Co-IoT – A Collaborative DDoS mitigation scheme in IoT environment based on blockchain using SDN

Zakaria Abou El Houda [1,2], Abdelhakim Hafid [1] and Lyes Khoukhi [2]

[1] Department of Computer Science and Operational Research, University of Montreal, Canada
[2] ICD/ERA, University of Technology of Troyes, France
zakaria.abou.el.houda@umontreal.ca, ahafid@iro.umontreal.ca
{zakaria.abou_el_houda, lyes.khoukhi}@utt.fr

***Abstract*— The recent proliferation of Internet of Things (IoT) is paving the way for the emergence of smart cities, where billions of IoT devices are interconnected to provide novel pervasive services and automate our daily lives tasks (e.g., smart healthcare, smart home). However, as the number of insecure IoT devices continues to grow at a rapid rate, the impact of Distributed Denial-of-Service (DDoS) attacks is growing rapidly. With the advent of IoT botnets such as Mirai, the view towards IoT has changed from enabler of smart cities into a powerful amplifying tool for cyberattacks. This motivates the development of new techniques to provide flexibility and efficiency of decision making on the attack collaboration in a software defined networks (SDN) context. The new emerging technologies, such as SDN and blockchain, introduce new opportunities for low-cost, efficient and flexible DDoS attacks collaboration for the IoT based environment. In this paper, we propose Co-IoT, a blockchain-based framework for collaborative DDoS mitigation; it uses the concept of smart contracts (i.e., Ethereum's smart contracts) to facilitate the collaboration among SDN-based domains and transfer attacks information in a decentralized manner. The implementation of Co-IoT is deployed on Ethereum official test network Ropsten [1]. The experimental results confirm that Co-IoT achieves flexibility, efficiency, security and cost effectiveness making it a promising approach to mitigate large scale DDoS attacks.**

Keywords— IoT; DDoS; SDN; Smart contract; Blockchain;

I. INTRODUCTION

DDoS attacks are still considered serious network security threats due to the availability of amplifying platforms (e.g., Botnet as-a-service) for cyberattacks. They have evolved to be destructive and powerful causing severe collateral damage to network operators and service providers. The rapid growth in the number of insecure IoT devices, with an estimated 50 billion devices by the end of 2020 [2], can enhance and facilitate the capability of large-scale attacks. On October 2 2016, Mirai botnet commanded a huge number of IoT devices (i.e., closed-circuit television cameras (CCTV)) to conduct a DDoS attack against Dyn DNS infrastructure, as a consequence, many popular Internet services, e.g., Amazon, Twitter, GitHub and PayPal became unavailable for several hours [3]. This attack [3] is considered as the largest ever DDoS attack, exceeding a rate of 1 Tbit/s. Recently, the Mirai source code was publicly released; consequently, a large number of insecure IoT devices have since been used to create large-scale botnets. This growing threat harms ISPs and cost millions of dollars of lost revenues for enterprises.

The continuous growth in size and complexity of current networks, such as enterprise networks and data centers, is giving rise to SDN as a novel technology that facilitates network management and provides new approaches to deploy and manage networks dynamically [4] – [9]. SDN separates data and control planes; this separation allows for more control over the network and brings a new way to deal with DDoS attacks.

Existing collaborative DDoS mitigation schemes [15-24] suffer from low flexibility, high cost and implementation complexity; more importantly, they are centralized. The centralized solution, by its nature, brings single-point-of-failure and is vulnerable to DDoS attacks that can make it difficult, or even infeasible, to share information, among ASs, and make effective decisions to mitigate the attacks. The new emerging technologies, such as blockchain, open new opportunities for low-cost, efficient and flexible collaboration across multiple ASs to mitigate DDoS attacks. Indeed, blockchain has been investigated to provide a decentralized collaboration in trustless network environments.

This paper presents the design, specification and implementation of a blockchain-based approach called Co-IoT; it provides an efficient mitigation along the path of an ongoing attack and effective mitigation near of source of the attack. The implementation of Co-IoT is deployed on Ethereum official test network Ropsten, an open blockchain platform. Our main contributions can be summarized as follows:

- We design a decentralized secure DDoS collaboration scheme (Co-IoT) based on blockchain using smart contract.
- We propose a smart contract-based scheme that makes use of Ethereum, to realize a decentralized, secure, flexible and low-cost collaboration, among multiple SDN-based domains, to mitigate against DDoS attacks.
- We evaluate the performance of Co-IoT in terms of flexibility, efficiency, security and cost effectiveness. The experiments results show that Co-IoT can effectively ensure a secure collaboration among multiple SDN-based domains and achieves the requirements of the new generation of flexible, secure, efficient and low-cost DDoS collaboration schemes.

This paper is organized as follows. Section II presents related work. Section III presents Co-IoT. Section IV presents the implementation of Co-IoT. Section V evaluates Co-IoT. Finally, Section VI concludes the paper and presents future work.

II. Related work

Blockchain technology (e.g., Bitcoin [10] and Ethereum [11]) is considered as a new technology to secure and store information in a decentralized manner without any trusted tier; it has proven its effectiveness and success in multiple application domains (e.g., Healthcare [12], financial field [13]) in achieving high level of security and transparency. One such application domain is the IoT [14] due to its decentralized structure and the resource-constraints of its devices. Using blockchain technology, which ensures trust between nodes in a trustless environment, can be an efficient approach to facilitate the future underlying infrastructure for IoT. Security and privacy for IoT have been an active research topic for decades and several collaboration DDoS mitigation schemes have been proposed. Here, we present some of the most prominent as well as their security issues.

In [15], IETF (Internet Engineering Task Force) proposes the development of a new collaborative protocol called DOTS (DDoS Open Threat Signaling) to advertise DDoS attacks. In DOTS protocol, there are DOTS client and DOTS controller. When DOTS detects an attack, DOTS client requests the mitigation service from DOTS controller that is responsible for inter-domain communication and coordination. The effectiveness of DOTS depends on global deployment which may be infeasible due to implementation complexity; moreover, the collaboration process can be easily compromised. To alleviate this, a secure public-key infrastructure (PKI) can be used; nonetheless, it is costly to maintain and setup and. In [16], Steinberger et al. proposed a similar scheme to DOTS [15]; it uses flow-based event exchange format in order to simplify the deployment and the collaboration between domains. This scheme [16] also requires PKI which is costly to maintain and setup. In [17], Giotis et al. proposed a collaborative DDoS mitigation scheme across multiple SDN based domains. They extend Border Gateway Protocol (BGP) protocol to repost incidents as URIs in BGP signals. However, any modification to BGP is a challenging endeavor. Moreover, the incident report latency may be large given that domains do not report in real-time. More importantly, this scheme [17] does not verify the authenticity of incident reports resulting in a scheme that is vulnerable to spoofed incident reports from illegitimate domains. In [18], G. Zhang et al. proposed a gossip-based approach to exchange attacks information between detection points. This scheme [18] is built as a peer-to peer overlay network to disseminate attacks information to other peers rapidly. A similar scheme was proposed in [19], using also a gossip-based protocol to exchange information in overlay network. However, the deployment and integration of such schemes become complex since existing solutions need to be modified to support these protocols. In [20], Bahman et al. proposed a DDoS defense mechanism, called CoFence, to facilitate collaboration among network function virtualization (NFV) based domains. When a NFV-based domain is under attack, it redirects traffic to other NFV-based domains to filter the packets. First, CoFence has a privacy issue since it redirects traffic to other NFV-based domains. Moreover, this process of redirection increases incident report latency. In [21], B. Rodrigues et al. proposed a scheme, which uses blockchain and smart contracts, to advertise blacklisted IP addresses. However, this scheme [21] requires a central entity to issue certificates of ownership of IP addresses. In [22], G. Spathoulas *et al*. proposed the integration of a reputation scoring scheme in the Ethereum smart contract for malicious reporting of IPs addresses. However, the process of collaboration can be easily compromised. Many other schemes have been proposed (e.g., [23]-[24]); however, the complexity of deployment and overhead, that is generated, remain challenging issues in these schemes

To address the weaknesses of existing solutions [15-24], we propose a secure, efficient, easy-to-deploy and low-cost inter domain mitigation scheme; it allows multiple SDN based domains to securely collaborate and transfer attack information in a decentralized manner based on blockchain using smart contract. The use of new technologies, such as blockchain, helps avoiding the complexity of developing new protocols and/or the modification of existing ones (e.g., [17]-[19]); in addition, Co-IoT removes the need to use a central entity and enforces permissions to participate in the collaboration.

III. Co-IoT

A. Overview

In this section, we present an overview of Co-IoT. More specifically, we briefly describe how Co-IoT effectively enables collaboration to transfer attacks information (i.e., IP addresses that are suspicious of generating attacks) in a secure decentralized manner using blockchain.

As DDoS attacks evolve rapidly and become more devastating, cooperation among several domains has become very necessary to ensure sophisticated mitigation across multiple domains and cope with large scale DDoS attacks.

DDoS collaboration requires multiple SDN based domains (e.g., ASs: A, B, C, D, E and F) to collaborate as shown in Fig. 1. SDN based domains communicate with each other using Co-IoT which is based on blockchain and smart contract. First, the owner of the smart contract needs to create the collaboration contract. Then, he adds authorized participants (i.e., collaborators). Therefore, when attackers, which are distributed across multiple domains, control a large number of compromised IoT devices and generate an attack towards the victim (e.g., hosted at AS C; see Fig. 1), the domain under attacks uses a mitigation scheme (e.g. [25]) to detect and mitigates the attack; it also stores the suspicious IP address, denoted by ip_address in the smart contract. When the next block is mined (each 14 seconds in Ethereum blockchain), each of the authorized participants of the collaborative scheme will have access to the list of addresses to be blocked (i.e., suspicious IP addresses). This will allow for an efficient mitigation along the path of an ongoing attack and an effective mitigation near to the origin of attack. To report/receive attack information, each SDN based domain runs an instance of one of the Ethereum client (e.g., geth client [26]). Each SDN based domain collaboratively creates and maintain a list of IP addresses of malicious IoT devices generating the attack.

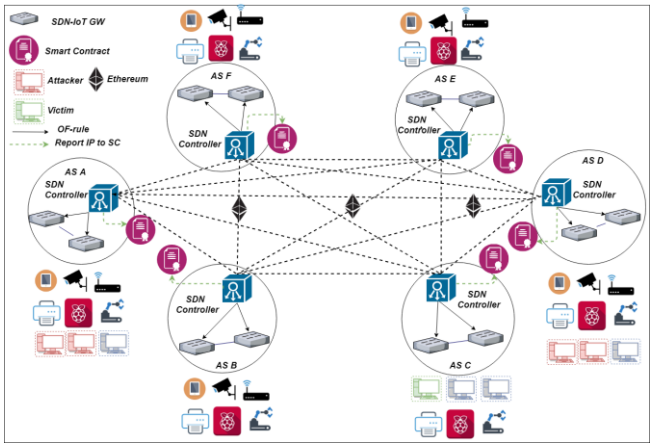

Fig.1 Co-IoT blockchain-based framework

Fig.2 shows the high-level architecture of *Co-IoT*. ASs are classified into 3 types of network domains, *source domain*, *intermediate network domains* and a *destination domain*. The *source domain is* the network (i.e., AS) in which the attacker starts the attack. *Intermediate network domains* forward illegitimate traffic. The *destination domain* is the domain where the victim is hosted. Once the SDN controller of the victim domain detects the attack, it mitigates the attack inside the domain (1), e.g., it uses a scheme like our previous work [25]. Then, SDN controller of the victim domain, sends a transaction, if it is authorized, to the Ethereum smart contract to report suspicious ip_address (2). Once the transaction is confirmed, an event is emmitted by the contract and is received by the registered/authorized collaborators of the smart contract (3), e.g., SDN controller of *source and intermediate network domains*. Finally, upon receipt of the event, the collaborators block the illegitimate traffic close to its source (4).

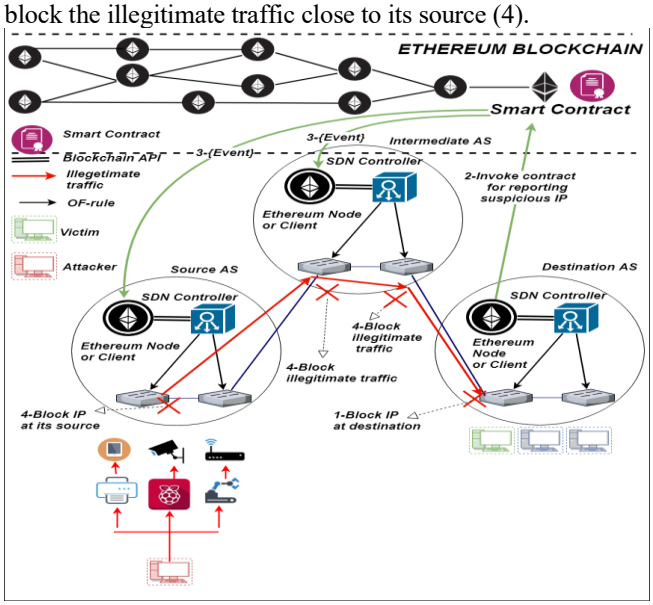

Fig. 2. High-level architecture of Co-IoT.

B. *Co-IoT'S SMART CONTRACT*

We consider an organization (i.e., contract owner) that would like to manage a collaboration process between different ASs around the world. First, it creates Co-IoT's smart contract and deploys it on the Ethereum blockchain. The use of the blockchain in the collaboration process allows for transparency while maintaining "pseudonymity". First, to initialize Co-IoT, the organization (o) generates a keypair of private key and the corresponding Externally Owned Account (EOA) (i.e., address of EOA is the hash of the corresponding public key). This keypair will be used to create the smart contract (SC) and execute the functions of the SC (see Fig.3). The creation of the keypair can be done using several options(e.g., Ethereum wallets, MetaMask [27]). We denote o.EPK and o.EOA as the private key and the EOA of the owner of the collaboration contarct, respectively. Then, the organization adds, via the smart contract, the collaborators into the system. It includes the collaborator's address and some other information (e.g., collaborator notes). The smart contract allows (1) the organization to add collaborators to the contract; (2) the organization to manage and modify the collaboration process in a transparent manner; (3) the organization to delete collaborators from the collaboration process if needed; (4) collaborators to report suspicious ip_address in a secure and efficient manner; and (5) collaborators to delete the reported ip_address from the contract if needed.

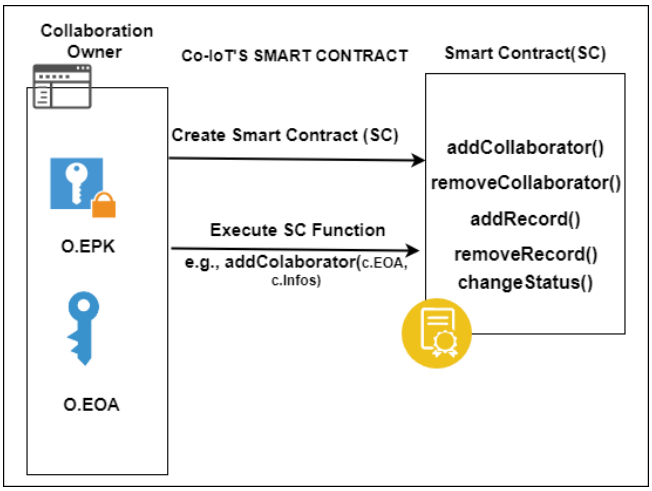

Fig.3. Co-IoT'S smart contract

The *Collaboration Contract* mainly provides the following functions where c denotes an instance of Collaborator and r an instance of record (i.e., the suspicious IP address):

addCollaborator(c.EOA, c.Infos): This function can only be invoked by the owner of the smart contract to add collaborators; it takes as input the Externally Owned Account (c.EOA) of the collaborator and the information about the collaborator (c.Infos) and adds the collaborator to smart contract as well as the timestamp of when the collaborator was added. This happens if the contract is activated and the colloborator' identity is authenticated.

removeCollaborator(c.EOA): This function can only be invoked by the owner of the smart contract to remove collaborators; it takes as input the Externally Owned Account (c.EOA) of the collaborator and removes the collaborator from the smart contract.

addRecord(r.IP): This function can only be invoked by either the owner of the smart contract or the collaborator that has already been added in the smart contract to report suspicious ip_address. It takes as input the suspicious ip_address and adds the record to the smart contract.

TABLE 1: Transaction details of Co-IoT

	Details of Co-IoT Creation Transaction in Ropsten test network
TxHash	0xc799243025bdf546cd70c81262ee3b99ae5e52b5d0c7a7a28d800c1574a1fe02
Block Height	5569072(1207 Block Confirmations)
Timestamp	May-10-2019 10:39:47 AM +UTC
From	0xa70836a9a115f774cb848134d0f8b2473e27d181
To	0x8dc749bec875edebaae59d8d6b302b698a7e6e95
Gas Used by Tx	1765888

removeRecord(r.IP): This function can only be invoked by either the owner of the smart contract or the collaborator to remove records; it takes as input an IP address and removes the corresponding record, if it exists, from the smart contract.

ChangeStatus(bool status): This function can only be invoked by the owner of the smart contract to either activate or deactivate the smart contract.

IV.IMPLEMENTATION

We implemented *Co-IoT* using both private (Ganache simulator [28]) and public blockchain (Ethereum official test network Ropsten). Once the smart contract is deployed, it can be self-executed without any human intervention. The deployment process is elaborated using truffle framework [29], a decentralized application development framework (see Fig. 4). First, we code the collaboration contract using the high-level language programming solidity [30]. Then, we compile the contract into EVM byte code; once the contract gets compiled, it generates the EVM byte code and Application Binary Interface (ABI). Afterwards, we deploy the smart contract to the blockchain. Initially, we have deployed the smart contract on a private blockchain using Ganache, an Ethereum simulator used for testing the smart contract in a fast way. Then, we have deployed the smart contract on Ethereum official test network Ropsten. The contract lifecycle is shown in Fig.5. Once deployed, the contract can be invoked using ABI definition and the address of the contract. If needed, the contract can be deleted (cannot be invoked anymore). Table 1 shows *Co-IoT* creation transaction in Ropsten official test network. The details of a given transaction can be found using Ropsten Etherscan [31].

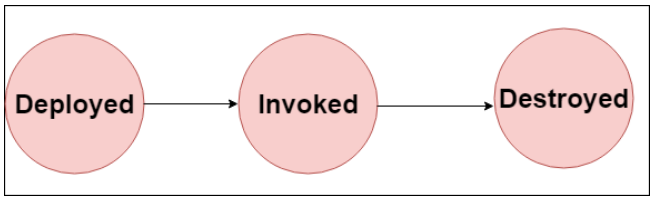

Fig.4. The deployment processes

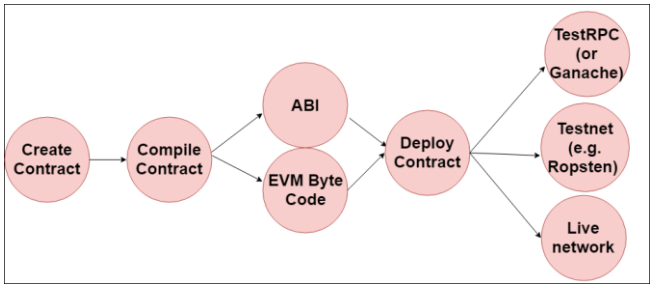

Fig.5. The contract lifecycle

V. EVALUATION

The main objective of *Co-IoT* is to provide a secure, easy-to-deploy, low-cost, efficient and flexible DDoS attacks mitigation scheme based on Ethereum blockchain using smart contract. In this section, we answer the question: how does *Co-IoT* provide these features?

1. *Flexibility/Easy-to-Deploy*

Co-IoT provides two levels of flexibility: (1) *Co-IoT* provides the organization (i.e., contract owner) with the flexibility to easily (add/remove) collaborators (to/from) the system using (addCollaborator()/removeCollaborator()) functions. Similarly, collaborators can easily (add/remove) records (to/from) the system using (addRecord()/removeRecord()) functions; and (2) *Co-IoT* provides the organization (contract owner) with the flexibility to easily join or leave the system. To join the system, the organization needs to deploy the Collaboration contract. To leave the system, the organization can easily deactivate the contract using ChangeStatus() function. All these updates can be verified by anyone in the network (Ethereum blockchain network).

2. *Security/Eligibility*

Only authorized collaborators that have permissions can report suspicious ip_address. This is achieved by *Co-IoT* using modifiers. For example, the modifier "OnlyOwner" allows only the owner of the contract to execute the addCollaborator(), removeCollaborator() and changeStatus() functions. If a compromised user tries to execute these functions in order to either add illegitimtae collaborators to report fake IP addresses or remove legitimate collaboartors, the execution will fail and no action will be recorded on the blockchain. The same restriction rule applies for the "OnlyCollaborators" modifier for the execution of addRecord() and removeRecord() functions; only

collaborators (and also the contract owner) can add/remove the records.

3. Low cost

In this section, we estimate the cost of the creation of the collaboration contract as well as the execution of each function used in *Co-IoT*. When we conducted the experiment, the gasPrice was set to $1Gwei$, where $1Gwei = 10^{9}wei = 10^{-9}ether$, and , and 1 ether was equal to $172{,}53USD$.

TABLE 2: Co-IoT creation and functions costs

Function	Gas Used	Actual Cost(ether)	USD
create Co-IoT	1765888	0.00176588	0.304
addCollaborator()	139826	0.000139826	0.024
removeCollaborator()	38427	0.000038427	0.006
addRecord()	121350	0.00012135	0.002
removeRecord()	26807	0.000026807	0.004
changeStatus()	29017	0.000029017	0.005

Table 2 shows the cost of the execution of different functions in *Co-IoT*. We observe that the highest cost corresponds to the creation of *Co-IoT* at 0.304 USD. However, it is performed only once to setup the collaboration system. All functions, provided by the smart contract, have low costs. Thus, *Co-IoT* is cost effective compared to exiting related schemes.

4. Analysis

First, Co-IoT preserves pseudonymity and does not allows traceability of identities of collaborators (e.g., IP address of the collaborator). *Co-IoT* does not suffer from single point of failure problem since it runs on Ethereum (over 16000 nodes running the smart contract [32]). Furthermore, it is decentralized scheme; thus, there is no need to a centralized authority (or a third party) to maintain the collaboration system; the reliability and availability of the records, recorded on the blockchain, are guaranteed. Co-IoT removes the need to use a central entity and enforces permissions to participate in the collaboration in contrast to existing schemes (i.e., [21] and [22]) using blockchain-based collaboration schemes.

VI. CONCLUSION

In this paper, we proposed a smart contract-based framework that makes use of Ethereum's smart contract technology to facilitate the collaboration among SDN-based domain peers. The collaboration contract has been tested/evaluated and deployed on Ethereum official test network Ropsten; the appendix shows Co-IoT address in Ropsten. For future work, we intend to integrate our previous work [25] to Co-IoT in order to ensure two levels of mitigation (i.e., intra-domain and inter-domain DDoS mitigation).

APPENDIX

The collaboration contract was deployed on the Ropsten Testnet of Ethereum with the following address:

Organization Owner of account address: 0xa70836a9a115f774cb848134d0f8b2473e27d181.
Co-IoT address: 0x8dc749bec875edebaae59d8d6b302b698a7e6e95
Using this address, the transactions can be seen at: https://ropsten.etherscan.io/.

REFERENCES

[1] Etherscan. The Ethereum Block Explorer: ROPSTEN (Revival) TESTNET. Accessed: Mai. 1, 2019. [Online]. Available: https://ropsten.etherscan.io.

[2] D. Evans, "The internet of things: How the next evolution of the internet is changing everything" CISCO white paper, vol. 1, no. 2011, pp. 1-11, 2011.

[3] B. Schneier. Lessons From the Dyn DDoS Attack. Accessed: Mai. 1, 2019. [Online]. Available: https://www.schneier.com/blog/archives/2016/11/lessons_from_th_5.html.

[4] K. Kalkan, L. Altay, G. Gür and F. Alagöz, "JESS: Joint Entropy-Based DDoS Defense Scheme in SDN," in *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358-2372, Oct. 2018.

[5] P. Kumar, M. Tripathi, A. Nehra, M. Conti and C. Lal, "SAFETY: Early Detection and Mitigation of TCP SYN Flood Utilizing Entropy in SDN," in *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1545-1559, Dec. 2018.

[6] K. Kalkan, G. Gur and F. Alagoz, "Defense Mechanisms against DDoS Attacks in SDN Environment," in *IEEE Communications Magazine*, vol. 55, no. 9, pp. 175-179, Sept. 2017.

[7] Yan, Q., Yu, F.R., Gong, Q., and Li, J.,'Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges', IEEE Commun. Surv. Tutor., 18, pp. 602–622, 2016.

[8] Y. Yu, L. Guo, Y. Liu, J. Zheng and Y. Zong, "An Efficient SDN-Based DDoS Attack Detection and Rapid Response Platform in Vehicular Networks," in *IEEE Access*, vol. 6, pp. 44570-44579, 2018.

[9] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 623–654, 2015.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[11] G. Wood, ''Ethereum: A secure decentralised generalised transaction ledger,'' Yellow Paper. Accessed: Jan. 1, 2019. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[12] J. Zhang, N. Xue and X. Huang, "A Secure System For Pervasive Social Network-Based Healthcare," in IEEE Access, vol. 4, pp. 9239-9250, 2016.

[13] Blockchain for Financial Services. Accessed: Mai. 1, 2019. [Online]. Available: https://www.ibm.com/blockchain/financial-services

[14] P. K. Sharma, M. Chen and J. H. Park, "A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT," in IEEE Access, vol. 6, pp. 115-124, 2018.

[15] K.Nishizuka, L.Xia, J.Xia, D.Zhang, L.Fang, C.Gray : "Inter-organization cooperative DDOS protection mechanism". Accessed: Mai. 1, 2019. [Online]. Available:Draft. https://tools.ietf.org/html/draft-nishizuka-dots-inter-domain-mechanism-02.

[16] Steinberger, J., Kuhnert, B., Sperotto, A., Baier, H., Pras, A.: Collaborative DDOS defense using flow-based security event information. In: NOMS 2016–2016 IEEE/IFIP Network Operations and Management Symposium, pp. 516–522, April 2016.

[17] K. Giotis, M. Apostolaki, and V. Maglaris. A reputation-based collaborative schema for the mitigation of distributed attacks in sdn domains. In IEEE/IFIP Network Operations and Management Symposium, 2016.

[18] Zhang, G., Parashar, M.: "Cooperative defence against DDoS attacks". J. Res. Pract. Inf. Technol. 38(1), 69–84 (2006)

[19] Velauthapillai, T., Harwood, A., Karunasekera, S.:" Global detection of flooding based DDOS attacks using a cooperative overlay network". In: Network and System Security (NSS), pp. 357–364. IEEE (2010)

[20] B. Rashidi, C. Fung and E. Bertino, "A Collaborative DDoS Defence Framework Using Network Function Virtualization," in *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2483-2497, Oct. 2017.

[21] B. Rodrigues, T. Bocek, A. Lareida, D. Hausheer, S. Rafati, and B. Stiller, "A blockchain-based architecture for collaborative DDoS mitigation with smart contracts," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence

and Lecture Notes in Bioinformatics), ser. Lecture Notes in Computer Science, D. Tuncer, R. Koch, R. Badonnel, and B. Stiller, Eds., vol. 10356 LNCS.

[22] G. Spathoulas *et al*., "Towards Reliable Integrity in Blacklisting: Facing Malicious IPs in GHOST Smart Contracts," *2018 Innovations in Intelligent Systems and Applications (INISTA)*, Thessaloniki, 2018, pp. 1-8.

[23] Y. Chen, K. Hwang and W. Ku., "Collaborative Detection of DDoS Attacks over Multiple Network Domains," in IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 12, pp. 1649-1662, Dec. 2007.

[24] S. Simpson, S. N. Shirazi, A. Marnerides, S. Jouet, D. Pezaros and D. Hutchison, "An Inter-Domain Collaboration Scheme to Remedy DDoS Attacks in Computer Networks," in *IEEE Transactions on Network and Service Management*, Sept. 2018.

[25] Z. A. El Houda, L. Khoukhi and A. Hafid, "ChainSecure - A Scalable and Proactive Solution for Protecting Blockchain Applications Using SDN", 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-6.

[26] Go Ethereum. Accessed: Mai. 1, 2019. [Online]. Available: https://geth.ethereum.org/.

[27] Metamask. Accessed: Mai. 1, 2019. [Online]. https://metamask.io/

[28] Ganache. Accessed: Mai. 1, 2019. [Online]. Available: https://truffleframework.com/docs/ganache/overview.

[29] Truffle. Accessed: Mai. 1, 2019. [Online]. Available: https://truffleframework.com/

[30] "Solidity", Accessed: Mai. 1, 2019. [Online]. Available: https://solidity.readthedocs.io/en/develop/.

[31] Ropsten. Accessed: Mai. 1, 2019. [Online]. Available: https://ropsten.etherscan.io/

[32] Ethereum node. Accessed: Mai. 1, 2019. [Online]. https://www.ethernodes.org/network/1