International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

DDoS Mitigation by Blockchain With Approach of Cost Model

Laxmi Poonia
Department of Computer Science and Engineering, JECRC University
Jaipur-303905, Rajasthan, India
laxmipoonia022@gmail.com

Seema Tinker
Department of Mathematics, JECRC University,
Jaipur-303905, Rajasthan, India
seematinker@gmail.com

Abstract—Computer networks and internet services are increasingly threatened by attacks like Distributed Denial-of-Service (DDoS). DDoS attack mitigation techniques now in use are ineffective due to a lack of resources and a lack of adaptability. Using blockchains like Ethereum, DDoS attacks can be thwarted in innovative ways. With smart contracts, it is possible to track down the IP addresses of attackers without additional hardware. This study examines blockchain-based solutions to combat DDoS attacks for feasibility, effectiveness, as well as cost and performance. The cost model delves into economic aspects like gas, gas price, and Ether value. In it, the evaluation of various smart contracts for the signalization of DDoS attacks is documented and compared to assess three system variants, analyzing gas costs, deployment, speed, and accuracy. It also details Ethereum's ecosystem and how that affects smart contract design and it also acknowledges scalability challenges and suggests outsourcing data for a more scalable solution, advocating for specialized blockchains for DDoS signaling applications. The analysis provides insights into the gas costs associated with different variants, considering various scenarios and highlighting the trade-offs and efficiencies of each approach.

Keywords-Attack, Block-chain, DDoS, Innovative, Internet, Ethereum, Smart contracts

I. Introduction

It is a chain of 'blocks' that make up a decentralized database known as a Blockchain. Each block is linked to the one before it and cannot be altered without causing the subsequent blocks to break. As fresh data is added to the chain's terminus, the blockchain expands. Digital currency, the most commonly used implementation of which is Bitcoin, is the most popular application for blockchains [1]. Users of the Bitcoin network can securely exchange tokens within a system that is entirely decentralized, free from central control and intermediaries. As of August 2017, the market value of Bitcoin was above $60 billion, which has become $725 billion as of November 2023, and its tokens were actively traded by users on digital exchanges. While Ethereum and Bitcoin share many similarities, the latter features a scripting language called Solidity that makes it possible for anybody to create blockchain-compatible applications. Ethereum has the potential to host many different types of applications, including those for gaming and financial services like venture capital funds and initial coin offerings (a company raising funds by selling shares of itself to investors) [2]. The automatic enforcement of obligations is a key feature of smart contracts. The designer of a smart contract has complete discretion over the criteria and behaviors of mutations and the access levels granted to users by providing the necessary code. The presented DDoS mitigation system can be implemented on the Ethereum blockchain because of the network's Turing completeness [3].

Disruption of a machine or network resource that should be available is known as a Denial of Service (DoS). This can either be done by constructing a request payload that requires the target system to perform extensive computation, or by flooding it with requests. There are a variety of reasons for a DoS assault, including a dislike of the victim's service, a competitor, or a desire to disrupt the victim's business. It is possible to launch numerous distinct Denial-of-Service attacks at the same time as a DDoS attack [4]. A Denial of Service attack's traffic volume can soar and its control more difficult because of the attack's dispersed requests. As many internet-connected devices as possible are taken over by the attacker, who then directs them to attack the victim. To stop a DDoS assault, simply block the attacker's traffic. An IP (Internet Protocol) address is a unique identification included in every packet of data. The attack can be prevented by looking at the traffic based on the source IP address. IP addresses of DDoS attackers can be tracked down and reported to upstream providers, who can stop traffic before it reaches a victim's network [5].

This paper explores implementing a DDoS attack notification system on the Ethereum blockchain using smart contracts. It discusses blockchain technology, Ethereum, and the challenges of DDoS attacks. The background covers existing DDoS defense technologies and introduces Ethereum as a decentralized solution. The workflow addresses testing challenges and the Solidity compiler, emphasizing IP address ownership verification difficulties. Security considerations highlight vulnerabilities in smart contracts. The cost model delves into economic aspects like gas, gas price, and Ether value. The evaluation section assesses three system variants, analyzing gas costs, deployment, speed, and accuracy. The result acknowledges scalability challenges and suggests outsourcing data for a more scalable solution, advocating for specialized blockchains for DDoS signaling applications.

II. Background detail

DDoS attacks, growing in frequency and sophistication, target diverse sectors worldwide, using novel techniques like ransom-driven strategies. Shorter, intense attacks pose challenges to countermeasures, impacting business continuity and necessitating effective mitigation strategies. These attacks are often employed as smokescreens for more malicious actions, emphasizing the need for proactive security measures. Organizations need to stay updated on evolving threats and adopt robust DDoS mitigation strategies in the dynamic cybersecurity landscape [6].

Blockchain technology has solidified its presence in real-world applications, offering advantages such as accelerated cross-border payments, identity management, smart contracts, cryptocurrencies, and improvements in supply chain processes. This technology, akin to the transformative impact of the Internet, has established itself as a lasting innovation. Unlike previous attempts at digital currency, blockchain overcomes security and trust issues by operating without the need for a central authority, placing control in the hands of its users. Its inherent resistance to alteration or forgery has sparked considerable market excitement and demand. Beyond cryptocurrency, blockchain has expanded its footprint into diverse practical applications, marking a shift toward simplification from its initial complex conceptualization. Noteworthy characteristics include decentralization, integrity, immutability, verification, fault tolerance, anonymity, audibility, and transparency [7]. There are two prominent types of blockchains: Bitcoin and Ethereum. Bitcoin primarily facilitates the transfer of digital assets, while Ethereum is primarily tailored for the execution of smart contracts. Smart contracts are essentially self-executing software responsible for managing or fulfilling contractual agreements. They operate independently on the blockchain infrastructure, running within a sandboxed Ethereum Virtual Machine. To mitigate Distributed Denial of Service (DDoS) attacks, smart contracts are executed and verified, incurring a cost measured in terms of "gas". The term gas refers to a unit of measurement for the computational effort required to execute operations or run programs on the Ethereum network. This term is integral to Ethereum's fee system, where users pay for the computation resources consumed during the execution of smart contracts. Researchers focusing on the intersection of DDoS and blockchain predominantly leverage the Ethereum blockchain and its structure based on smart contracts [8].

Multiple nodes in one network are using DefCOM, a peer-to-peer DDoS defense technology. When it comes to tasks, the framework features a distributed design. Classification, rate limiting, and alert generation are the maximum number of jobs that can be assigned to any given network node. In a network, each node can execute only the jobs it is best at. Message prioritization is also supported by the framework. Internally, DefCOM does not offer a solution for inter-organizational sharing [9]. DDoS response mechanisms are not included, but they provide a lightweight foundation for communication among nodes. DDoS Open Threat Communicating (DOTS) is a peer-to-peer protocol suggested to the Internet Engineering Task Force for signaling the originating IP addresses of distributed denial-of-service assaults. The authors' protocol interacts over HTTPS utilizing a REST-based API; this is because it is not decentralized. Organizations can talk to one another, or talk inside themselves. DOTS requirements include handshake calls, requests for mitigation with a variety of criteria, and updates on the effectiveness of the mitigation. So, the research in this paper was the impetus [10].

A. Workflow

Manual testing on the Ethereum main chain is not recommended during development. Because of the long transaction processing times, it is difficult for developers to gain fast feedback after publishing a contract into the main blockchain and incurring considerable fees. Because all network participants must download all blocks, using the main blockchain for testing purposes is also insensitive [11]. To conduct testing, an Ethereum blockchain known as the 'Testnet' was created. A global shared blockchain is not perfect for development, and the Testnet is no exception. To conduct testing on a local blockchain, the TestRPC library has been developed. Smart contracts may be deployed and transactions simulated instantly with TestRPC [12].

B. Compiler

The compiler included with the Solidity language is called Solc. The reference compiler was used since there were no specific rules against its use. In addition to Remix, a browser-based compiler, there are several other options [13].

C. Testing

Invalid code cannot be compiled by a Solidity compiler like Solc, which alerts the developer to the problem. Solc, for instance, will not compile code with incompatible operation types, redeclaration errors, invalid return types, or improper syntax. However, it does not provide full protection from runtime issues or gas limit mistakes, and it does not throw an error for unused variables, dead code, or missing arguments [14].

D. IP address ownership verification

A smart contract enables the owner of a destination IP address to furnish a list of source IP addresses that should consistently be denied access to the contract. The initial study proposed automating the verification of ownership for destination IP addresses but did not specify the implementation details [15].

While developing the prototype, this matter was explored, revealing its complexity. In Solidity, validating IP address ownership through certificates is currently impractical for several reasons. Ultimately, the challenge lies in achieving a certification. An indirect approach is required as there is no direct mathematical or logical proof establishing ownership of an IP address [16]. IP addresses might theoretically be subject to the domain certificate process. Issue and administering digital certificates is the business of Certificate Authorities (CAs). Certificates for domain ownership are issued by these experts. There are requirements that CAs must achieve before they will be accepted into the root key stores of OS and browser vendors. Firefox currently only trusts certificates from 60 distinct issuers. CAs must invest in a system that securely validates a domain and administers the certificates issued to ensure compliance with the demanding standards. A domain issuance fee is charged by practically all CAs that Firefox trusts. Even if an SSL (Secure Sockets Layer) certificate can be issued for an IP address, it is extremely rare. SSL is a cryptographic protocol that ensures secure communication over a computer network, commonly used for securing transactions on the internet. Firefox exclusively trusts certificates issued by GlobalSign, which is the only supplier that issues certificates for IP addresses. Since most IP addresses aren't in the RIPE database, a certificate isn't available to everyone [17]. A certificate costs $349 and requires that the IP address be registered. For IP address certificates, there are no viable suppliers, and building a certificate authority is an expensive operation. This concludes the description of the certificate issuance process. Even if certificates for IP addresses could be obtained and validated, the computational cost of doing so would likely exceed Ethereum's gas cap. However, this is only a hypothesis because of the lack of a certificate verification implementation in Solidity. It would be necessary to transfer OpenSSL's certificate verification code to Solidity, which is a laborious task [18]. However, there is a proposal to add language-level certificate validation. It's unclear exactly how this will be implemented at this time; some members of the community favor direct RSA signature verification, while others prefer BigInt Support, which could enable certificate validation. There is no need for certificate validation in Solidity because all data on the blockchain, including certificates and IP addresses, is accessible to the public. To guarantee the authenticity of the sender address, it is proposed that each Ethereum transaction be signed by the user and the network confirms the message sender value. A client's IP address can be pre-validated before they're added to a contract so that no reports can be added on behalf of the customer [19].

E. Security Considerations with Solidity

A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code. These contracts operate on blockchain technology, enabling them to be secure, transparent, and tamper-proof. Once a smart contract is deployed, its source code is usually made available so that users may verify the contract's behavior before they interact with it. As a result, there is a greater likelihood of discovering bugs. The data saved in the smart contract must be deemed public at all times. If you were to utilize this contract for gambling, you'd be able to get your hands on the first move from the blockchain: rock would be 0x60689557, while scissors and paper were each given a unique value. For the contracts specified in this contract, the most important takeaway is that the saved IP addresses will be available to everyone (even if disguised) [20]. Knowing the contract address makes it possible for attackers to discover the blacklisted IPs. In the event of a hack, Solidity author Christian Reitwiessner recommends activating a "fail safe" mode that locks the contract into a "read-only," "withdraw-only," state. According to Ethereum's creator, Vitalik Buterin, a list of vulnerabilities based on actual exploits has been compiled [21].

Gas limit failures can cause loops to become stopped. As a result, transaction parameters should not be used to limit the number of iterations in a loop. To avoid an overflow while using the var keyword, you should not use it in a for statement.

The "call stack depth" is the total number of nested function calls, which grows when one function calls another. An excessive call stack may be the result of excessive recursion. Solidity's call stack depth limit is 1024, hence it couldn't use recursion to get the 1025th Fibonacci number. This barrier is weak as a defensive measure. An attacker might write a code that repeatedly calls itself up to 1023 times before calling another vulnerable function on the 1024th iteration due to the call stack being full. When only a subset of a function is performed, the contract must be crafted to prevent exposure of the vulnerability to a call stack depth attack. The Ethereum core development team has proposed a language-level solution to this problem in EIP #150 [22].

III. COST MODEL

A reward system is needed since the blockchain is decentralized, and the people who verify transactions must be compensated for their work. This makes blockchain applications more expensive than a centralized service of the same type. A method for calculating consumption costs based on specific variables is provided [23].

A. Cost variables

There are at least five factors involved in a single transaction on the Ethereum blockchain. Block miners are paid a set amount of 'gas' each time they process a transaction or establish a new contract on the blockchain. Using the cost model, gas is considered a real-world expense. There are 31 possible assembly operations for each smart contract instance and transaction. ADD, which merely adds two numbers together, and SHA3, which calculates a hash value, are two elementary examples of such operations [24].

Listed under the heading 'Fee schedule,' the Ethereum yellow paper outlines the fees associated with various operations. SSTORE operations (storage operation in a smart contract), for example, cost 20000 gas, whereas transactions cost 21000 gas. As a rule of thumb, more complicated contracts and transactions cost more gas in general. For some reason, Ethereum's developers decided to set the fees for different types of operations at different amounts that aren't necessarily proportionate to the amount of computing work required. As a result of the non-proportionality, some operations may need to adjust their gas pricing in the future to restore equilibrium once the Metropolis hard fork, Ethereum's next version, has been widely adopted. A CALL procedure that costs 700 gas now costs 4000 gas after the Metropolis upgrade. These variables have been split in this model because of how volatile gas prices can be [25].

As a result, the Ethereum community and the Ethereum Foundation have a great deal of say over the cost of gas. Since the price of gasoline varies during the day, it is important to keep in mind that the same transaction could end up costing more or less gas depending on when it was memorized.

B. Gas price and desired speed

A specified amount of Ether is represented by one 'gas.' It is up to the users of the Ethereum network to decide on the price of gas, unlike the gas cost schedule, which is predetermined in the Ethereum clients and cannot be changed. A gas price is 'sent' by each Ethereum client. Go-Ethereum, the standard Ethereum implementation, has a gas price default of 20 shannon1, however, this can be changed. If the gas price in the miner's client is set to 20 Shannon or higher, the miner will only mine transactions with a gas price of 20 Shannon or above. If the gas price is set to 20 Shannon in a client, miners who receive at least that much from a transaction will be willing to pay that much for it. Etherscan.io's average gas price chart shows that the majority of the network's users don't deviate from the default setting of 23 Shannon [26]. Since consumers might suggest higher pricing, such as 24 Shannon, the true average gas price is slightly higher than the 20 Shannon level. This expedites the mining of the transaction. The workings of transaction mining can be compared to those of the stock market. A buyer's offer to purchase shares at an inflated price of, say, 102 USD will get to the top of the order book and be completed before any other orders for the actively traded stock with a market value of $100. Miners in Ethereum are prioritized based on the lowest gas prices. The ability to mine transactions more quickly can be worth more money for our application. The Ethereum community debates the default gas price and altered it once in March 2016 when Ether's price skyrocketed. After that, the price of gas was lowered from 50 to 20 shillings. Another hard fork may be incorporated now that Ether's price has hit a new level of scale [27]. At the time of publication, the equilibrium price of gas was calculated to be 16 Shannon on etherchain.org. According to data published on May 7th, 2017, 10% of the network's hash power is content with a gas price of merely 2 Shannon. Customers can help drive down gas prices by changing their account settings to display more affordable gas pricing on the site. Based on the same site's gas-time calculation, the average confirmation time for a 2 Shannon per gas transaction is 119 seconds. For 20 Shannon transactions, the average confirmation time is 44 seconds, whereas, for 28 Shannon transactions, it's only 30 seconds [28].

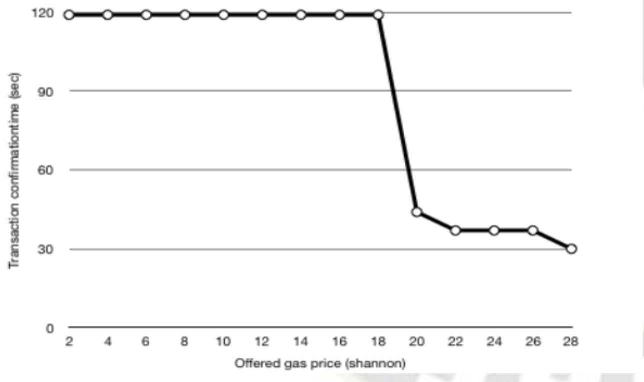


Figure 1. Average time until the transaction is confirmed

A larger reward for a transaction means that a node's confirmation rate is 4 times faster on average. Based on this information, the customer must decide how much gas he wants to offer.

C. *Price of Ether*

Ether can be obtained in two ways: through mining or by purchasing it on an exchange. On Nov 2021, the price of ether on the Coinbase exchange was $ 4444.53 and on Jan 2022, it was $ 2629.48, according to Dutta and Bouri [29], there are a lot of time-varying jumps in the Bitcoin market. On January 1st, 2017, the price of ether on the Coinbase exchange was $8.22 and on August 14th, 2017, it was $300.48. This shows the extreme volatility of the Ether pricing on exchanges [30]. Over 60 percent of Ether's value has been lost after a smart contract known as "TheDAO" was hacked in June 2016. Without any hacking incident, Ether's value decreased by two-thirds to $135 in just one month in July of 2017 [31]. The price of Ethereum dropped to $0.10 for a brief period in July 2017 on the GDAX (a Coinbase-operated exchange) because of a multi-million market sell order. Due to a lack of buy orders, the sell order could not be filled. There is a clear correlation between the price volatility and the lack of volume on Ethereum exchanges, indicating greater adoption of the cryptocurrency is needed to keep prices stable [32].

D. *Compiler being used*

The third factor is the discord between various compilers' gas estimates. This problem was discovered during a compiler upgrade as a result of receiving inconsistent gas price predictions for the same contract. The gas estimate is 318'552 gas in Solidity Compiler (solc) version 0.4.8, but it varies significantly between compilers [33].

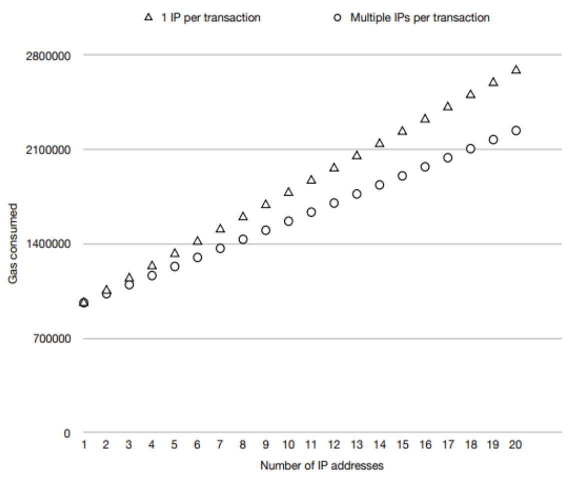
IV. EVALUATION

This section seeks to give a rough estimate of how much each smart contract will cost. Over time, the conversion rate of Ether to USD and the cost of network gas will change. Transaction confirmation speed is also a factor. As a precaution, only the amount of gas used is benchmarked in this section [34].

A. *Variant 1*

A benchmark script was written for variant 1. The benchmark calculates the total amount of gas used to create contracts and insert IP addresses. Calculations were made in two ways: In the worst-case scenario, each IP address was entered individually. Reports were combined into a single transaction in the best-case scenario. First, the benchmark ran on one address, then two addresses, and so on until it reached a total of 20 IP addresses [35].

Gas cost incurred using variant 1



B. *Variant 2*

It's easy to estimate the costs of additional infrastructure, but it's far more difficult to do so for alternative 2 (web resource pointer). The deployment cost was estimated using the estimate-gas script2. A benchmark script was developed for transaction costs3. The deployment consumes 600,000 gas, and each update consumes 150,000 gas. In terms of gas, version 2 is the cheapest. In practice, the costs range from $0.15 to $6.3, but they remain constant regardless of the number of IP addresses registered [36].

The additional infrastructure expenses are hard to predict because the standard only specifies a small subset of the available formats and leaves many implementation elements up to the user, such as which hosting provider to utilize.

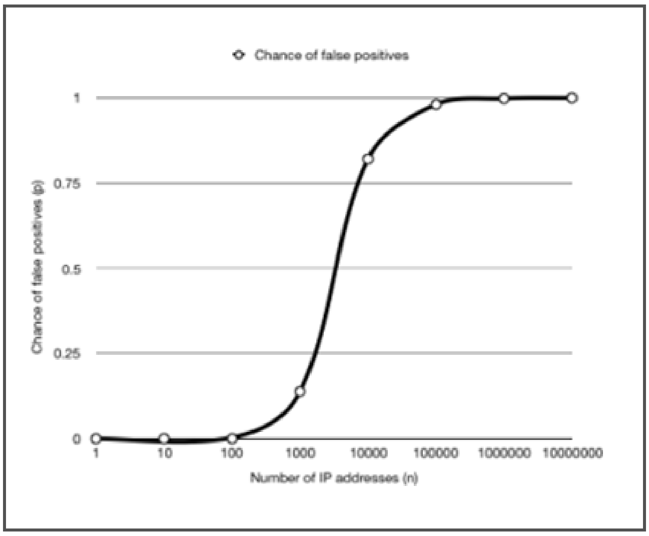


Figure 2. Probability of false positives

C. *Variant 3*

For variant 3, a unique benchmark script was developed (the bloom filter variant). We looked at the worst-case and best-case scenarios, which are quite similar to variant 1: in the worst case, each report is posted separately, while in the best case, multiple reports can be uploaded in a single transaction to reduce our gas use [37].

Because it takes up less space to keep all of the IP addresses, this variant was projected to perform better than variant 1. The benchmark, on the other hand, does not support this conclusion. When compared to storing an entire array of IP addresses, the bloom filter uses more gas. More than three times as much gas is used in variation 2 (105'000 gas) just to add one report. Because the price increases linearly after the first report, there are no cost savings to be achieved. Combining multiple reports into a single transaction saves 13% on gas, however, it is less effective than variant 1. In addition, there is a block gas limit of 17 reports that can't be added to a single transaction [38].

D. *Speed*

The actual rate is dynamic and is affected by several network parameters. The rate at which reports are inserted can also be modified by adjusting the quantity of gas allocated to the transaction. Because of these considerations, measuring speed without making assumptions about the network and user preference is challenging. We compute and map the range of speeds to the range of prices. Currently, the speeds that can be achieved span from 30 seconds to 120 seconds. Given the same gas price, all Ethereum transactions will take between these timeframes, making them all approximately the same speed [39].

E. *Accuracy*

Data in both Variant 1 (reports) and 2 (IP addresses) is saved in a lossless format, guaranteeing that any retrieved information will be identical to that which was entered. In contrast, the bloom filter does cause some accuracy loss in Variant 3; the exact amount is calculated below. Bloom filters' appropriate array size and the number of hash functions can be calculated given the number of indexed items and the desired level of performance. Research suggests that we may approximate the number of bits in the filter, m if we know two other parameters: n = the number of items in the filter, and p = the probability of a false positive [40, 41].

The array size of 14357134 bits (1.71 MB) and 4 hash functions will be sufficient if the scale of a DDoS attack is predicted to be in the millions (n = 1'000'000) and a false positive rate of 5 percent is acceptable (p = 0.05).

For up to 425 IP addresses, the bloom filter does have a false positive rate of less than 1%. Adding more IP addresses after that has a significant impact on the bloom filter's accuracy [42,43]. If 1000 insertions were added to version 3, 14% of legitimate traffic would be banned. Over half of all legitimate traffic would be mistakenly banned with 3'000 insertions. If the likelihood of false positives under the gas limit constraint is large enough to be unacceptable, then it is recommended to employ multiple contracts. Each user has a threshold for an acceptable rate of false positives [44,45].

V. Discussion

Ethereum is a new infrastructure for developing distributed programs of all stripes. A wide range of issues can be addressed by smart contracts that have been coded in the Turing-complete programming language Solidity [46,47].

Ethereum must make compromises to facilitate decentralized applications, such as dissenting computation-heavy or space-inefficient applications with cost and limits. The price of an Ethereum application depends on several factors. The price is affected by the characteristics of Ethereum clients, the current Ether price, and the complexity of the application. There is a direct correlation between transaction speed and cost, with greater fees associated with quicker transactions [48-50].

The analysis of smart contract cost estimation for different variants reveals several key insights.

Gas as the Benchmark: Gas usage serves as a fundamental metric for estimating costs in the Ethereum blockchain environment, with a focus on transaction efficiency.

Variant Comparisons:

Variant 1: The benchmarking of Variant 1 demonstrated different scenarios, emphasizing the trade-off between individual and batch transactions for IP address entries.

Variant 2: Despite challenges in estimating infrastructure costs, Variant 2 proved to be economically efficient in terms of gas, with consistent costs for deployment and updates.

Variant 3: The bloom filter variant, anticipated to perform better due to space efficiency, did not align with expectations. Gas costs were higher than storing the entire array of IP addresses, challenging the presumed advantages.

Implications:

Gas Costs and Predictability: Gas costs play a crucial role in determining the economic viability of smart contracts. The predictability of costs, especially in Variant 2, enhances financial planning for contract deployment and maintenance.

Space Efficiency vs. Gas Costs: The analysis underscores the importance of carefully weighing space efficiency against gas costs. While certain variants may seem theoretically advantageous, practical benchmarks reveal the actual trade-offs.

Recommendations:

Consideration of Gas Costs: Future smart contract implementations should prioritize an in-depth consideration of gas costs, ensuring that the chosen design aligns with economic efficiency and scalability.

Continuous Monitoring and Adaptation: Given the dynamic nature of blockchain environments, ongoing monitoring and adaptation of smart contract designs are essential to respond to changes in gas prices, network conditions, and overall system performance.

VI. Conclusion

We built, tested, and compared three versions of a DDoS attack notification smart contract for the Ethereum platform. All versions can be used to store IP addresses, and their functionality is not variant-specific. Smart contracts are practical and inexpensive options for a limited range of IP addresses. However, major scalability concerns are caused on the Ethereum blockchain by keeping more than a few hundred IPs directly in the contract. Directly storing all IP addresses in an array is an expensive method. The so-called "solution," the bloom filter, not only failed to reduce costs but also introduced accuracy issues and made it impossible to access the complete database of IP addresses.

Outsourcing huge data to a proven protocol, such as a list of IP addresses on the web, is the most scalable method of the three. The immutability features of the blockchain can be applied to the web resource by utilizing a hash to verify the resource's integrity. In contrast, this version falls short of the potential of a blockchain-based solution and is the least ambitious of the solutions considered. In conclusion, DDoS signaling applications are not a good fit for Ethereum's general-purpose blockchain. Although most of the problems stem from a lack of scalability, the established methods work well enough for transmitting moderate quantities of data. Specialized blockchains that are more tailored for this type of application can be developed, which will help to improve the concept of decentralized DDoS signaling. The analysis provides valuable insights into the economic aspects of deploying smart contracts, offering a basis for informed decision-making in the development and optimization of blockchain-based system.

References

[1] Wani S, Imthiyas M, Almohamedh H, Alhamed KM, Almotairi S, Gulzar Y. Distributed denial of service (DDoS) mitigation using blockchain—A comprehensive insight. Symmetry. 2021; 13(2): 227.

[2] Abou El Houda Z, Hafid AS, Khoukhi L. Cochain-SC: An intra-and inter-domain DDoS mitigation scheme based on blockchain using SDN and smart contract. IEEE Access. 2019; 7: 98893.

[3] Abou El Houda Z, Hafid A, Khoukhi L. Co-IoT: A collaborative DDoS mitigation scheme in IoT environment based on blockchain using SDN. IEEE Global Communications Conference. 2019:1-6. https://doi.org/10.1109/GLOBECOM38437.2019.9013542

[4] Dheeraj J, Gurubharan S. DDoS mitigation using blockchain. International Journal of Research in Engineering, Science and Management. 2018; 1(10): 622.

[5] Manikumar DVVS, Maheswari BU. Blockchain based DDoS mitigation using machine learning techniques. 2020 Second International Conference on Inventive Research in Computing Applications, IEEE. 2020:794.

[6] Singh J, Behal S. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. Computer Science Review. 2020; 37: 100279. https://doi.org/10.1016/j.cosrev.2020.100279

[7] Habib G, Sharma S, Ibrahim S, Ahmad I, Qureshi S, Ishfaq M. Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing. Future Internet 2022; 14(11): 341. https://doi.org/10.3390/fi14110341

[8] Kirli D, Couraud B, Robu V, Salgado-Bravo M, Norbu S, Andoni M, Antonopoulos I, Negrete-Pincetic M, Flynn D, Kiprakis A. Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. Renewable and Sustainable Energy Reviews. 2022; 158: 112013. https://doi.org/10.1016/j.rser.2021.112013

[9] Pavlidis A, Dimolianis M, Giotis K, Anagnostou L, Kostopoulos N, Tsigkritis T, Kotinas I, Kalogeras D, Maglaris V. Orchestrating DDoS mitigation via blockchain-based network provider collaborations. The Knowledge Engineering Review. 2020; 35: E16. doi:10.1017/S0269888920000259

[10] Kim K, You Y, Park M, Lee K. Ddos mitigation: Decentralized cdn using private blockchain. 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN) IEEE. 2018:693. DOI:10.1109/ICUFN.2018.8436643

[11] Chaganti R. Bhushan B, Ravi V. The role of Blockchain in DDoS attacks mitigation: techniques, open challenges and future directions. arXiv:2202.03617. 2022. https://doi.org/10.48550/arXiv.2202.03617

[12] Hayat RF, Aurangzeb S, Aleem M, Srivastava G, Lin JCW. ML-DDoS: A Blockchain-Based Multilevel DDoS Mitigation Mechanism for IoT Environments. IEEE Transactions on Engineering Management. 2022:1-14. DOI: 10.1109/TEM.2022.3170519

[13] Yang X, Liu B, Yang F, Wang C. A blockchain based online trading system for DDoS mitigation services. 2018 IEEE Intl Conf on Parallel & Distributed

Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications. 2018:1036-1037. DOI:10.1109/BDCloud.2018.00151

[14] Essaid M, Kim D, Maeng SH, Park S, Ju HT. A collaborative DDoS mitigation solution based on Ethereum smart contract and RNN-LSTM. 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS) IEEE. 2019: 1-6.

[15] Anita N, Vijayalakshmi M. Blockchain security attack: a brief survey. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE. 2019: 1-6.

[16] Killer C, Rodrigues B, Stiller B. Security management and visualization in a blockchain-based collaborative defense. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE. 2019:108-111.

[17] Abou El Houda Z, Khoukhi L, Hafid A. Chainsecure-a scalable and proactive solution for protecting blockchain applications using sdn. 2018 IEEE Global Communications Conference (GLOBECOM), IEEE. 2018: 1-6.

[18] Gruhler A, Rodrigues B, Stiller B. A reputation scheme for a blockchain-based network cooperative defense. 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). 2019:71.

[19] Abou El Houda Z, Hafid A, Khoukhi L. BrainChain-A Machine learning Approach for protecting Blockchain applications using SDN. 2020 IEEE International Conference on Communications (ICC). 2020:1-6.

[20] Singh R, Tanwar S, Sharma TP. Utilization of blockchain for mitigating the distributed denial of service attacks. Security and Privacy. 2020; 3(3): e96.

[21] Hajizadeh M, Afraz N, Ruffini M, Bauschert T. Collaborative cyber attack defense in SDN networks using blockchain technology. 2020 6th IEEE Conference on Network Softwarization (NetSoft). 2020:487-492.

[22] Li D, Peng W, Deng W, Gai F. A blockchain-based authentication and security mechanism for IoT. In 2018 27th International Conference on Computer Communication and Networks (ICCCN). 2018:1-6.

[23] Al'aziz BAA, Sukarno P, Wardana AA. Blacklisted IP Distribution System to handle DDoS attacks on IPS Snort based on Blockchain. 2020 6th Information Technology International Seminar (ITIS). 2020:41-45.

[24] Rodrigues B, Eisenring L, Scheid E, Bocek T, Stiller B. Evaluating a blockchain-based cooperative defense. 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) 2019:533-538.

[25] Jiang S, Yang L, Gao X, Zhou Y, Feng T, Song Y, Liu K, Cheng, G. BSD-Guard: A Collaborative Blockchain-Based Approach for Detection and Mitigation of SDN-Targeted DDoS Attacks. Security and Communication Networks. 2022; 1608689.

[26] Kumar S, Amin R. Mitigating distributed denial of service attack: Blockchain and software ‑ defined networking based approach, network model with future research challenges. Security and Privacy. 2021; 4(4): e163.

[27] Rajan DM, Priya SS. DDoS mitigation techniques in IoT: A Survey. 2022 International Conference on IoT and Blockchain Technology (ICIBT), IEEE. 2022;1-7.

[28] Patel D, Patel D. Collaborative Blockchain Based Distributed Denial of Service Attack Mitigation Approach with IP Reputation System. International Conference on Database Systems for Advanced Applications, Springer, Cham. 2022: 91-103.

[29] Dutta A, Elie B. Outliers and Time-Varying Jumps in the Cryptocurrency Markets. Journal of Risk and Financial Management. 2022;15:128.

[30] Amrish R, Bavapriyan K, Gopinaath V, Jawahar A, Kumar CV. DDoS Detection using Machine Learning Techniques. Journal of IoT in Social, Mobile, Analytics, and Cloud. 2022; 4(1):24-32.

[31] Sajjad SM, Mufti MR, Yousaf M, Aslam W, Alshahrani R, Nemri N, Afzal H, Khan MA, Chen CM. Detection and Blockchain-Based Collaborative Mitigation of Internet of Things Botnets. Wireless Communications and Mobile Computing. 2022; 1194899.

[32] Rodrigues B, Stiller B. The Cooperative DDoS Signaling based on a Blockchain-based System. 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM). 2021: 760-765.

[33] Yeh LY, Huang JL, Yen TY, Hu JW. A collaborative DDoS defense platform based on blockchain technology. 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media), IEEE. 2019: 1-6.

[34] Ko I, Chambers D, Barrett E. Self-supervised network traffic management for DDoS mitigation within the ISP domain. Future Generation Computer Systems. 2020;112:524-533.

[35] Sundareswaran N, Sasirekha S. Packet Filtering Mechanism to Defend Against DDoS Attack in Blockchain Network. In Evolutionary Computing and Mobile Sustainable Networks, Springer, Singapore. 2022: 201-214.

[36] Rodrigues B, Scheid E, Killer C, Franco M, Stiller B. Blockchain signaling system (bloss): Cooperative signaling of distributed denial-of-service attacks. Journal of Network and Systems Management. 2020; 28(4): 953-989.

[37] Feng H, Yan X, Zhou N, Jiang Z, Liu Y. A Cross-domain Collaborative DDoS Defense Scheme Based on Blockchain-SDN in the IoT. In Proceedings of the 2021 ACM International Conference on Intelligent Computing and its Emerging Applications. 2021: 77-82. https://doi.org/10.1145/3491396.3506508

[38] Friha O, Ferrag MA, Shu L, Nafa M. A robust security framework based on blockchain and SDN for fog computing enabled agricultural Internet of Things. In 2020 International Conference on Internet of Things and Intelligent Applications (ITIA), IEEE. 2020: 1-5.

[39] Abou El Houda Z, Hafid A, Khoukhi L. Blockchain meets AMI: Towards secure advanced metering infrastructures. In ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE. 2020: 1-6.

[40] Sheikh MA, Khan GZ, Hussain FK. Systematic Analysis of DDoS Attacks in Blockchain. In 2022 24th International Conference on Advanced Communication Technology (ICACT), IEEE. 2022: 132-137.

[41] Raikwar M, Gligoroski D. DoS Attacks on Blockchain Ecosystem. arXiv preprint arXiv:2205.13322. 2022.

[42] Swami R, Dave M, Ranga V, Tripathi N, Shaji AK, Sharma A. Towards Utilizing Blockchain for Countering Distributed Denial-of-Service (DDoS). In Revolutionary Applications of Blockchain-Enabled Privacy and Access Control, IGI Global. 2021: 35.

[43] Yeh LY, Lu PJ, Huang SH, Huang JL. SOChain: A privacy-preserving DDoS data exchange service over soc consortium blockchain. IEEE Transactions on Engineering Management. 2020; 67(4):1487-1500.

[44] Shah Z, Ullah I, Li H, Levula A, Khurshid K. Blockchain Based Solutions to Mitigate Distributed Denial of Service (DDoS) Attacks in the Internet of Things (IoT): A Survey. Sensors, 2022; 22(3):1094.

[45] Franchina L, Carlomagno G. Correction to: A Comparison between SWIFT and Blockchain from a Cyber Resiliency Perspective. In International Conference on Critical Information Infrastructures Security, Springer, Cham. 2019: C1.

[46] Morganti G, Schiavone E, Bondavalli A. Risk assessment of blockchain technology. 2018 Eighth Latin-American Symposium on Dependable Computing (LADC), IEEE. 2018: 87-96.

[47] Imthiyas M, Wani S, Abdulghafor RAA, Ibrahim AA, Mohammad AH. DDoS mitigation: A review of content delivery network and its DDoS defence techniques. International Journal on Perceptive and Cognitive Computing. 2020; 6(2):67-76.

[48] Kataoka K, Gangwar S, Podili P. Trust list: Internet-wide and distributed IoT traffic management using blockchain and SDN. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), IEEE. 2018: 296-301.

[49] Poonia, L. ., & Tinker, S. . (2023). DDoS Mitigation by Software-Defined Network (SDN) in the Context of ICMP And SYP Approach. International Journal of Intelligent Systems and Applications in Engineering, 12(1), 173–182.