Hindawi
Security and Communication Networks
Volume 2022, Article ID 2255047, 13 pages
https://doi.org/10.1155/2022/2255047

Research Article

A Covert-Aware Anonymous Communication Network for Social Communication

Xinda Cheng,[1] **Ning Hu**,[1,2] **Yue Zhao**,[3] **Kaijun Wu**,[3] **Jincai Zou**,[1] **and Yixing Chen**[1]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China*
[2]*Peng Cheng Laboratory, Shenzhen 518000, China*
[3]*Science and Technology on Communication Security Laboratory, Chengdu 610041, China*

Correspondence should be addressed to Ning Hu; huning@gzhu.edu.cn

Received 31 January 2022; Revised 30 April 2022; Accepted 7 September 2022; Published 15 November 2022

Academic Editor: Ilsun You

To effectively protect the communication content and communication behavior of social networks, anonymous communication technologies are widely used. However, the anonymous communication networks represented by Tor and I2P lack the covertness of the control plane design, which leads to important user behavioral characteristics in the process of accessing anonymous communication networks. Therefore, network monitors can analyze users' communication behavior by tracking these characteristics. In this paper, the concept of covert measurement is proposed. On this basis, a software-defined anonymous communication network architecture is presented, which also considers the covertness of the anonymous communication network control plane and the data plane. According to the theoretical analysis and experimental results, the anonymous communication network architecture proposed in this paper has better anonymity and usability than traditional anonymous communication networks, such as Tor.

1. Introduction

The rapid development of Internet technology has accelerated the integration of physical space and cyberspace. An increasing number of social behaviors and economic behaviors are deeply dependent on the Internet environment. However, due to the lack of security mechanisms and regulatory mechanisms, users will inevitably be subject to network traffic monitoring while using the Internet for communication or data transmission and, thus, reveal business secrets and personal private information. Even if virtual private network (VPN) technology based on an encryption algorithm is adopted, it still faces threats, such as malicious blocking and source traceability of communication behavior. While maintaining the confidentiality of information content, anonymous communication networks introduce the concept of anonymity to prevent network listeners from analyzing and tracing network communication behaviors, such as Tor [1] and I2P [2]. In recent years, attacks and metrics around anonymous communication network systems have become a new topic.

Anonymous communication networks can provide anonymity protection for personal communication in the Internet environment. Unfortunately, existing anonymous communication networks generally have problems, such as relatively simple communication architecture, difficulty in guaranteeing node reliability, and inability of resisting large-scale advanced persistent threat (APT) attacks [3]. The vulnerability of the anonymous communication network itself may reduce its ability of protecting users' communication behavior. For example, in Tor and I2P, to obtain network status, the client needs to exchange information with a specific control node, thereby exposing the client's behavioral intention. From the information transmission perspective, although the anonymous communication network can hide the communication content and users' communication behavior, due to the lack of covertness of the anonymous communication network control plane, users may have domain names of control nodes while using the anonymous communication network. With weaknesses, such as fixed IP and obvious traffic characteristics, the attacker can use these characteristics to infer the user's communication behavior and maliciously block it [4]. Therefore, the covertness of anonymous communication networks in the process of access and information transmission has also become an important indicator that developers of anonymous communication networks need to consider in their designs. Accurately measuring the covertness of anonymous communication networks has also become a problem worth considering.

In practical applications, to prevent supervisors from monitoring or proactively detecting traffic, security researchers usually use different methods to hide various features in the two stages of client access to the anonymous communication network and network traffic forwarding to realize the covertness of the anonymous communication network system. Users can use the following three methods to access anonymous communication networks: restricted resource acquisition, proxy forwarding, and traffic obfuscation. Tor uses emails to provide a very small list of nodes and limits the access rate of related nodes [5] by using key-based space division [6], social network-based [7], and variable speed proxy [8]. Moreover, manually importing a small number of time-effective seed file addresses [9] in I2P to find P2P nodes all belongs to the method of restricting resource acquisition to prevent communication behavior from being discovered. The bridge node used in Tor hides the recipient of the data by proxy forwarding of the authoritative directory server [10]. The various generations of OBFS [11] meet the needs of major anonymous communication networks and proxy servers for traffic confusion. With the development of web technology and cloud computing, some services of large cloud service providers are applied to covert communication systems to hide one or more parts of the communication. For instance, Meek technology used in Tor to hide authoritative directory servers and various nodes [12], a series of out-of-band requests and steganography techniques, such as dnslog [13], is used to hide real data, the domain fronting technology widely used to hide real requests [13], and the popular domain borrowing technology for high reputation domain names and their certificate utilization. The method of ensuring covertness in the network traffic forwarding phase is mainly based on countertraffic analysis, including various methods to eliminate traffic time slots and message entropy features. These methods ensure the covertness of anonymous communication networks to a certain extent but do not fundamentally change the shortcomings of anonymous communication network architecture and reliability at this stage. Therefore, this paper proposes an anonymous communication network based on software definition, which achieves good covertness with the help of centralized architecture and reliable nodes. At the same time, to evaluate and compare the covertness between the anonymous communication network described in this paper and the decentralized anonymous communication network at this stage, this paper proposes an anonymous communication network covertness evaluation method based on a probability block diagram.

The main contributions of this paper are as follows:

(1) Aiming at the problem of covertness analysis of anonymous communication networks, an analysis model of security threats is proposed, and the ability constraint assumption is made for the listeners and users of anonymous networks. In theory, if the listeners of anonymous networks have God's perspective and strong unlimited capabilities, then any anonymous communication network cannot provide effective anonymous protection. Therefore, when analyzing the covertness of an anonymous communication network, it is necessary to provide a security threat model.

(2) An architecture and implementation proposal of a software-defined anonymous communication network system is proposed. Compared with anonymous communication networks, such as Tor and I2P, the anonymous communication network proposed in this paper has better usability and programmability, can be used for social network applications, and is customized for specific needs.

(3) A covertness measure is proposed and used to compare the system with onion networks. Analysis and experimental results show that the anonymous communication network proposed in this paper has better covertness and availability than traditional anonymous networks, such as Tor and I2P.

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 introduces the threat model, Section 4 proposes the anonymous communication network system measured in this work, Section 5 details the modeling method, and Section 6 presents the comparative experiment and, finally, summarizes and looks forward to the work described in the article.

2. Related Work

2.1. Anonymity Measurement. With the vigorous development of anonymous networks, anonymity assessment has attracted increasing attention from researchers. The mainstream idea of this problem is to put the identity of communication parties into an indistinguishable set, and all identities in the set may become the communication subject. The observer can only know the mapping relationship from one communication set to another but does not know the specific communication correspondence. By the anonymity assessment of an anonymous communication network, people can judge its specific performance in sending, receiving, and judging the communication relationship to hide each communication entity. Currently, researchers have developed, from the earliest qualitative analysis to quantitative analysis, the anonymity of anonymous networks from various angles. In 1998, Reiter [15] used the degree of anonymity to analyze the anonymity of anonymous communication networks based on mixed technology. Anonymity degree divides the order of network anonymity, which can be divided into six levels. Since then, researchers have proposed quantitative evaluation methods of anonymity according to different theories. Since the measurement method based on Shannon entropy [8] in information theory was proposed in 2002, many different measurement methods have been derived from the information theory and entropy measurement methods, such as normalized entropy [16], Rényi entropy [17], and conditional entropy [18]. Then, some scholars proposed methods based on time [11], game theory [19], and differential privacy [20] to measure the anonymity of anonymous communication networks. In 2018, Das et al. [21] proposed the triangle dilemma of an anonymous communication network, emphasized that only two of the factors of anonymity, delay overhead and traffic overhead, can be selected and utilized their evaluation model to evaluate the anonymous communication network based on onion routing.

2.2. Covert Access and Detection. Covert access includes confusion, encryption, and fields disguised as normal protocols to eliminate the traffic characteristics during software access. This paper introduces the technologies used in common anonymous communication networks, such as Tor. 1. Meek: its principle is to use the nonprohibited protocol as the tunnel to pass the Tor traffic in the tunnel. It uses domain fronting technology [22], utilizing HTTPS and CDN to bypass censorship. Meek detection is mainly based on machine learning. Shahbar and Zincir-Heywood [23] used the decision tree classifier to analyze the time span, the number and repeatability of connections, the amount of data transmitted and the number of connections established, or the packet size, the number of bytes sent, and the maximum packet size; then, the traffic can be identified after learning. Qureshi et al. [24] indicated that the total duration of TCP connections of normal HTTPS and meek is different and the length distribution proportion of TCP payload is also different. Zhao et al. [25] studied Tor traffic classification using the state-of-the-art algorithm, which included J48, J48Consolidated, BayesNet, jRip, OneR, and RRPTree. In addition, the entropy characteristic of Meek also has a certain effect in detection when using machine learning.

The Obfs includes obfs2, obfs3, obfs4, and Scrabblesuit [26]. At present, obfs4 is the most commonly used. Its principle is to encrypt the traffic, which makes it look like random bytes, to avoid fingerprint detection based on a blacklist. Obfs4 can combat active probing attacks [11] with key negotiation to prevent reviewers from utilizing connection discovery bridges. In terms of detection, Wang et al. [27] found that joint detection based on entropy detection and simple heuristic algorithms (such as length detection) can identify Obfs traffic. Other detection methods include packet length detection and truncated sequential probability ratio testing.

2.3. Covert Traffic Detection. At present, traffic detection technology [28] can be divided into the following four categories: (1) semantic-based detection, (2) entropy-based detection, (3) machine learning-based detection [25], and (4) combined detection [29]of DPI and firewall, which can reconstruct the complete traffic, analyze the specific protocol, identify the keywords of packets, and actively detect suspicious servers to avoid false-positives.

2.4. SDN in Anonymous Communication Network. The development of a software-defined network also provides a new optimization scheme for the traditional anonymous communication network, such as deploying the software-defined anonymous communication protocol of the network layer on the autonomous domain router of the network service provider (such as lap [30] and Phi [31]). These new anonymous communication networks separate the control plane from the forwarding plane, making the information transmission path programmable.

3. Threat Models

This paper assumes that there is large-scale supervision of ISPs in the network and that the relevant monitoring platform of each cloud platform server has been exposed to a supervisor. However, supervisors do not have direct control over individual hosts. Therefore, nodes in the Internet are defined in this paper as the following four different nodes: client nodes used for users to access the network, nodes used for forwarding information in anonymous communication networks (hereafter referred to as controlled nodes), malicious nodes controlled by supervisors (hereafter referred to as malicious nodes), and dazed third-party nodes.

For the client node, in the environment described in this paper, the supervisor can only see the traffic at the entrance and exit of the node but cannot obtain the control authority of the node. Therefore, the supervisor detects whether users use anonymous communication networks and try to obtain users' communication relationships and other information through wiretapping, recording, replay and traffic analysis, and other means.

For the controlled nodes, due to their wide distribution, this paper assumes that the supervisor can only monitor and analyze the controlled node state and its incoming and outgoing traffic in a certain physical area. However, it cannot obtain all the node states in the anonymous communication network. Nevertheless, the supervisor can add the malicious nodes under its control to the anonymous communication network to achieve a man-in-the-middle attack or a witch attack.

For malicious nodes, this paper only considers that a large number of malicious nodes controlled by supervisors are concentrated in one physical region and a small number are distributed in other regions. Beyond that, supervisors can only access data from some Internet infrastructure providers. Therefore, because anonymous communication networks are distributed all over the world, their design rules include the fact that each node in the path does not exist in the same country or region to ensure that the case that all traffic is tracked in a transmission path is not considered.

In this paper, the dazed third-party nodes mainly refer to some Internet infrastructure platforms with massive users and data files, such as web storage for storing media files, social platforms for publishing information, and various Git repositories for hosting code. This article assumes that the custodian has the same access rights as the user and does not have access to the user's usage records of these dazed third parties.

4. Anonymous Communication Network

4.1. Tor. Tor is a widely deployed and popular anonymous communication network and its main purpose is to prevent attackers from identifying communication parties or associating communication links with a single user. Tor is based on the P2P network architecture and uses the onion routing protocol. Its data are transmitted through a series of uncontrolled voluntary nodes in the Internet; that is, there are controlled nodes, malicious nodes, and dazed third-party nodes in the Tor network.

Tor works as follows: clients build a link by selecting entry, intermediate, and exit nodes. The Tor client obtains the current Tor network consensus file from the current Tor's authoritative directory server. This file contains basic information, such as the IP address, bandwidth and location of each forwarding node in the current Tor network, and the services supported by the node, and this information is updated every hour. The client selects three nodes from the nodes listed in the consensus file. For randomly selected nodes, the selection probability is approximately proportional to the bandwidth weight of the node. When creating and using links, layered encryption of onion routing ensures that each forwarding node only knows the information of the previous hop and the next hop in the link and no single forwarding node can transmit the client's information to the destination [32].

To improve the security and anonymity of services, Tor clients use different access protection mechanisms when they access Tor networks. In the client access process, Tor adopts a series of security mechanisms, such as bridge node [33], Meek covert channel construction, Obfs obfuscation, and FTE encryption, to protect user traffic from supervision during access. The protection mechanism during access is randomly selected by users, and the probability of the optional nodes obtained for link establishment is proportional to the bandwidth [34]. To be an optional node, a forwarding node must meet a number of selected criteria to ensure good performance and increase the cost of being attacked. The selection criteria are as follows: first, the forwarding node must be measured by Tor's bandwidth measurement system, which takes two weeks [23]. Second, the forwarding node must have enough bandwidth to make its weight reach at least 2000. The bandwidth value measured by Gerry Wan [35] et al. is approximately 35.5 Mbit/s. Third, the forwarding node must always be online to be considered a stable state. Fourth, forwarding nodes must remain online long enough to be considered familiar nodes.

4.2. Anonymous Communication Network Based on Software Definition. This paper proposes a software-defined anonymous communication network that can achieve good covertness. The network architecture is displayed in Figure 1. The access method based on Internet public service is used in the network of the user access stage. In the data transmission stage, the data will pass through two parts: an isolated network and a core network. The isolated network consists of a control center with multiple Internet infrastructures (dazed third parties). The core network consists of a control center and several controllable forwarding nodes distributed all over the world.

4.2.1. User Access. This stage is jointly completed by the client, the anonymous communication network (controller, access agent), and the dazed third party, realizing the process of establishing an implicit communication relationship with the dazed third party and obtaining response data files under the condition that the client and the servers in the anonymous communication network are unaware of each other. The specific process is demonstrated in Figure 2.

The working mechanism of the client, access agent, and the control center can be described by the following Algorithm 1.

The client in Figure 2 obtains the temporary address of the registry in out-of-band mode like SMS or hiding the key information in tiktok [36] and obtains the identity identifier, the public and private key pair, and the list of accessible access agents from the registry before accessing the communication network. After that, it downloads the real resource until it obtains the responses from the access agent.

The access agent forwards content from both the client and the controller Algorithm 2.

For the controller, when it receives the request from the access agents, it would verify the messages and send the real resource to the web storage. The address of this storage is sent to the access agent Algorithm 3.

Before accessing the anonymous communication network, the client obtains the temporary address of the registry in out-of-band mode and obtains the identity identifier, the public and private key pair, and the list of accessible access agents from the registry. When accessing the anonymous communication network, the access request message is sent to the access agent. After receiving the access request message, the access request message is forwarded to the control center server. Additionally, after receiving the access request message, the control center server packages the control information corresponding to the access request and saves it to a third-party web storage. After receiving the reply message from the authoritative directory server, the access agent returns the reply message to the client and notifies the client to read the control information from the specified third-party storage node. After receiving the response message from the access agent, the client reads the control information from the specified third-party storage node.

4.2.2. Isolate Network. Through the file exchange rather than the data streaming anonymous information based on the file name and the file encryption implementation content encryption, the isolation network transmission method based on the Internet of Basic Public Services implements anonymous communication users through asynchronous communication, fragmentation, and screen flow mechanisms and ensures traffic data from the client before entering the covertness of the anonymous communication networks.

At this stage, the data sent by the client is transmitted to each dazed third-party platform in the form of a file, and

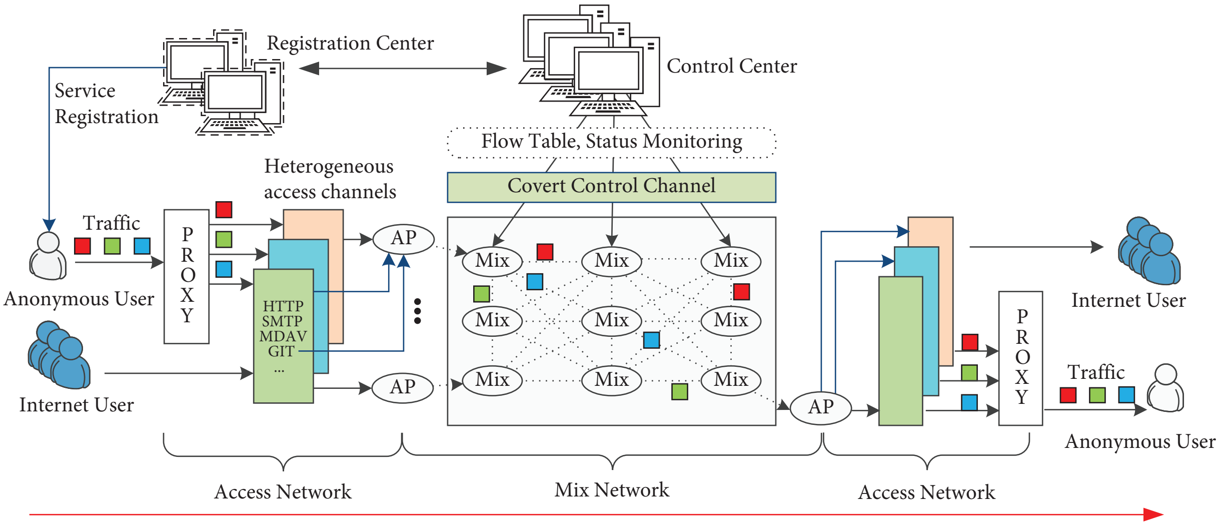

Figure 1: Architecture of the SD-anonymous communication network.

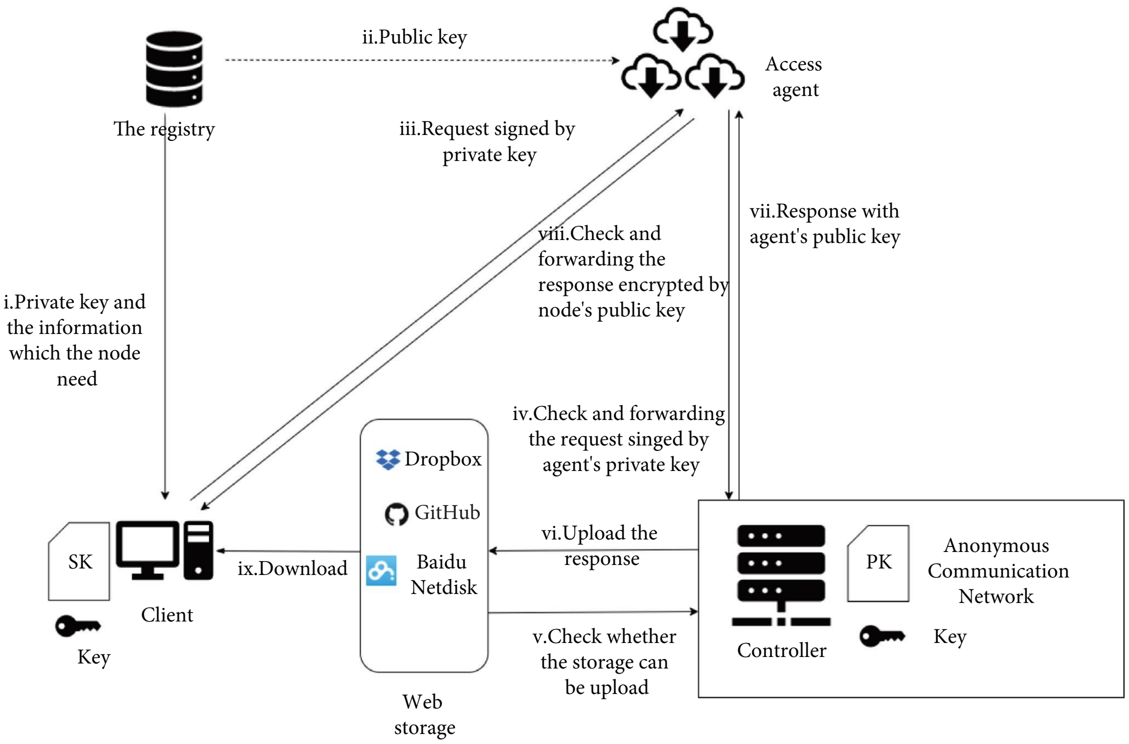
ii.Public key
The registry
Access agent
iii.Request signed by private key
vii.Response with agent's public key
viii.Check and forwarding the response encrypted by node's public key
i.Private key and the information which the node need
iv.Check and forwarding the request singed by agent's private key
Dropbox
GitHub
Baidu Netdisk
vi.Upload the response
SK
Client
Key
ix.Download
v.Check whether the storage can be upload
Web storage
Controller
PK
Anonymous Communication Network
Key

Figure 2: The specific process by which the node accesses the files when it joins the real network.

then the corresponding A-nodes in the core switching network obtain the file from the dazed third-party platform. Due to the public nature of the third party, the regulator's perspective cannot simply identify the controlled nodes and the transmission traffic.

4.2.3. Core Network. The core network borrows the idea of software definition and adopts the form of the controller and the controlled forwarding node to realize the programmable node and the forwarding path. To realize the covertness of the network communication, the system uses file exchange instead of message exchange to realize asynchronous communication. After being removed from the isolated network, the data to be transmitted are synchronized to each intermediate node in the form of a file, and the intermediate node transmits the data to the receiver according to the specified forwarding path. The two parties do not directly transmit encrypted traffic.

(1) Architecture. The system proposed in this paper consists of N nodes and K console servers and is shown in Figure 3.

As a communication user, a node also provides file storage and forwarding services for the anonymous communication of other nodes. Each node maintains *N* folders

```
Input: AA, $SK_{node}$, $PK_{AA}$
Output: res
K = random ()
sign = sign(ReqAccess,K,$SK_{node}$)//ReqAccess is client ask for access to the network
message = [ReqAccess, K, sign]
ciphertext = enc(message, $PK_{AA}$)
requset.setAddr (AA).setData (ciphertext)
request.send ()
waitForResponse ()
message = dec(response.data, $SK_{node}$)
If verify (message, $PK_{AA}$, K) == False then
    return error
res = download(message.address)
return res
```

Algorithm 1: client.

```
Input: request, controller, $SK_{AA}$, $PK_{node}$, $PK_{controller}$
Output:address
message = dec(request.data, $SK_{AA}$)
K = message.K
If verify (message, $PK_{node}$) == False then
    return error
sign = sign(message.ReqAccess, K,$SK_{AA}$)
message.sign = sign
ciphertext = enc(message, $PK_{controller}$)
requset.setAddr (controller).setData (ciphertext)
request.send ()
waitForResponse ()
If verify (response, $PK_{controller}$, K) == False then
    return error
message = dec(response.data, $SK_{AA}$)
address = message.address
sign = sign(address, K,$SK_{AA}$)
message = [address, K, sign]
ciphertext = enc(message , $PK_{node}$)
response = request.getResponse().setData(ciphertext)
response.send()
return res
```

Algorithm 2: access agent.

and $N-1$ backup files, where N folders correspond to each node i. The console server is the core of the system, which controls the IP addresses of all nodes and determines whether each node participates in the communication process. Then, the sender can set the forwarding route through the console server before communication. The console server can also control whether the node performs file synchronization. Due to the controller's large throughput, the system needs to use multiple controllers to prevent the supervisor from tracing the source.

As the core of the system, the controller server controls the IP address of all nodes.These nodes running in the internet are silent at first, and can be activated by the controller server for the communication process.. The sender can determine the forwarding route through the console server before communication. The console server can also control whether the nodes synchronize files.

(2) Route Selection. Route selection is performed by the controller selecting m controlled nodes ($m < N-1$.) or the sender selects m nodes to form the path R, which can be seen in the following equation:

$$R = \{N1, N2, N3, \ldots, Nm-1, Nm\}. \qquad (1)$$

In addition, the design of this route has the following constraints: it needs to go through different countries; it needs to go through different VPS manufacturers; and at least three controlled nodes must be passed. The control center server then sends synchronization configuration commands to each node, as depicted in Figure 4.

```
Input: request, $SK_{controller}$, $PK_{AA}$, webStorageList
Output: address
message = dec(request.data, $SK_{controller}$)
K = message.K
If verify(message, $PK_{AA}$) == False then
    return error
address = null
for webStorage in webStorageList do
    If checkAvailable( webStorage) then
        address = webStorage
        break;
upload(address RSP)// RSPcontains the network's cache and other resources
sign = sign(address, K,$SK_{controller}$)
message = [address, K, sign]
ciphertext = enc(message , $PK_{AA}$)
response = request.getResponse().setData(ciphertext)
response.send()
return res
```

Algorithm 3: controller.

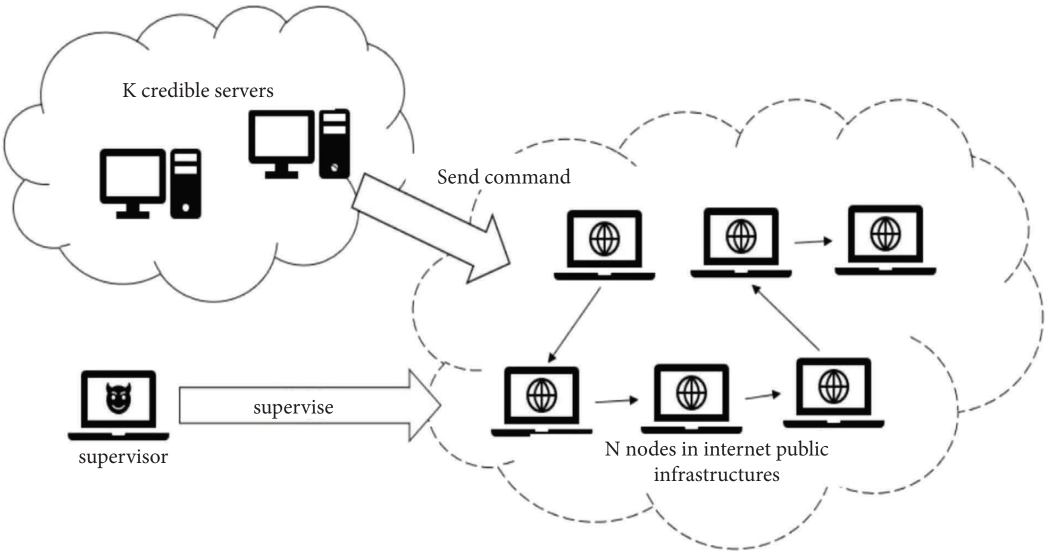

Figure 3: The core network.

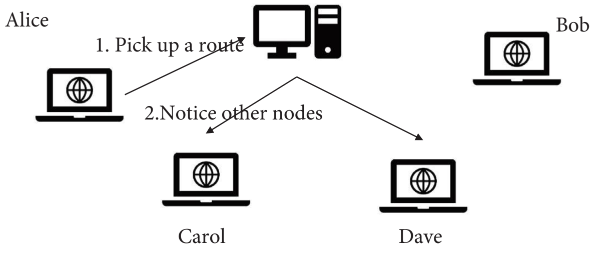

Figure 4: Route selection.

(3) Information Transmission. As shown in Figure 5, node A synchronizes information to the nodes in the forwarding path in the form of an A file, for example, the exchanged keys, but based on Wildcard identity-based encryption [37]. The nodes in the path synchronize information in turn until node B receives the file and returns the receiving identifier in the same way. In this process, the traffic identified by the supervisor is that node A communicates with another node C and Bob communicates with another node D.

Finally, A and B realize the complete communication process in the core switching network. During the whole process, BOTH A and B perform file synchronization operations with multiple nodes, masking the real traffic transfer information. Third parties cannot track specific data traffic.

4.2.4. Security Analysis. The background of the proposed system is to build an anonymous communication system implemented by controllable nodes at the application layer in an uncontrolled network. It shields all information below

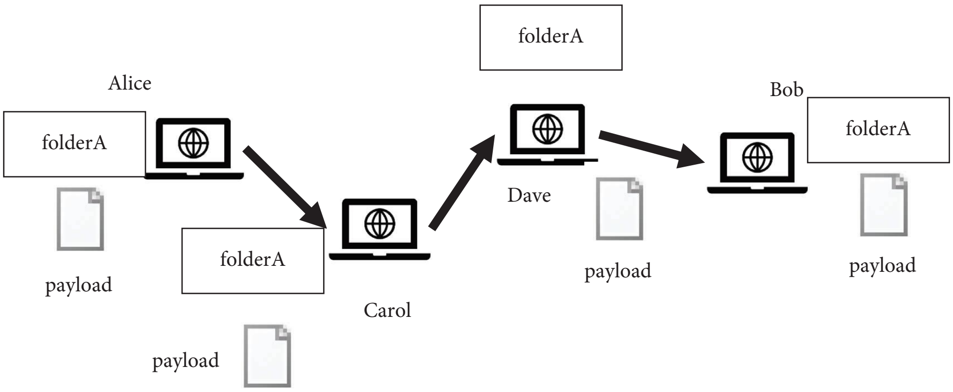

Figure 5: Information transmission.

the application layer. Except for the two parties of the communication, other users and supervisors cannot obtain the relevant information of the two parties of the communication from the network layers.

(1) Security. In terms of security, this article considers several common attacks: Sybil attacks, man-in-the-middle attacks, and DoS attacks. A Sybil attack refers to the fact that a few nodes in a P2P network control the majority of nodes and obtain multiple false identities, making it no longer a peer-to-peer network. In this system, since the console server is credible, the scenario of the Sybil attack is that the node is controlled by the attacker and all synchronized files are obtained by the attacker. In fact, what distinguishes this system from other P2P networks is that the console server is a trusted central control node that can control and monitor the abnormal traffic of all nodes and notify the node user when there is an abnormality. Abnormal nodes are quickly separated from the network, ending the witch attack. A man-in-the-middle attack means that the information of the communicating parties is intercepted and forwarded by the attacker. However, this system not only uses TLS1.3 to encrypt the traffic at the network layer but also uses digital signatures and encryption for valid information at the application layer; therefore, only the receiver can successfully decrypt it and avoid man-in-the-middle attacks. According to Abhishta [38], the possible DOS attacks in this system occur during the communication process when a node is maliciously controlled, and before the console server takes it offline, a large amount of malicious data is sent to other nodes, which causes the network bandwidth to be occupied and other normal forwarding services cannot be performed. In the information transmission of 3.1, this paper has proposed the fact that the system will send a maximum time limit during precommunication. Therefore, when the sender node in the network does not receive the flag information returned by the node within the maximum time limit, the console server sends data to ensure that all nodes discard the malicious data, thereby preventing DOS attacks. Besides that, the cost and effectiveness must be two targets for each communication network. Naiwei Liu [39] came up with a method for trustzone, in which way we could get the same way to find out the cost and effectiveness of our node.

(2) Antitraceability. Since this article implements routing in the form of file forwarding using software in an uncontrolled network environment, it can better resist traditional network-level traceability attacks, including passive traceability and active traceability. In passive traceability, all types of attacks come from correlation attacks. A correlation attack occurs when the attacker can control the nodes of the anonymous channel and can observe the ingress and egress traffic of the anonymous channel at the same time. Next, they can compare the traffic packets and their sequence within a certain time delay and then analyze the corresponding information to achieve a traceability effect. Correlation attacks require that both ends of the communication be under control, but, in large-scale network confrontations, the network where the sender and receiver nodes are located is within the supervision of the supervisor, and the traffic at the network layer is monitored and correlated by the supervisor. In our system, the sender and the receiver only exist once in the point-to-point communication at the application layer, the amount of payload data that they have is small, and most of the data are encrypted and transmitted through other nodes. Therefore, within a certain time delay, the supervisor cannot associate the traffic of the two communicating parties from the massive traffic, which guarantees the noncorrelation. Active traceability is mainly based on network watermarking attacks. A network watermarking attack means that when the traffic enters the anonymous channel, the network supervisor inserts specific watermark information into the traffic, and when the traffic is received by the receiver, the two are correlated, thereby destroying the anonymity. According to the watermark form, it can be divided into four forms based on content, delay, packet length, and ratio. The common point of this type of attack method is that the object is a network stream. Therefore, the produced watermark is inevitably lost in multiple asynchronous forwarding of multiple nodes in different physical environments. The supervisor cannot obtain the relationship between the node and the console server, which guarantees the noncorrelation of the system. In addition, due to the different paths used to forward valid data each time, the supervisor cannot distinguish the real recipient, thus ensuring anonymity.

5. The Modeling Method

The goal is to quantify the invisibility of an anonymous communication network, which consists of the invisibility of client access and the invisibility of traffic in the network. Therefore, this paper indicates the need to detect both client programs and traffic covertness. The covertness detection model can be represented by the covertness block diagram, which is similar to the malware detection model [40]. This block diagram is a logical graphical description method that determines the probability of covertness behavior by probability analysis of each available data and obtains a relative covertness score based on this probability. Therefore, as long as the general characteristic data collection is stipulated, the covertness of the client access stage and data transmission stage of any anonymous communication network can be evaluated.

In the general model, the detection program collects all commonly available data. Based on the threat modeling in this paper, the data collected by the detection program will not be tampered with by attackers.

5.1. Covertness Block Diagram. The covertness block diagram is defined as the following path from left (start state) to right (end state). Each node in the path corresponds to a condition, according to which the probability can be determined as PI. Thus, the probability of each node I on the entry path is as follows:

$$P = \min\left(\left(1 + p_{i-1}\right) * P_i, 1\right). \tag{2}$$

The order in the direction is determined by the order from small to large according to the judgment probability of the collected data.

5.2. Universal Covertness Detection Model

5.2.1. Access Covertness. In order to observe users accessing anonymous communication networks, observers first need to observe egress traffic data. As seen from the relevant work in Section 2, the mainstream domain fronting technology now mainly uses large Internet cloud service providers, such as Microsoft, Amazon, and Cloudflare. Therefore, the domain names, DNS query records, and IP addresses accessed by the outbound traffic have become critical observability indicators. Second, to further check whether the host where the client resides has covert access behavior, the detection program periodically samples and scans the node within a certain time T after detecting the preegress traffic to see if there is traffic with the same destination. If there is traffic, $\mathrm{P_i}$ will prove that the node has no covert access.

5.2.2. Covertness of Transmission. In the transmission stage, the traffic and the performance status of each node before and after user data enter the anonymous communication network should be considered simultaneously. Considering the specific parameters involved in the access and data transmission of an anonymous communication network, the following table is given in this paper Table 1.

Based on the above methods, the model constructed in this paper is displayed in Figure 6.

The corresponding equation is as follows:

$$\begin{aligned} C = &-\log\left(P\text{process} * P\text{ports}\right) \\ &- \log\left[P_{I/o} * \left(PN1 * PN2 * PN3 + P\ D\right)\right]. \end{aligned} \tag{3}$$

6. Experiment

By building and deploying the system, this chapter shows the results of the basic performance, including the response of the control center, the forwarding delay, and the throughput of the core network. In addition, the results of the covertness of the system and Tor have been measured.

6.1. Response Time of the Control Center. The response of the control center mainly includes the delay of flow table switching, the response time of distribution, and the response of node state acquisition. The data collected in this section are the differences between the reading database record time and the execution operation time of the web system. The test results show that the response time of these activities in the actual test is less than 5 seconds and in most cases not more than 3 seconds, which can be regarded as a real-time response.

6.2. Forwarding Delay. The forwarding delay of the core network is the time required by both sides of communication from sending data to receiving data. The sender will split the original data to be sent into different slices and send them to the receiver, and the receiver will restore the original data. The test results of this section are shown in the following Table 2.

It can be seen from the experimental results that when the number of slices increases, the delay is considerably reduced. This is because the scheme described in this paper transmits file units. When the file size is less than the size of the data transmitted in unit time, the transmission queue can maintain fast parallel transmission. When the file size is too large, each file is transmitted in a single queue. Therefore, when the business scenario is faced with the need to reduce delay, the requirements can be customized according to packet fragmentation.

6.3. Forwarding Throughput of the Core Network. The forwarding throughput of the core network is the data forwarding volume of each node in the process from sending data to receiving data. The sender will split the original data to be sent into a fixed number of pieces and send it to the receiver through multiple links, and the receiver will restore the original data. The test results of this section are shown in the following Table 3.

It can be seen from the experiment that when the number of links increases, the load of each link is relatively balanced with that of each receiver. Therefore, it is not

Table 1: Various parameters.

Parameters	Probability	Reasons
Throughput	$P_{I/O}$	The throughput can show whether the node exchanges data with unknown computers
DNS	P_{DNS}	DNS can figure whether some in common domain name is used or the domain fronting is used
DstIP	P_{DstIP}	The IP is in the blacklist or not
Process	$P_{Process}$	Some software such as Tor client can be shown by the cloud provider or supervisor
Port	P_{Port}	Corresponding port can be some software's fingerprint such as syncthing and its 8384&22000

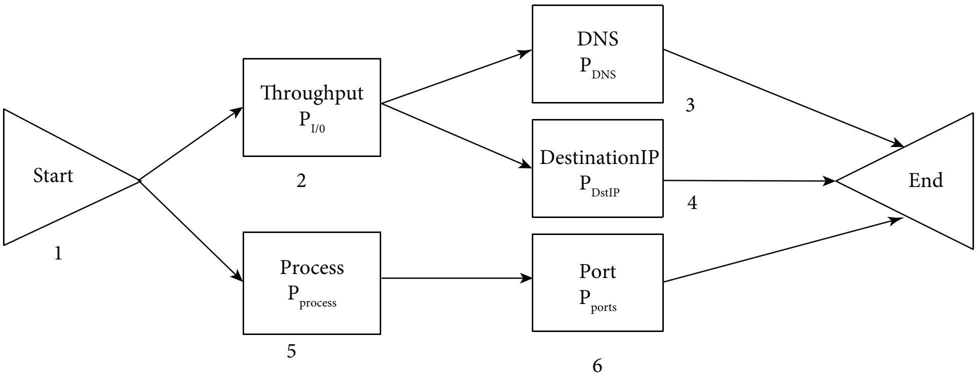

Figure 6: Block diagram.

Table 2: Test results of forwarding delay of core network.

Data sizes	Slices	Delay(ms)
1 M, 10 M, 100 M	1	10, 6240, 15000
1 M, 10 M, 100 M	2	10, 6387, 8093
1 M, 10 M, 100 M	4	10, 2000, 6025
1 M, 10 M, 100 M	8	10, 179, 4383

Table 3: Test results of forwarding the throughput of the core network.

Data size (MB)	Slices	Routes	Sender throughput	Routes throughput	Receiver throughput
10	8	1	13.75	27.5	13.75
10	8	2	13.75	12.5 : 15	13.75
10	8	3	13.75	5 : 12.5 : 10	13.75
10	8	4	13.75	5 : 10 : 7.5 : 5	13.75

difficult to determine that high concurrency can be achieved through multiple links. When the business scenario is faced with the need to improve transmission efficiency, the demand can be customized by increasing the number of routes.

6.4. Security Lower Bound Assessment. Because the anonymous communication network described in this paper adopts the idea of software definition, the requirements can be customized according to different scenarios. Therefore, this section attempts to obtain different levels of covertness scores by adjusting the number of nodes and carries out experimental tests on them. The specific method is that, in the system described in this paper, the file with a data size of 10 MB is forwarded from the overseas node to the domestic node. On the basis of ensuring that each data exchange in the link in the transmission stage occurs in nodes in two different countries, the covertness score of a set of anonymous communication networks can be calculated by manually adjusting the number of nodes that passed by the client traffic. We calculate the covertness score according to the above covertness test method (Figure 7).

It is easy to know that the data transmission delay increases linearly with the increase in the number of relay nodes. From the global perspective, the more nodes there are on the link, the greater the probability that forwarding behavior has the same characteristics.

Therefore, when the number of nodes is greater than 5, the covertness score does not increase substantially. In the current scenario, when using the anonymous communication network scheme in this paper, the security lower bound of the relay node is 5.

6.5. Covertness Comparison. The anonymous communication network described in this paper transmits data through files, while Tor and other anonymous communication systems

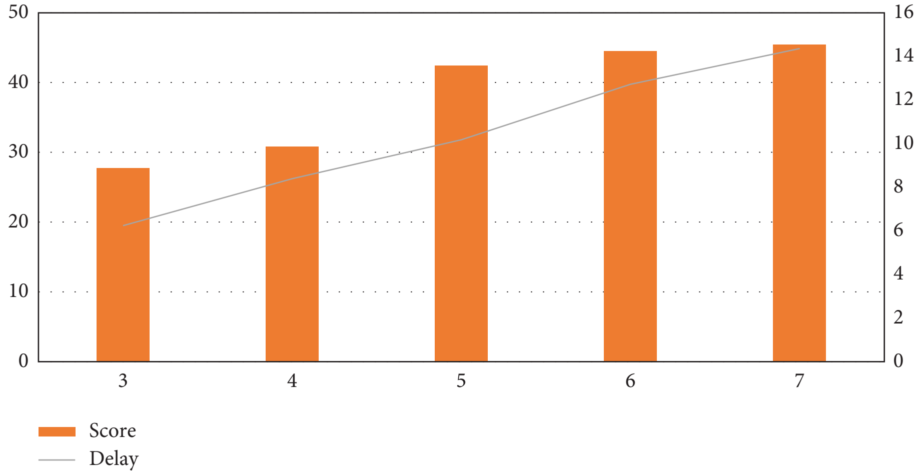

Figure 7: The covertness score.

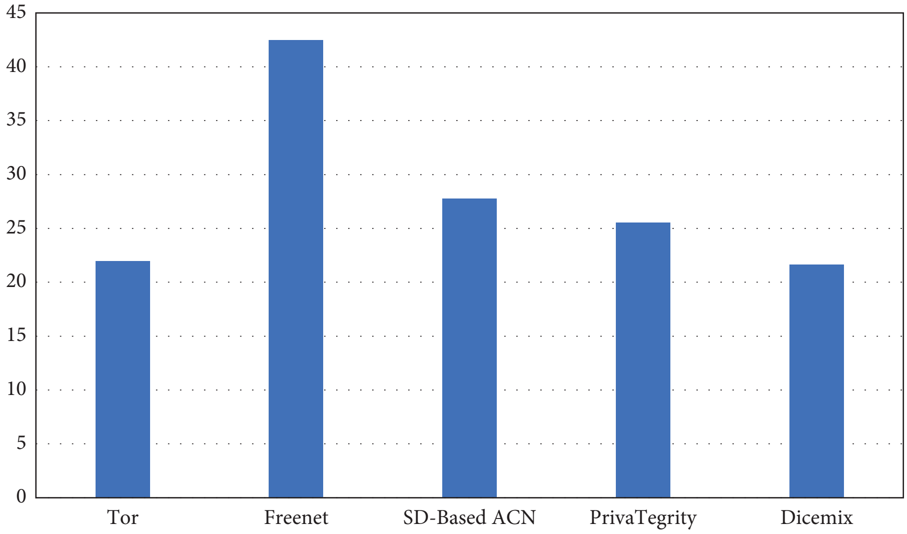

Figure 8: The covertness score.

transmit data in the form of streams. Considering that the link selection of Tor cannot be set manually and locally, this paper simulates the Tor network through shadow on the server during covertness comparison and runs the anonymous communication network client described in this paper on a centos7 virtual machine according to Tor's simulation communication log, restoring the communication relationship of the simulation log. Then, we collect the data of the two communication processes and measure the covertness. In order to add the systems to be compared, this paper selected several systems that can be simulated in the Intranet. As an anonymous file sharing system with high latency, freenet can deployed through docker. Besides that, both PrivaTegrity and Dicemix, which is based on cMix [41], could be built and evaluated.

In this experiment, we can adjust the probability of detecting characteristic traffic in the outlet traffic by adjusting the number of redundant segments (1/2/4/8) when the client program sends data. At the same time, a curl is used to send requests to the domain names of major Internet cloud service providers according to different ratios to set the probability of indicators in another I/O. The data graph obtained at the end of this experiment is as follows.

When using the anonymous communication network described in this paper, we send the same picture to another server located abroad through the designated client.

We run Wireshark on the host of the centos7 virtual machine to capture the virtual machine program and the network card, which is used to simulate the cloud server operator and the defender to detect its export traffic and service status. We run tcpdump on the controlled node to simulate the supervisor to supervise each node.

Finally, by comparing and transmitting the communication behavior in the simulation log many times, the covertness score comparison between the anonymous communication network described in this paper and Tor is obtained, as shown in Figure 8 as follows.

It can be seen from the figure that the covertness evaluation method can evaluate the covertness of onion routed and mix-based anonymous communication network. In addition, in some specific network scenarios, software-defined anonymous communication networks can obtain higher covertness points than Tor. Thanks to the high latency, freenet could get the highest covertness points.

7. Conclusions and Discussions

This paper focuses on the construction of an anti-eavesdropping anonymous communication network system under the condition of the uncontrolled existence of various traffic characteristic detection environments and analyzes the problems existing in all stages of anonymous communication networks from access to data transmission. This paper proposes an anonymous communication network system based on the idea of software definition. Due to the current situation that there is no good quantitative evaluation method to solve these problems, this paper proposes a method to measure invisibility and uses this method to compare the proposed system with the onion network to prove the effectiveness of the system on invisibility.

What is more, there are still many details to be improved. Besides information in the transport layer or application layer, an optimized firewall anomaly resolution improved by Fulvio Valenza [42] can make specific rules changed more quickly. And the total of our channels can be increased. For example, Sherifdeen Lawa [43] introduced microfrontend and it could be used to deploy the microservice faster and more flexible [14].

Data Availability

All relevant data used to support the findings of the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study

Acknowledgments

This work was supported by National Key Research and Development Plan (Grant No.2018YFB1800702), National Natural Science Foundation of China (Grant no. 61976064), Natural Science Foundation of Hunan Province No.2020JJ6064, the Major Key Project of PCL (Grant nos. PCL2022AS21, PCL2021A02, and PCL2021A09), Guangzhou Science and Technology Plan Project (202102010471), and Guangdong Province Science and Technology Planning Project (2020A1414010370).

References

[1] F. Rochet, R. Wails, A. Johnson, P. Mittal, and O. Pereira, “CLAPS: client-location-aware path selection in tor,” in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 17–34, IEEE, Virtual Event, USA, November 2020.

[2] I2Porg, *I2P - Invisible Internet Project*, I2Porg, 2010.

[3] C. Egger, J. Schlumberger, C. Kruegel, and G. Vigna, “Practical attacks against the I2P network,” *Lecture Notes in Computer Science*, vol. 8145, pp. 432–451, 2013.

[4] Q. Wang, X. Gong, G. T. K. Nguyen, A. Houmansadr, and N. Borisov, “CensorSpoofer,” in *Proceedings of the 2012 ACM conference on Computer and communications security - CCS ’12*, pp. 121–132, IEEE, New York, NY, USA, October 2012.

[5] D. Rebollo-Monedero, J. Parra-Arnau, C. Diaz, and J. Forné, “On the measurement of privacy as an attacker’s estimation error,” *International Journal of Information Security*, vol. 12, no. 2, pp. 129–149, 2012.

[6] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, “Anonymous and traceable group data sharing in cloud computing,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[7] E. J. Infeld, “Symmetric disclosure: a fresh look at k-anonymity,” in *Proceedings of the 4th USENIX Work. Free Open Commun. Internet, FOCI 2014, co-located with USENIX Secur. 2014*, San diego, CA, USA, August 2014.

[8] P. Venkitasubramaniam and A. Mishra, “Anonymity of memory-limited chaum mixes under timing analysis: an information theoretic perspective,” *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 996–1009, 2015.

[9] D. Chaum, F. Javani, A. Krasnova, A. Kate, J. de Ruiter, and A. T. Sherman, “cMix: Anonymization by High-Performance Scalable Mixing,” TechReport, eprint, Kearney, MI, USA, 2016.

[10] M. Backes, A. Kate, S. Meiser, and E. Mohammadi, “(Nothing else) MATor(s): monitoring the anonymity of Tor’s path selection,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 513–524, IEEE, New York, NY, USA, November 2014.

[11] Y. He, L. Hu, and R. Gao, “Detection of Tor Traffic Hiding under Obfs4 Protocol Based on Two-Level Filtering,” in *Proceedings of the 2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, June 2019.

[12] S. R. Sheffey and F. Aderholdt, “Improving MEEK with Adversarial Techniques,” in *Proceedings of the FOCI @ USENIX Security Symposium*, Santa Clara, CA, USA, August 2019.

[13] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, “Blocking-resistant communication through domain fronting,” *Proceedings on Privacy Enhancing Technologies*, vol. 2, pp. 46–64, 2015.

[14] G. Cybenko and S. Huntsman, “Analytics for directed contact networks,” *Appl. Netw. Sci.*vol. 4, no. 1, p. 106, 2019.

[15] M. K. Reiter, “Crowds:Amonymous for Web Transactions,” *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, 1998.

[16] J. Qiao and Y. Li, “Resource leveling using normalized entropy and relative entropy,” *Automation in Construction*, vol. 87, pp. 263–272, 2018.

[17] R. Archibald and D. Ghosal, “A comparative analysis of detection metrics for covert timing channels,” *Computers & Security*, vol. 45, pp. 284–292, 2014.

[18] K. Chatzikokolakis, C. Palamidessi, and P. Panangaden, “Probability of error in information-hiding protocols,” in *Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF’07)*, pp. 341–351, IEEE, Venice, Italy, July 2007.

[19] P. Venkitasubramaniam and L. Tong, "A game-theoretic approach to anonymous networking," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 892–905, 2012.

[20] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi, "ANOA: a framework for analyzing anonymous communication protocols," in *Proceedings of the 2013 IEEE 26th Computer Security Foundations Symposium*, pp. 163–178, IEEE, New Orleans, LA, USA, June 2013.

[21] D. Das, S. Meiser, E. Mohammadi, and A. Kate, "Anonymity trilemma: strong anonymity, low bandwidth overhead, low latency - choose two," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, pp. 108–126, San Francisco, CA, USA, May 2018.

[22] G. Muradova and M. Hematyar, "Securing and hiding the destination of confidential medical information with domain fronting," in *Proceedings of the 2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT)*, October 2019.

[23] K. Shahbar and A. N. Zincir-Heywood, "How Far Can We Push Flow Analysis to Identify Encrypted Anonymity Network Traffic?" in *Proceedings of the NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018.

[24] S. Qureshi, D. Gordhan, S. Tunio, F. Ullah, A. Nazir, and A. Wajahat, "Performance analysis of open source solution 'ntop' for active and passive packet analysis relating to application and transport layer," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, 2019.

[25] J. Zhao, X. Jing, Z. Yan, and W. Pedrycz, "Network traffic classification for data fusion: a survey," *Information Fusion*, vol. 72, pp. 22–47, 2021.

[26] D. Fifield, "Turbo Tunnel, a Good Way to Design Censorship Circumvention Protocols," in *Proceedings of the 10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*, USENIX Association, Berkeley, CA, USA, August 2020.

[27] Z. Wang, J. Zhang, Q. Liu, X. Cui, and J. Su, "Practical Metrics for Evaluating Anonymous Networks," *Science of Cyber Security*, pp. 3–18, 2018.

[28] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis, "The Loopix anonymity system," in *Proceedings of the 26th USENIX Secur. Symp*, pp. 1199–1216, Vancouver, BC, USA, May 2017.

[29] K. Shahbar and A. N. Zincir-Heywood, "An Analysis of Tor Pluggable Transports under Adversarial Conditions," in *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, November 2018.

[30] Y. Li, "LAP-Net: Level-Aware Progressive Network for Image Dehazing," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[31] C. Chen and A. Perrig, "PHI: path-hidden lightweight Anonymity protocol at network layer," *Proceedings on Privacy Enhancing Technologies*, vol. 1, pp. 100–117, 2017.

[32] R. Annessi and M. Schmiedecker, "NavigaTor: finding faster paths to anonymity," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 214–226, IEEE, Saarbruecken, Germany, March 2016.

[33] D. Fifield and L. Tsai, "Censors' Delay in Blocking Circumvention Proxies," *Delay in Blocking Circumvention Proxies*, 2016.

[34] P. Kotzanikolaou, G. Chatzisofroniou, and M. Burmester, "Broadcast anonymous routing (BAR): scalable real-time anonymous communication," *International Journal of Information Security*, vol. 16, no. 3, pp. 313–326, 2016.

[35] G. Wan, A. Johnson, R. Wails, S. Wagh, and P. Mittal, "Guard placement attacks on path selection algorithms for tor," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 272–291, 2019.

[36] A. Kitana, I. Traore, and I. Woungang, "Towards an epidemic SMS-based cellular botnet," *J. Internet Serv. Inf. Secur.*vol. 10, no. 4, pp. 38–58, 2020.

[37] D. H. Duong, W. Susilo, and V. C. Trinh, "Wildcarded identity-based encryption with constant-size ciphertext []and secret key[J]. J. Wirel. Mob. Networks ubiquitous comput," *Dependable Appl.*vol. 11, no. 2, pp. 74–86, 2020.

[38] A. Abhishta, W. van Heeswijk, and M. Junger, "Why would we get attacked? An analysis of attacker's aims behind DDoS attacks," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*vol. 11, no. 2, pp. 3–22, 2020.

[39] N. Liu, M. Yu, and W. Zang, "Cost and effectiveness of TrustZone defense and side-channel attack on ARM platform," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*vol. 11, no. 4, pp. 1–15, 2020.

[40] G. Cybenko, G. Stocco, and P. Sweeney, "Quantifying covertness in deceptive cyber operations," in *Cyber Deception: Building the Scientific Foundation*Springer, Cham, Switzerland, 2016.

[41] W. K. Chan, J. J. Chin, and V. T. Goh, "Simple and scalable blockchain with privacy," *Journal of Information Security and Applications*, vol. 58, Article ID 102700, 2021.

[42] F. Valenza and M. Cheminod, "An optimized firewall anomaly resolution," *J. Internet Serv. Inf. Secur.*vol. 10, no. 1, pp. 22–37, 2020.

[43] A. Pavlenko, N. Askarbekuly, and S. Megha, "Micro-frontends: application of microservices to web front-ends," *J. Internet Serv. Inf. Secur.*vol. 10, no. 2, pp. 49–66, 2020.