Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.026939

ARTICLE

RAISE: A Resilient Anonymous Information Sharing Environment

Ning Hu[1], Ling Liu[1], Xin Liu[3], Kaijun Wu[2] and Yue Zhao[2,*]

[1]Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China

[2]Science and Technology on Communication Security Laboratory, The 30th Research Institute of China Electronics Technology Group Corporation, Chengdu, 610041, China

[3]College of Computer Engineering and Applied Math, Changsha University, Changsha, 410022, China

*Corresponding Author: Yue Zhao. Email: yuezhao@foxmail.com

Received: 04 October 2022 Accepted: 09 March 2023 Published: 03 August 2023

ABSTRACT

With the widespread application of cloud computing and network virtualization technologies, more and more enterprise applications are directly deployed in the cloud. However, the traditional TCP/IP network transmission model does not fully consider the information security issues caused by the uncontrollable internet environment. Network security communication solutions represented by encrypted virtual private networks (VPN) are facing multiple security threats. In fact, during the communication process, the user application needs to protect not only the content of the communication but also the behavior of the communication, such as the communication relationship, the communication protocol, and so on. Inspired by blockchain and software-defined networking technology, this paper proposes a resilient anonymous information sharing environment, RAISE. The RAISE system consists of user agents, a core switching network and a control cluster based on a consortium blockchain. User agents are responsible for segmenting, encrypting, and encapsulating user traffic. The core switching network forwards user traffic according to the rules issued by the controller, and the controller dynamically calculates the forwarding rules according to the security policy. Different from onion routing technology, RAISE adopts the controller to replace the onion routing model, which effectively overcomes the uncontrollability of nodes. The dispersed computing model is introduced to replace the TCP/IP pipeline transmission models, which overcomes the problems of anti-tracking and traffic hijacking that cannot be solved by VPNs. We propose a blockchain control plane framework, design the desired consensus algorithm and deploy a RAISE system consisting of 150 nodes in an internet environment. The experimental results show that the use of blockchain technology can effectively improve the reliability and security of the control plane. While maintaining high-performance network transmission, it further provides network communication security.

KEYWORDS

Software-defined anonymous communication network; blockchain; network communication security

1 Introduction

As increasingly more data involving government decisions, business secrets and personal privacy are transmitted over the internet, the internet has evolved from an early academic research

communication platform to an infrastructure that carries government and commercial application communications. In recent years, the economic losses caused by internet security communication problems have increased year by year, and internet security communication has become the focus of widespread attention [1–3]. On the other hand, with the improvement of cloud computing technology and services, building enterprise applications based on the internet has become a development trend. Therefore, how to provide programmable and controllable secure transmission services for cloud-based distributed enterprise applications has become an urgent question.

At present, internet applications mainly rely on encrypted virtual private network (VPN) technology to ensure the security of network transmissions, such as IPSec VPN and SSL [4,5]. Unfortunately, with the rapid development of global computing power and artificial intelligence technology, secure transmission solutions based solely on encryption technology are facing multiple threats. The first is traffic monitoring risk. The exposure of the PRISM event and the Quantum platform showed that network traffic monitoring is ubiquitous. With the continuous growth of global computing power, encryption algorithms are facing the possibility of being cracked. For example, algorithms such as DES, 3DES, and MD5 have been shown to be insecure. In addition, the Heartbleed bug of OpenSSL suggests that the implementation of encryption-based secure communication solutions may be vulnerable to information disclosure. The second threat is traceability risk. Since the transmission model of the TCP/IP network performs routing and forwarding based on IP addresses, it is difficult to prevent attackers from tracing traffic to the source. Finally, there is the risk of traffic hijacking. By publishing fake routes, attackers can cause user traffic to be forwarded to the wrong destination, disrupting the communication process.

To compensate for the lack of encrypted VPN technology, anonymous communication networks, such as Tor and I2P, have been proposed. Anonymous communication networks cannot only protect the confidentiality of communication content but also prevent third parties from tracking and tracing network traffic. Currently, the anonymous communication network is composed of a large number of internet volunteer nodes, which prevent third parties from tracing the source by using onion routing and hybrid network technology. However, due to the lack of controllability and programmability, existing anonymous communication networks are not suitable as infrastructure for carrying enterprise internet applications. Anonymous communication networks mainly adopt onion routing to prevent traceability, but this routing method cannot implement a more flexible routing strategy, such as actively avoiding areas where there is a threat of monitoring. In addition, anonymous communication networks are composed of volunteer nodes that users cannot control and cannot provide stable and reliable online time, transmission bandwidth, and trusted services. Attackers can launch sybil attacks and eclipse attacks by deploying a large number of volunteer nodes.

Software defined network (SDN) is a new type of network architecture. By decoupling the control plane and data plane, users can define network forwarding behavior at the application layer, making the network system highly adaptable to user needs. Compared with onion routing technology, SDN has better programmability and controllability. The security of SDN controllers has been a research hotspot in recent years. Blockchain, as a decentralized trusted data storage technology, is widely used to build a trusted infrastructure. The use of blockchain technology in the construction of SDN networks can overcome the security problems brought about by the centralized control model of the SDN control plane [6–11].

Inspired by the above research work, this paper proposes RAISE, a resilient anonymous information sharing environment, which improves the programmability and controllability of anonymous

communication networks by changing the architecture and transmission model of anonymous communication networks. Our main contributions are as follows:

(1) We propose a user-programmable anonymous communication network that uses an independent control plane instead of onion routing, overcomes the singularity of the onion routing model, and supports routing programming based on multiple constraints.

(2) We design a blockchain-based control plane for anonymous communication networks, using blockchain technology to achieve data synchronization and trusted computing across multiple controllers.

(3) We propose a decentralized computing-based transmission model that takes full advantage of the programmable advantages of anonymous communication networks.

The remainder of this paper is organized as follows: Section 2 introduces the motivation; Section 3 proposes the anonymous communication network system. Section 4 presents the comparative experiment; Section 5 reviews related work; and finally, we summarize and look forwards to the work described in the article.

2 Motivation

Existing encrypted VPN services for enterprise applications generally face threats such as content monitoring, source tracking, and traffic hijacking. Consider the scenario shown in Fig. 1. When users transmit data in an internet environment via encrypted VPN, the traffic will be monitored by ISP and attackers. With the rapid increase in global computing power, it is possible to use supercomputers to decrypt encrypted traffic. In addition, there may be bugs in the code implementation of encrypted VPNs, and attackers may exploit these bugs to decipher the content of user communications. With the continuous improvement of the commercial value of the internet, by analyzing a user's network behavior, it is possible to obtain the user's business secrets. Therefore, how to prevent tracking and traceability has also become an important factor in user information security. The pipeline transmission model of TCP/IP cannot effectively resist traceability attacks.

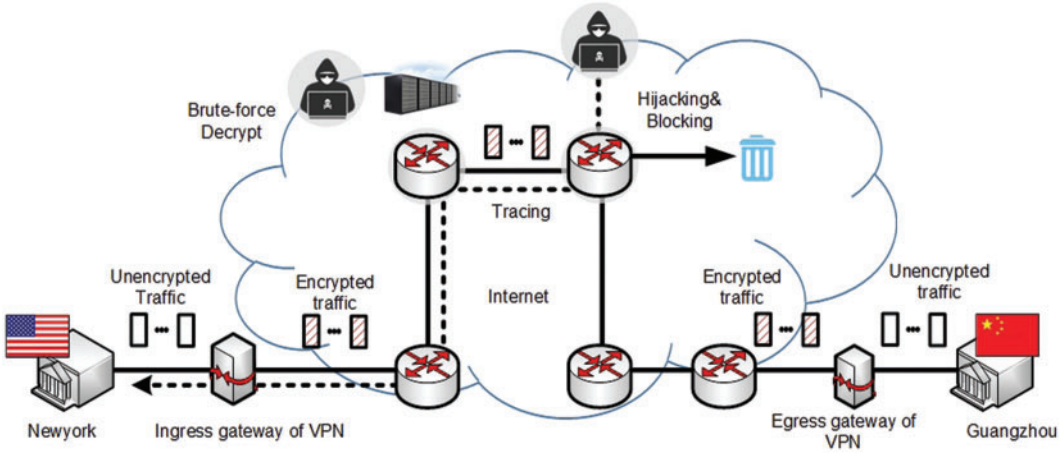

Figure 1: Security threats faced by internet applications

With the development of cloud computing and network virtualization technologies, an increasing number of enterprises deploy their business systems in the cloud. Application cloudification puts forwards new requirements for the programmability and controllability of communication networks. Programmability requirements mainly include message encoding, link selection, routing policy, etc. Although an encrypted VPN can provide security protection for data content, it cannot flexibly change the packet encoding method or dynamically change the transmission path, and it is difficult to prevent content monitoring and traffic tracking. Controllability requirements mainly include forwarding behavior, online time, security policies, etc. However, encrypted VPNs are controlled by service providers, users cannot control VPN gateways, and anonymous communication networks represented by Tor are mainly composed of a large number of voluntary nodes, whose reliability and security cannot be guaranteed.

In summary, to better meet the network security communication requirements of cloud-based applications, it is necessary to build a secure communication network that can be configured and deployed independently based on the internet. Users can define the network topology, routing strategy and transmission behavior by themselves.

3 Software-Defined Anonymous Communication Network

3.1 Design Goal

To effectively resist threats such as traffic monitoring, route tracing, and malicious hijacking, we need to change the traditional TCP/IP pipeline transmission model and propose a new network architecture with programmable data encoding, path selection, and forwarding control. Moreover, the communication network should also adopt a decentralized control architecture to ensure high availability of the control plane. To this end, we propose a blockchain-assisted software-defined anonymous communication network, RAISE, as shown in Fig. 2. The network is an overlay network that can be deployed on a set of VPSs in an internet environment. All network nodes are deployed and controlled by the users themselves, and the data encoding method, path selection and forwarding control of network nodes can be programmed through the control plane. The control plane adopts the alliance blockchain method to form a control cluster to prevent Byzantine failures. For these purposes, we present a novel anonymous communication network for enterprise internet applications–RAISE (Resilient Anonymous Information Sharing Environment).

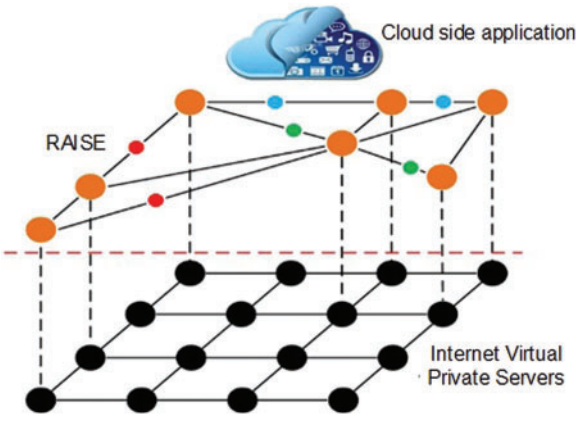

Figure 2: Overlay network based on Internet VPS

3.2 Transmission Model

The data transmission model of the traditional TCP/IP network is a pipeline model. In the process of data transmission, except for packet fragmentation caused by the limit of the link MTU size, network devices usually do not process IP packets. When the link is stable, the IP packets of the same data flow are all forwarded according to a fixed transmission path, and the traffic characteristics remain unchanged during transmission.

To hide user traffic characteristics, RAISE adopts a transmission model based on mix-nets [12], as shown in Fig. 3. The transmission model of RAISE breaks the pipeline model of TCP/IP, mainly including four aspects as follows. First, the mixed network nodes of RAISE can perform data padding, secondary encryption and traffic shaping on traffic while forwarding packets. Since traffic characteristics are constantly changing during transmission, it is difficult for network inspectors to identify traffic. Second, RAISE implements the transport protocol at the application layer and can rely on a variety of common internet applications to carry packets. The network censors cannot analyses the encrypted application layer protocol. Third, unlike the Tor network based on onion routing, RAISE data packets do not carry routing information. When the mix node forwards packets, it calculates the forwarding path according to the flow table issued by the controller. Therefore, the controllability of RAISE is better. Finally, due to the separation of the control plane and the data plane, different packets of the data flow can quickly switch paths during transmission, preventing traffic from being traced.

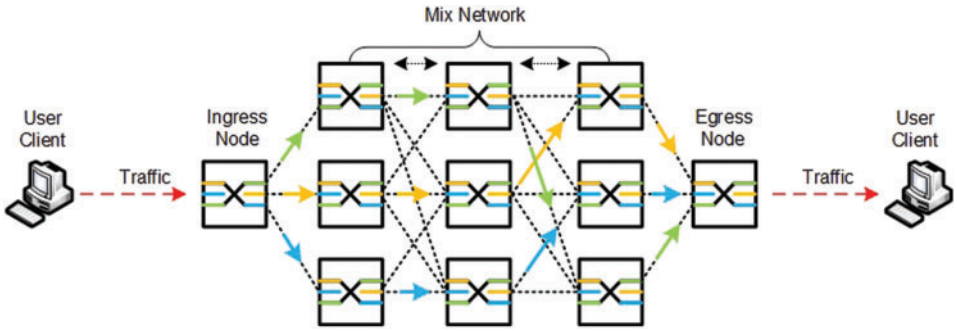

Figure 3: Programmable transmission model

3.3 Routing Model

The control plane is composed of a group of controllers, which are deployed on internet virtual private servers (VPS). The controller is responsible for calculating the transmission paths between the ingress node and egress node according to the network views issued by the manager. RAISE adopts the flow table to record the forwarding path of the packet. The flow table is composed of a group of flow entries. The composition of the flow entry is shown in Fig. 4.

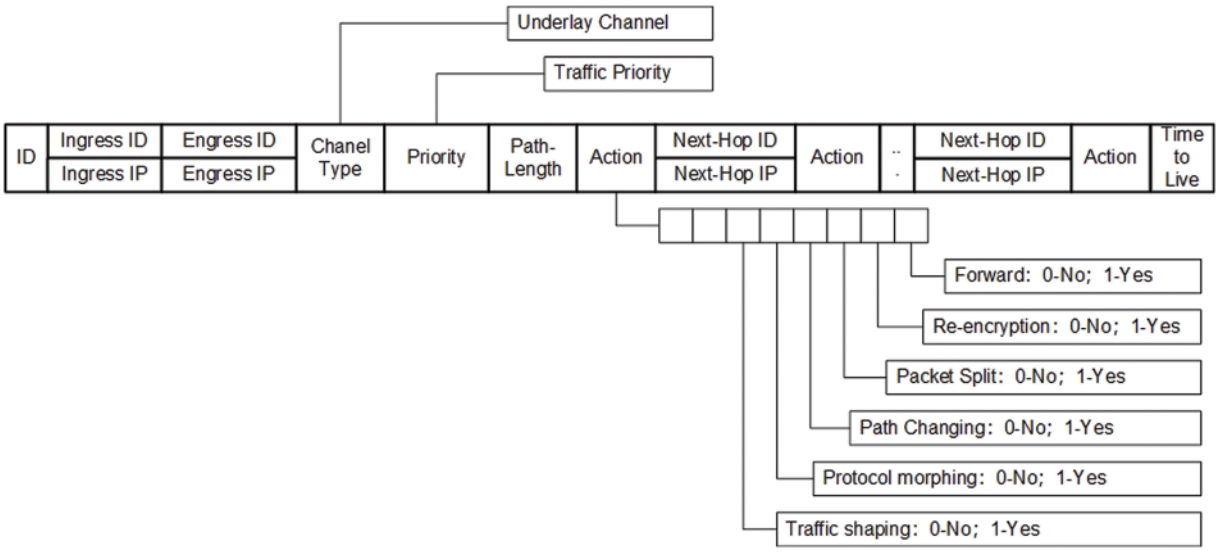

Figure 4: The composition of flow entry

To prevent complete path leakage caused by a compromised mix-net node, the controller further disassembles the flow table into multiple sub-paths, and each sub-path contains only adjacent nodes, which is called a flow table fragment. The mix-net node performs traffic forwarding according to the flow table fragment without perceiving the global network topology and does not care about the complete forwarding path. The routing control model of RAISE is shown in Fig. 5.

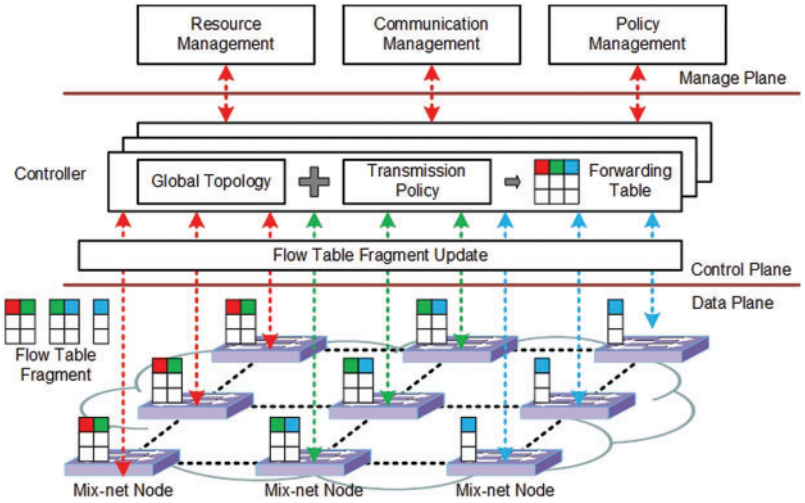

Figure 5: The routing control model of RAISE

Fig. 6 provides an example of the flow table configuration. As shown in Fig. 6, the controller creates two transmission paths between entrance E001 and entrance E002, namely, E001 → F001 → F002 → F006 → E002 and E001 → F004 → F005 → F006 → E002. Therefore, the flow table maintained by the controller contains two complete transmission paths. The controller divides the flow table into different fragments and advertises them to the mix-net nodes. Since each mix-net node can only obtain the IP address of the next-hop transmission node, it cannot obtain the complete

transmission path, and during the process of data forwarding, it can only know the target ID of the data but cannot know the real source ID of the data (the user source ID is encrypted by the agent). Therefore, even if a single transmission node is out of control, the communication relationship between users will not be leaked.

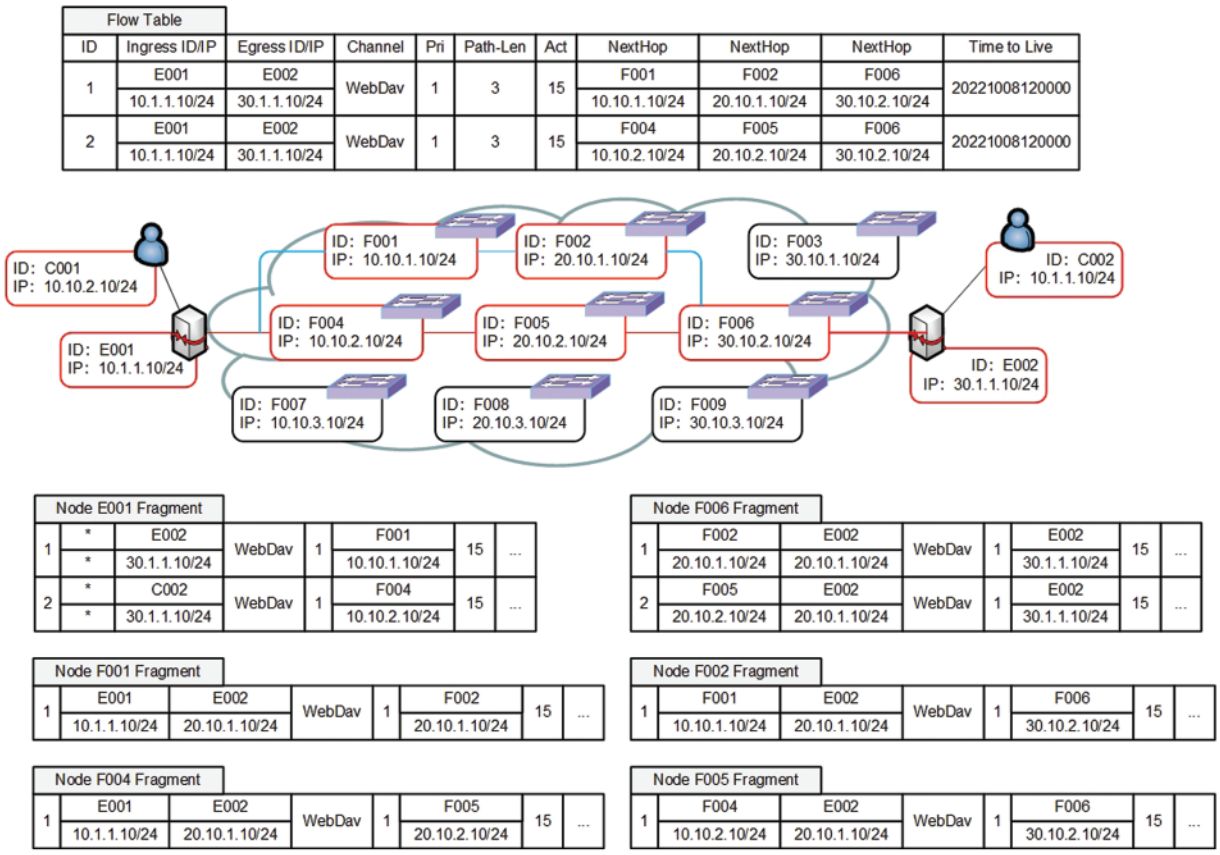

Figure 6: Example of flow table configuration

3.4 Blockchain-Assisted Control Plane

Onion routing technology can hide the communication relationship between users, but this anonymity is based on the randomness assumption of massive volunteer nodes, and the reliability and controllability of volunteer nodes cannot meet the requirements of enterprise applications. In addition, the attacker may construct a Sybil or eclipse attack by deploying a large number of malicious nodes. To reduce the attack surface, RAISE defines the data transmission path through the controllable control plane and prevents a single node from mastering the global path by means of flow table fragmentation. This routing model has better programmability and controllability, but it also puts forwards higher security requirements for the control plane. To implement a more secure controller, we propose a blockchain-based controller. The control plane of RAISE is shown in Fig. 7.

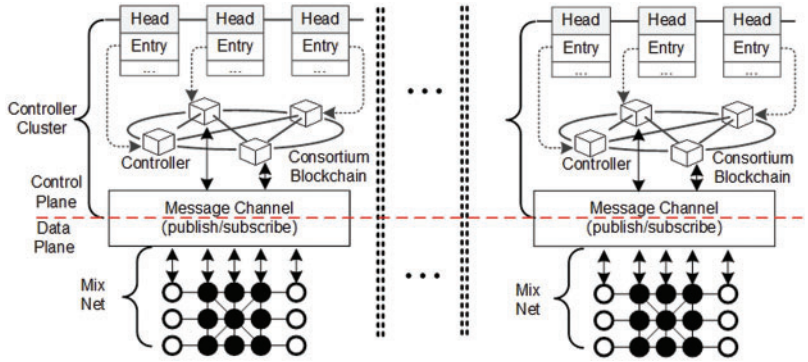

Figure 7: Blockchain-assisted control plane

RAISE organizes multiple controllers into a consortium blockchain and uses the consensus mechanism of the blockchain to realize data synchronization between controllers. In the RAISE system, we adopt the Practical Byzantine Fault Tolerant (PBFT) algorithm as the consensus algorithm between controllers. PBFT can resist Byzantine attacks and can effectively prevent individual compromised controllers from publishing fake flow table information. By optimizing the selection efficiency of nodes, the efficiency of the PBFT algorithm can be further improved. In an anonymous communication network, enterprise applications do not have high requirements for frequent changes in transmission paths. In addition, in another research work of ours, using the PBFT algorithm to synchronize domain name zone data between DNS servers achieved good results [12]. Therefore, RAISE adopts the basic PBFT algorithm. The working process of the algorithm is shown in Fig. 8.

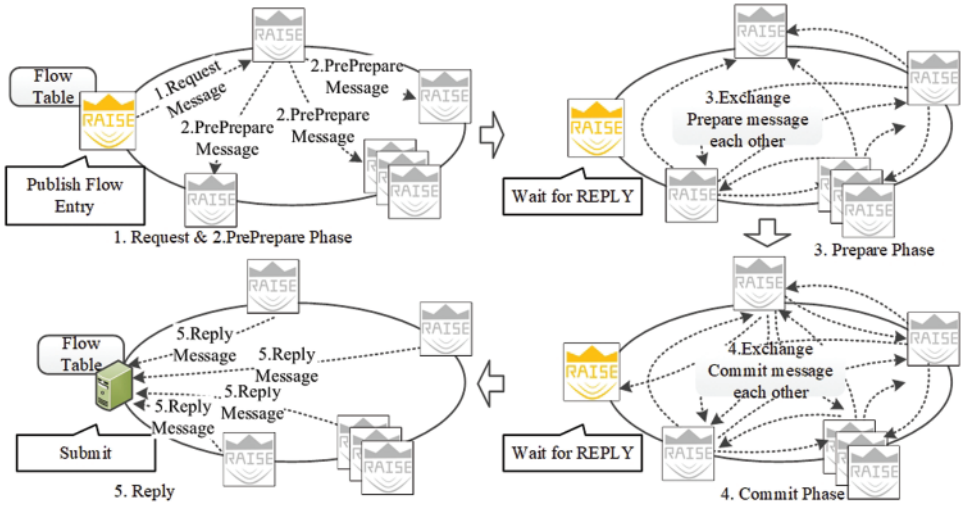

Figure 8: The publication flow table procedure

We have implemented a consensus algorithm based on the basic principles of PBFT in RAISE. The principle code is as Table 1.

Table 1: Consensus algorithm of RAISE controller

Input: REQUEST *message, viewID, timestamp, cert of the node, handle of log file*
Output: *result of operation, log record*

```
1   /* when RAISE controller receives request message from any node, it execute following code,
    the request message has the form as follows:
2
3                          < REQUEST, opeation, timeStamp, client >δc
4   */
5
6   enum status {IS_IDLE, IS_PREPREPARE, IS_PREPARE, IS_COMMIT};
7
8   status = IS_IDLE;
9   while (status == IS_IDLE) do {
10    msg = listen (anyNode);
11    switch msg.type {
12      case REQUEST:
13      if valid_Request_MSG(msg) && isPrimaryNode() && status == IS_IDLE {
14          pp_msg = create_Pre_Prepare_MSG(PRE_PREPARE, view, number, digst)
15          /*<< PREPREPARE, view, number, digst >δp, m >*/
16          multicast_MSGpp_msg
17          status = IS_PREPREPARE
18          log(msg); log(pp_msg);
19      \}/* Initiator Code */
20      break;
21      case PRE_PREPARE:
22        if valid_PrePrepare_MSG(msg) && isBackupNode() && status == IS_IDLE {
23            status = IS_PREPREPARE;
24            if msg_Acceptable(msg) {
25              p_msg = createPrepareMSG(PREPARE, view, number, digst, i);
26              multicastMSG(p_msg);
27              status = IS_PREPREPARE_PHASE
28              log(msg); log(p_msg);
29            }
30        }
31      break;
32      case PREPARE:
33        if valid_Prepare_MSG (msg) && status = IS_PREPARE {
34            log(msg);
35            if prepared(m, v, n, i) {
36              commitMSG = CreateCommitMSG(COMMIT, view, number, digst(m), i);
37              /* << COMMIT, view, number, digst (m) , i >δi, m >*/
38              multicastMSG(commitMSG);
39              status = IS_COMMIT;
```

(Continued)

Table 1 (continued)

Input: REQUEST *message, viewID, timestamp, cert of the node, handle of log file*
Output: *result of operation, log record*

```
                    log(commitMSG);
                }
            }
        break;
        case COMMIT:
            if valid_Commit_MSG(msg) && status = IS_COMMIT {
                log(msg);
                if committed-local(m, v, n) {
                    result = execute(m.operation);
                    replyMSG = createReplyMSG(REPLY, view, timeStamp, client, i, result);
                    /*< REPLY, view, timeStamp, client, i, result >δc */
                    Status = IDLE; /*finished*/
                }
            }
        break;
        default:
            DiscardMSG(msg);
        break;
    } /*end of switch*/
    } /* end of while*/
```

3.5 The Composition of RAISE

As shown in Fig. 9, the composition of the RAISE system includes the transmission agent, underlying transmission channel, hybrid network, controller, manager and a set of application layer protocols.

RAISE defines a cell format at the application layer, which is transmitted as the data payload of common internet application protocols. The cell format and protocol stack of RAISE are shown in Fig. 10.

The transfer agent is the client software deployed on the user side, through which the user application can access the services provided by the RAISE system. Different from IPSec VPN, RAISE implements transmission control at the application layer, relying on common internet services as undelay transmission channels, such as WebDAV, GIT, UDP, etc. Since the application layer control message can be encrypted, the network censor cannot identify the traffic of RAISE through the protocol classification technology. The mix network is an overlay network based on an IP network, consisting of a group of access nodes and mix nodes, both of which are deployed on an internet virtual private server (VPS) as service. RAISE adopts an architecture with a decoupled control plane and data plane. The control plane is composed of a group of distributed controllers, and the data plane is composed of a hybrid network. The controller issues control policies to the mixed network in the form of forwarding flow tables. The manager of RAISE provides user registration, policy management, parameter configuration and other services.

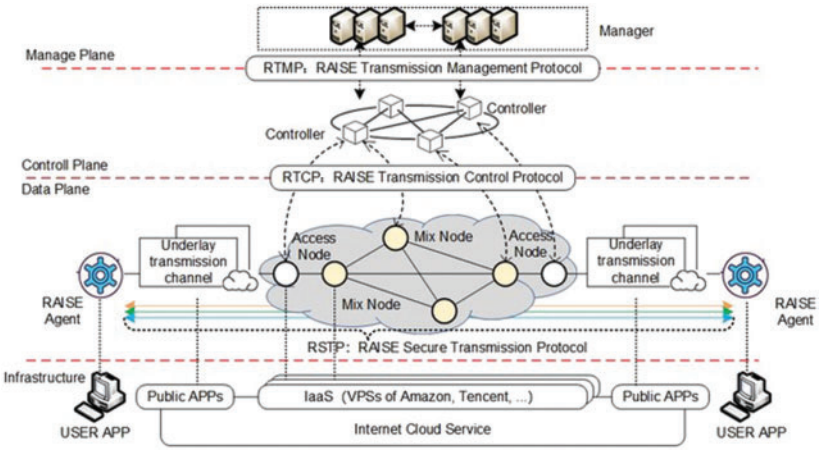

Figure 9: The architecture of RAISE

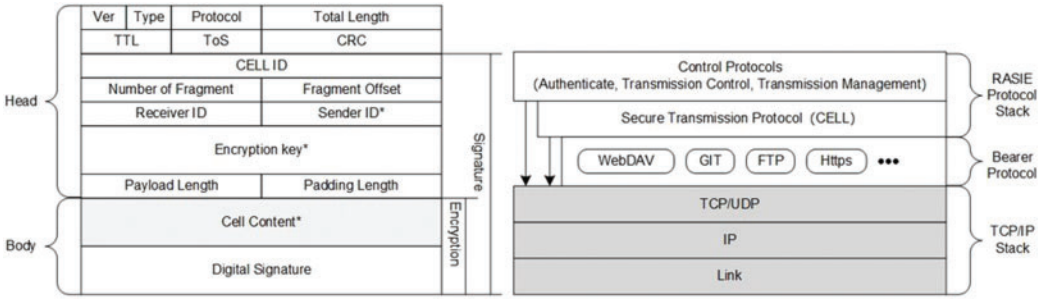

Figure 10: The protocol stack of RAISE

Based on the above design ideas, we implemented the prototype system of RAISE [13]. The mix-net network of the prototype system is based on the VPS deployment of cloud service providers such as Amazon and Google. It contains more than 30 nodes and can carry out covert data transmission based on applications such as Git. Its operation process is shown in Fig. 11.

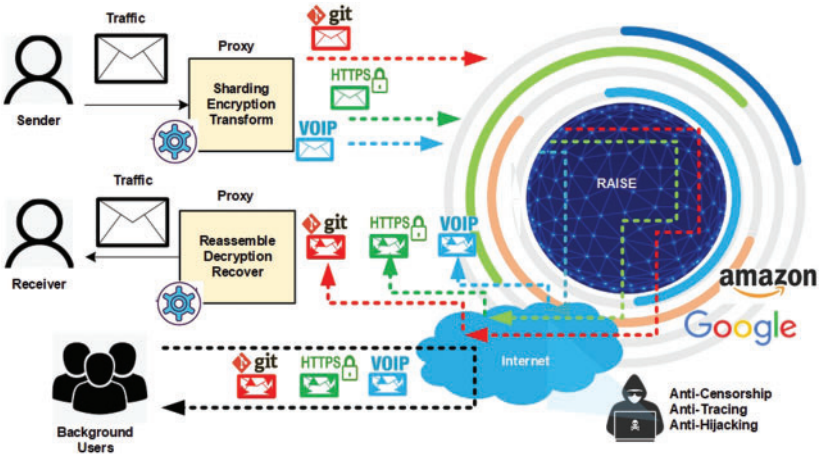

Figure 11: Schematic diagram of the operation process of RAISE

4 Experiment

Since it cannot meet the requirements of enterprise applications in terms of transmission performance and reliability, the anonymous communication network represented by Tor still cannot replace traditional VPNs. It is impossible for enterprises to hand over their business data to networks that cannot be perceived and controlled for transmission.

RAISE combines the ideas of SDN and onion routing to realize an anonymous communication network based on a new architecture, which is expected to become an alternative to traditional VPNs. To evaluate the availability of RAISE, we conducted a comparative test with Tor on the internet. The main test contents include the transmission performance test, transmission channel establishment time, and transmission delay test. In our experimental environment, the mix-net consists of 150 nodes, and these VPS nodes are mainly located in Hong Kong and Beijing, China. Each node is configured with 2 cores, 4G memory, and a network bandwidth of 4 Mbps. We evaluate the differences between RAISE and Tor in terms of system throughput, circuit establishment and transfer latency. Through the test results of these experiments, it is possible to discover the performance improvement brought about by the architectural changes.

4.1 System Throughput

We conducted a week-long file transfer performance test on RAISE and Tor, downloading files through public servers, onion servers and RAISE servers and recording the download time. Referring to the test standard provided by Tor Metric [14], we tested the downloading of 50 KB, 1 MB and 5 MB files. The test results are shown in Figs. 12–14. The vertical axis represents time in seconds, the value is the daily average time, and the horizontal axis represents the test date.

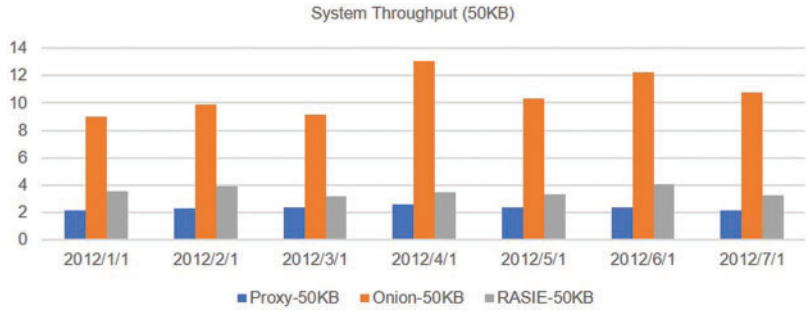

Figure 12: Time consumed to download 50 KB

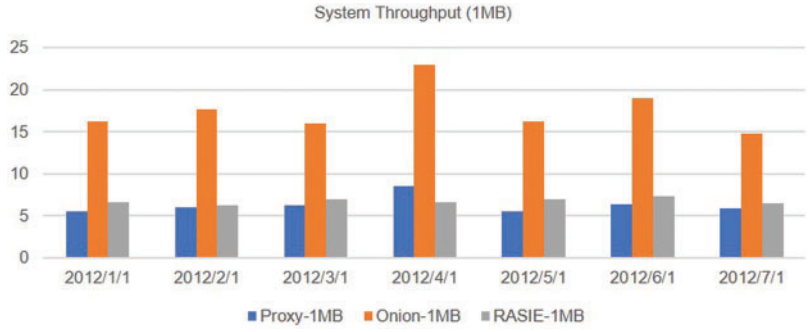

Figure 13: Time consumed to download 1 MB

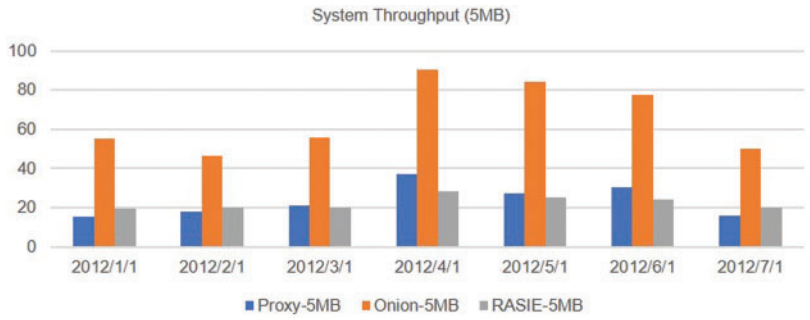

Figure 14: Time consumed to download 5 MB

Although affected by changes in network link quality, test data cannot fully reflect the real system throughput performance. We can still determine that the system throughput of the Tor network is the lowest. This is mainly due to the processing performance of volunteer nodes and the need for layer-by-layer decryption of onion routing. The RAISE system benefits from the performance improvement brought about by decoupling the control plane and the data plane. When the mix-net node forwards data, it does not need to decrypt the transmission path and can fully utilize the processing power. When a large amount of data is transmitted at a time, the system throughput is even better than that of a proxy server and can basically achieve the same transmission performance as a VPN.

4.2 Circuit Build Times

We separately tested the time consumption to establish a complete transmission circuit through the VPN proxy, onion router and RAISE system. Since the transmission path of Tor defaults to 3 hops, when we test the VPN proxy and RAISE, we also pass through 3 different nodes in sequence. We use Onionperf as a test tool when testing the onion router. The test results are shown in Fig. 15.

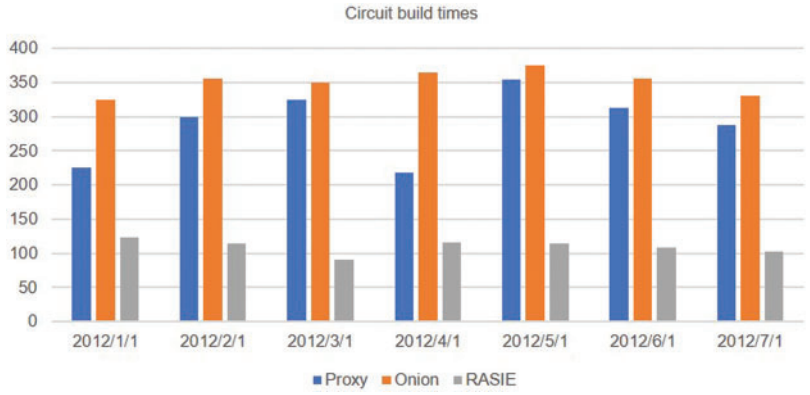

Figure 15: Circuit build times

The vertical axis represents the establishment time of the virtual link in seconds, the value is the average time, and the horizontal axis represents the test date. The experimental results show that RAISE takes the shortest time to establish the transmission circuit because the controller of RAISE can build the complete transmission circuit in advance and send it to the mix-net nodes. The VPN proxy needs to temporarily establish a transmission channel according to the destination address hop

by hop, so circuit establishment takes a long time. The Tor network, because of its complex calculation process of onion routing, leads to the longest link establishment time.

4.3 Circuit Round-Trip Latencies

We also performed a round-trip latency test, that is, the delay between when a byte is sent from the terminal and when the first byte is returned. To be fair, when testing the VPN proxy and RAISE system, the test is also performed through 3 hops. The test results are shown in Fig. 16. The vertical axis represents the round-trip delay in milliseconds, the value is the average value, and the horizontal axis represents the test date.

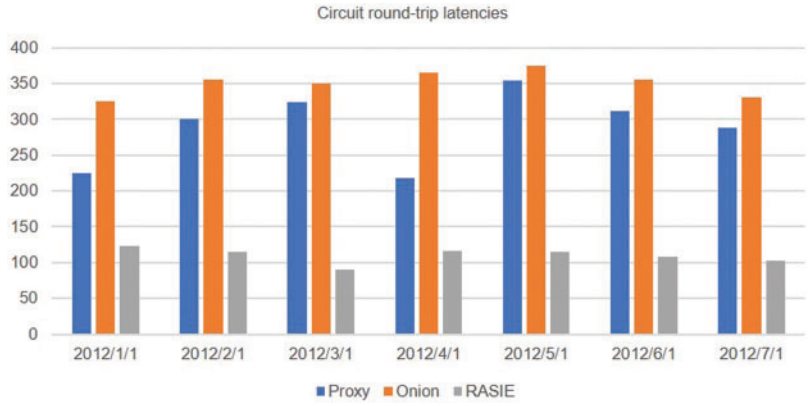

Figure 16: Circuit round-trip latencies

From the experimental results, it can be found that the round-trip latency of the RAISE system is the smallest because RAISE can establish the round-trip path in advance, and the data can be forwarded according to the action policy specified in the flow entry during the forwarding process. However, for the VPN proxy, when it establishes a session relationship for the first time, it also needs to temporarily negotiate a session key, resulting in a large processing delay. Finally, the round-trip path of the onion server may consist of two different unidirectional paths, so the processing overhead is the largest and the delay is the longest.

5 Related Work

To protect the communication privacy of users, the technology of an anonymous communication network is proposed. Current anonymous communication networks are mainly divided into high-latency anonymous communication networks and low-latency communication networks. High-latency anonymous communication networks adopt a connectionless communication model, which can well hide the communication relationship, but it is difficult to effectively support connection-oriented communication services. High-latency communication networks mainly include mix-nets [12], Mixmaster [15], etc. Low-latency communication networks mainly include Tor [16], Pisces [17] and Tarzan [18]. After years of development of anonymous communication network technology, the mechanism of mix-nets and onion routing has been fully verified, and it has a good effect in providing the anonymity of communication relationships. However, the abovementioned anonymous communication networks mainly solve the problem of communication anonymity protection and lack sufficient consideration in terms of controllability and programmability.

With the cloudification of enterprise applications becoming a trend, how to provide secure and reliable network communication for distributed enterprise applications has become a focus. Traditional VPNs have functional flaws in terms of privacy protection. Research on how to use anonymous communication networks to serve enterprise applications is still in its infancy. Based on the architecture of SDN, Wallker et al. [6] proposed a forwarding method for IP packets with encrypted target addresses. Rauf et al. [7] proposed an SDN-based anonymous communication framework for enterprise networks, which solves the problem of communication privacy protection in enterprise networks. Elgzil et al. [19] tried to build a Tor network controller to improve the resource utilization and programmability of anonymous communication networks. Zhu et al. [9] integrated the idea of anonymous communication into the data center SDN network to solve the privacy protection problem of cloud application communication. In general, although SDN technology is used to improve the controllability and flexibility of anonymous communication networks, new anonymous communication networks based entirely on SDN architecture are rare.

Blockchain technology is a distributed accounting technology. Through a specific consensus algorithm, data synchronization between a group of distributed nodes can be achieved without a third party [20,21]. Although blockchain technology was originally proposed because of Bitcoin, in recent years, blockchain technology has been widely used for data synchronization in distributed systems. Since the SDN network was proposed, it has been widely considered because of its efficient routing calculation mode. However, how to ensure the safety of the controller is still an open question [10]. Derhab et al. [22] proposed a blockchain-based SDN control plane, which is composed of multiple controllers. Each controller manages some nodes, forms a blockchain with each other, and uses a consensus algorithm to synchronize control information. Luo et al. [23] proposed a blockchain-based control plane implementation method, which improves the security of the SDN network control plane.

6 Conclusion

Enterprise application cloudification brings new technical challenges to traditional VPN solutions. Users have increasingly urgent demands for the customization of communication networks. Traditional VPN technology is limited by the TCP/IP pipeline model and has insufficient capabilities in anti-tracking and malicious blocking. As an important technology for communication privacy protection, anonymous communication networks can provide a good ability to hide communication relationships. However, because they lack enough programmability and controllability, they still cannot replace VPNs. The RAISE system proposed in this paper is a new anonymous communication network based on SDN architecture. By introducing blockchain technology, the reliability and security of the control plane are improved, and it can provide more secure and credible communication services for enterprise applications. Our follow-up work will focus on how to improve the reliability and performance of the organization.

Acknowledgement: We thank the anonymous reviewers for their constructive comments.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 61976064).

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

References

1. Jangjou, M., Sohrabi, M. K. (2022). A comprehensive survey on security challenges in different network layers in cloud computing. *Archives of Computational Methods in Engineering, 29(6),* 3587–3608. https://doi.org/10.1007/s11831-022-09708-9
2. Vinoth, S., Vemula, H. L., Haralayya, B., Mamgain, P., Hasan, M. F. et al. (2022). Application of cloud computing in banking and e-commerce and related security threats. *Materials Today: Proceedings, 51,* 2172–2175.
3. Li, C., Liang, S. Y., Zhang, J., Wang, Q., Luo, Y. (2022). Blockchain-based data trading in edge-cloud computing environment. *Information Processing & Management, 59(1),* 102786. https://doi.org/10.1016/j.ipm.2021.102786
4. Nam, T. S., Van, T. H., Van, L. N. (2022). A high-throughput hardware implementation of NAT traversal for IPSEC VPN. *International Journal of Communication Networks and Information Security, 14(1),* 43–50. https://doi.org/10.17762/ijcnis.v14i1.5260
5. Liu, Y., Zhu, M., Zhang, Y., Chen, Y. (2022). Software vulnerability prediction based on statistical learning. *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 390–394. Dalian, China.
6. Wallker, P., Santhya, R., Sethumadhavan, M., Amritha, P. P. (2020). Anonymous network based on software defined networking. *4th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 619–624. Tirunelveli.
7. Rauf, B., Abbas, H., Sheri, A. M., Iqbal, W., Bangash, Y. A. et al. (2021). CACS: A context-aware and anonymous communication framework for an enterprise network using SDN. *IEEE Internet of Things Journal, 9(14),* 11725–11736. https://doi.org/10.1109/JIOT.2021.3132030
8. Elgzil, A., Chow, C. E., Aljaedi, A., Alamri, N. (2017). Cyber anonymity based on software-defined networking and onion routing (SOR). *IEEE Conference on Dependable and Secure Computing*, pp. 358–365. Taipei.
9. Zhu, T., Feng, D., Wang, F., Hua, Y., Shi, Q. et al. (2017). Efficient anonymous communication in SDN-based data center networks. *IEEE/ACM Transactions on Networking, 25(6),* 3767–3780. https://doi.org/10.1109/TNET.2017.2751616
10. Sharma, P., Singh, S., Jeong, Y. S., Park, J. H. (2017). DistBlockNet: A distributed blockchains-based secure SDN architecture for IoT networks. *IEEE Communications Magazine, 55(9),* 78–85. https://doi.org/10.1109/MCOM.2017.1700041
11. Yao, S., Wang, M., Qu, Q., Zhang, Z. Y., Zhang, Y. F. et al. (2022). Blockchain-empowered collaborative task offloading for cloud-edge-device computing. *IEEE Journal on Selected Areas in Communications, 40(12),* 3485–3500. https://doi.org/10.1109/JSAC.2022.3213358
12. Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM, 24(2),* 84–90. https://doi.org/10.1145/358549.358563
13. Hu, N. (2023). RAISE-Resilient Anonymous Information Sharing Environment. http://101.33.211.84/
14. Tor Metrics. https://metrics.torproject.org/
15. Cottrell, L., Palfrader, P., Sassaman, L. (2003). Mixmaster protocol. http://www.abditum.com/mixmasterspec.txt
16. Syverson, P., Dingledine, R., Mathewson, N. (2004). Tor: The secondgeneration onion router. *Proceedings of the 13th USENIX Security Symposium*, pp. 303–320. Boston.
17. Mittal, P., Wright, M., Borisov, N. (2012). Pisces: Anonymous communication using social networks. arXiv preprint arXiv:1208.6326.
18. Freedman, M. J., Morris, R. (2002). Tarzan: A peer-to-peer anonymizing network layer. *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 193–206. Washington.

19. Elgzil, A., Chow, C. E., Aljaedi, A., Alamri, N. (2017). Cyber anonymity based on software-defined networking and onion routing (SOR). *2017 IEEE Conference on Dependable and Secure Computing*, pp. 358–365. Taipei.
20. Meng, W., Tischhauser, E. W., Wang, Q., Wang, Y., Han, J. (2018). When intrusion detection meets blockchain technology: A review. *IEEE Access, 6,* 10179–10188. https://doi.org/10.1109/ACCESS.2018.2799854
21. Tian, Z., Li, M., Qiu, M., Sun, Y., Su, S. (2019). Block-DEF: A secure digital evidence framework using blockchain. *Information Sciences, 491,* 151–165. https://doi.org/10.1016/j.ins.2019.04.011
22. Derhab, A., Guerroumi, M., Belaoued, M., Cheikhrouhou, O. (2021). BMC-SDN: Blockchain-based multicontroller architecture for secure software-defined networks. *Wireless Communications and Mobile Computing, 2021,* 1–12. https://doi.org/10.1155/2021/9984666
23. Luo, J., Chen, Q., Yu, F. R., Tang, L. (2020). Blockchain-enabled software-defined industrial internet of things with deep reinforcement learning. *IEEE Internet of Things Journal, 7(6),* 5466–5480. https://doi.org/10.1109/JIoT.6488907