, 22(6): 644-654, 1976.

[34] M. Rouse, "What is watering hole attack?," Retrieved March. 2015; 6: 2019.

[35] https://geti2p.net/en/comparison/tor

[36] M.G. Reed, P.F. Syverson, D.M. Goldschlag, "Anonymous connections and onion routing," IEEE J. Sel. Areas Commun., 16(4): 482-494, 1998.

[37] Z.J. Newman "A high-bandwidth, low-latency system for anonymous broadcasting," Diss. Massachusetts Institute of Technology, 2020.

Biographies

Mirsaeid Hosseini Shirvani received his B.Sc., M.Sc., and Ph.D. all in Computer Software Engineering Systems at Universities in Tehran, IRAN. He has been teaching miscellaneous computer courses in several universities in Mazandaran province of IRAN since 2001. He also published several papers in authentic and worldwide well-reputed journals. Currently, he is an Assistant Professor in Computer Engineering Department at IAU (Sari-Branch). His research interests are in the areas of cloud computing, fog computing, IoT, distributed systems, parallel processing, machine learning, and evolutionary computation.

- Email: mirsaeid_hosseini@iausari.ac.ir
- ORCID: 0000-0001-9396-5765
- Web of Science Researcher ID: AAO-2012-2021
- Scopus Author ID: 55459128300
- Homepage: NA

Amir Akbarifar received his B.Sc. and M.Sc. in Computer Software Engineering Systems in Islamic Azad University in IRAN. Currently, He is a PhD candidate in Computer Engineering Department at IAU (Sari-Branch). He has published numerous articles in prestigious magazines in Iran. Amir is information Security Analyst and his fields of study include: Security, Software Architecture, Artificial Intelligence, Deep Learning, Quantum Computing, and Cryptography.

- Email: msc.akbarifar@gmail.com
- ORCID: 0000-0003-3227-2847
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

How to cite this paper:
M. Hosseini Shirvani, A. Akbarifar, "A comparative study on anonymizing networks: TOR, I2P, and riffle networks comparison," J. Electr. Comput. Eng. Innovations, 10(2): 259-272, 2022.

DOI: 10.22061/JECEI.2021.8027.466

URL: https://jecei.sru.ac.ir/article_1630.html