A Comprehensive Robustness Analysis of Storj DCS Under Coordinated DDoS Attack

Rohon Kundu*+, Christian Gehrmann†+ , Maria Kihl‡+
Department of Electrical and Information Technology
Lund University
Lund, Sweden
Email: *rohon.kundu@eit.lth.se, †christian.gehrmann@eit.lth.se, ‡maria.kihl@eit.lth.se

***Abstract*—Decentralized Cloud Storage (DCS) is considered to be the future for sustainable data storage within Web 3.0, in which we will move from a single cloud service provider to creating an ecosystem where anybody could be a cloud storage provider. Currently, the cloud storage market is highly dominated by centralized players like Amazon S3, Google Cloud, Box, etc. Decentralized projects like Storj, Filecoin, and Sia have seen rising popularity with the advent of Web 3.0 applications. At the same time, any blockchain network is susceptible to large-scale DDoS attacks. This work focuses on the Storj DCS, where we aimed to analyze the robustness of the system under the influence of a coordinated DDoS attack which can be carried out by an adversary or a group of adversaries taking down a set of storage nodes. The novelty of our work lies in threefold: First, we use statistical methods to mathematically model the content distribution as well as the loss of a file or a segment from the system. Our model captures both the cases where we have homogeneous and non-homogeneous nodes. Secondly, we develop a cost-analytic approach to perform a robustness analysis of the Storj system and implement the proposed model in MATLAB. Finally, we calculate the cost of a DDoS attack that the adversary has to incur in order to be successful with the attack. Also, we propose a set of better parametric choices for erasure piece distribution under which the system has proved to be more robust than the parametric values implemented in Storj DCS.**

***Index Terms*—DDoS Attack, Storj, Robustness, Decentralized Cloud Storage, Data Loss, Security**

I. Introduction

In 2022, the global data storage market size stands at a staggering USD 217.02 Billion [1]. The market cap is expected to increase to USD 247.32 Billion in 2023 and to USD 777.98 Billion by 2030, according to [1]. With such an increase in data, efficient and secured data storage solutions are in high demand. An increase in the demand for creating an ecosystem of Decentralized Cloud Storage (DCS) services has been noticed in the last few years. Peer-to-Peer protocols like IPFS [2], Filecoin [3] and, Storj [4] have seen an increasing demand both from the commercial as well as research aspect. With the recent development of Web 3.0, Non-Fungible Tokens and Metaverse have increased demand for secured and efficient cloud storage services that are compatible with Web 3.0 infrastructure. The metadata of NFTs needs to be stored in a secure way, as the loss of its metadata would render the NFT value less. Similarly, the Decentralized Applications (DAPPs) built over blockchain generate billions of metadata that need to be stored in a secure and efficient way.

With Web 3.0 becoming more mainstream, blockchain is revolutionizing the way we perceive cloud storage. The cloud storage industry witnessed a dynamic shift from centralized cloud storage facilities like Amazon S3 and Google Cloud to blockchain projects like Storj [4], Filecoin [3] and Sia [5] which provides decentralized cloud storage solution. The main advantage that the decentralized players enjoy over the existing centralized solutions in terms of reducing the cost of service and availability of multiple storage nodes that act as independent cloud storage facilities where the content is stored. Both Storj [4] and Sia [5] are *sharding*-based protocols that focus on enhancing security and privacy for the stored data by first fragmenting the file followed by encrypting and distributing them over the different nodes available globally. In this way, the host node could not derive any information about the stored file. On the other hand, Filecoin [3] which is built over InterPlanetary File System (IPFS) [2], [6], [7], [8] aims to create a Decentralized Storage Network (DSN) which can efficiently store and retrieve data at a hyper-competitive price.

With the increasing demand and popularity of Web 3.0 applications, it is equally important to address the security threats that could arise in various blockchain protocols. One of the most significant threats to any blockchain protocol is a Distributed Denial-of-Service (DDoS) attack. The decentralized nature of the blockchain makes them resistant to such an attack but not fully immune to it. DDoS attack on a blockchain network can happen due to multiple reasons [9]. Types of DDoS attack on blockchain network includes: a) *Transaction Flooding* b) *Poorly Designed Smart Contract* c) *Node Failure*. In this paper, we focused on the Storj protocol and studied its architecture in detail.

+This work is financially supported by the SMARTY project, funded by the Swedish Foundation for Strategic Research (SSF), grant RIT17-0035. Maria Kihl and Christian Gehrmann are part of the Excellence Center at Linköping-Lund on Information Technology (ELLIIT) Strategic Research Area, and the Wallenberg Artificial Intelligence (AI), Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

2690-5965/23/$31.00 ©2023 IEEE
DOI 10.1109/ICPADS60453.2023.00102

We considered a specific form of DDoS attack that can happen when an adversary controls a set of storage nodes or there is a group of adversaries that coordinate with each other to control a set of storage nodes. Storj is an Ethereum-based blockchain protocol and in this paper, we do not perform a DDoS attack on the blockchain but rather on the cloud storage through multiple node failures. In the past few years, we have seen an increasing trend of DDoS attacks on various blockchain networks [10] including Ethereum [11], Bitcoin [12], Solana [13], etc. We focused on the effect of such a large-scale coordinated DDoS attack on data storage and retrieval by analyzing the probability of loss of a file from the Storj system.

II. Main Contributions

The novelty of our work lies in the fact that we have developed a statistical method to mathematically model the content-distribution principle implemented in Storj. Erasure analysis for a file or a segment has been conducted for the Storj system. Our developed model holds good for real protocol. Next, we developed a cost-analytic approach to analyze the robustness of the Storj system under different parametric values for a coordinated DDoS attack. Finally, we propose a better parametric choice for erasure piece distribution by performing an optimality test under certain attack scenarios. No previous study has been conducted focusing on the data loss model in the advent of a DDoS attack through node failure on the Storj system. In this paper, we have addressed this specific vulnerability and have done a comprehensive robustness analysis of the Storj system when there is a coordinated DDoS attack.

III. Related Work

Kapusta et.al [14] is one of the earliest works which emphasizes how data fragmentation/sharding could help to maintain resilience along with data confidentiality while enhancing the scalability of the system. Erik et.al [15] did an elaborate comparative study of the different decentralized cloud storage protocols like Filecoin [3], Storj [4], and Sia [5] protocol. Each protocol was analyzed based on content distribution and basic building blocks. A broader aspect of future challenges that could arise in any decentralized cloud storage was discussed in this paper. Vimercati et.al [16] discussed the broader challenges in the aspect of privacy and security that could arise in any cloud-based services. When it comes to data storage providing confidentiality of the stored data and selective information sharing are the major concerns. Vimercati et.al [16] discussed the impact of *sharding* for Decentralized Cloud Storage to improve the integrity and confidentiality concerns for any cloud-based services. At the same time, securing data in a decentralized cloud service is a fundamental challenge as the data is distributed and stored by different node operators across various geographical locations. The work done by Bacis et.al [17], [18] addressed this challenge by proposing a solution where the resource owner would have control over the confidentiality of their own data and can also delete their data while relying on a decentralized framework.

Zhang et.al [19] focused on exploiting a frameup attack against the Storj network. It was one of the earliest works that focused on the security aspect of Storj. The authors showed that it is possible for an adversary to store unencrypted data which could be visible on the respective node operator's system. A fix to the mentioned vulnerability was also proposed in the paper. The former version of Storj contained this vulnerability, but later it was fixed in the current Storj V3 version. A more recent work by Figueiredo et al. [20] exploited a Denial-of-Service (DoS) vulnerability in the dev./test environment [21] of Storj. The authors focused on the single point of failure vulnerability caused by the *satellite* and proposed potential mitigation. The Storj team was informed of the vulnerability present in the dev./test environment, but it doesn't have any potential effect on the original Storj's production system. Another interesting line of work has been conducted by Yuefeng et.al [22] where the private storage auditing framework based on a reputation system that is implemented in Storj has been referred to as an open question in the decentralized paradigm. This is mostly due to the fact that such an auditing framework can lead to trust issues in a decentralized setup. In order to resolve the challenges posed by the private storage auditing framework, Su et.al [23] proposed a Decentralized Self-Auditing Scheme (DSAS) for decentralized cloud storage solutions.

Data modification attack or pollution attack in Distributed Storage Systems (DSS) has been explored by Rossano et.al [24], [25]. A generalized data erasure model was proposed by Rossano et.al [24] which calculates the probability of loss of a file under a pollution attack. In our work, the adversarial model is based on the content distribution used in Storj and we considered a specific form of DDoS attack that could arise due to node failure in a blockchain network when multiple nodes are controlled by an adversary or a coordinated group of adversaries.

IV. Content Distribution Model

In this section, we develop a statistical approach to create a mathematical model based on the content distribution principles used in Storj [4]. The aim of such a mathematical model is to undergo erasure analysis under different threat scenarios. Let N be the total number of *nodes* distributed globally. In order to create a model which closely represents real-world architecture it is important to consider both *unvetted* and *vetted nodes* [4]. At any given time let there be N_v-*vetted nodes* and N_u-*unvetted nodes* in the network. For the Storj protocol, each *node* should have a minimum of 500-GB of storage available and there is no upper limit on the amount of storage space that could be rented out. For our mathematical model, we consider that there are a total of T files stored in the network say $\{F_1, ..., F_T\}$.

The file F_j undergoes *segment*-wise uploading. Each *segment* s_i has k-erasure *pieces*, out of which d- *pieces* are required to reconstruct back the *segment* s_i. So at first the k-erasure *pieces* of a *segment* s_i are uploaded among the N-*nodes* which contains both *vetted* and *unvetted nodes*. So every time one *segment* of file F_j is to be uploaded, k-distinct *nodes* are chosen without replacement. In our mathematical model first, we consider the basic scenario where we treat all *nodes* equally. This means that every *node* has a similar level of trust and reputation when it comes to being selected for storing the erasure *pieces* of different *segments* of a file F_j. The basic model is used for the purpose of comparison with the non-homogeneous model and is not according to the principle of Storj. Next, we consider the non-homogeneous case in which we have two types of *nodes*: *vetted* and *unvetted*. *Vetted nodes* are considered to be more reputed than the *unvetted nodes* [4]. This is an extension of the basic model and the non-homogeneous model corresponds to the principle used in Storj. The content distribution will vary in such a case which will be discussed in the upcoming sections.

A. Basic Model: All storage nodes are homogeneous

For the first step of the model, we consider any one segment say s_i of file F_j and later extend the model for all the segments of file F_j. All N-nodes are considered to be homogeneous in terms of reliability. To store all the k-erasure *pieces* of s_i, k out of N nodes are chosen at random without replacement. Let $S_N = (n_1, ..., n_k)$ denote a sample of k-*nodes* where all the k-erasure *pieces* of s_i are stored. Let A denote the number of nodes controlled by the adversary. Then, given an equal probability that a node selected by the adversary is a node actually used to store a particular segment s_i, follows the distribution of selection without replacement, i.e. the hypergeometrical distribution. Furthermore, the segment will only be lost if only the adversary controls greater than or equal to $k - d + 1$- erasure *pieces* of the *segment* s_i. Let us denote E_{s_i} as the event when segment s_i is lost given A-nodes are malicious in the case of homogeneous nodes, where the probability that E_{s_i} occur be calculated by using the hypergeometric distribution of erasure pieces among the adversarial and non-adversarial nodes. Depending on the original size of F_j, each file undergoes segmentation into m-segments say $\{s_1, s_2, ..., s_m\}$. Similar to that of E_{s_i} one can define $E_{s_2}, E_{s_3}, ..., E_{s_m}$ as the events when the segments $s_2, s_3, ..., s_m$ are deleted respectively given the adversary is controlling A number of nodes. The content distribution for the file F_j in our model is based on the following assumptions which are in alignment with the Storj protocol:

- **Assumption 1:** The file F_j undergoes *segment*-wise uploading and each *segment* of the file F_j is uploaded independently. Each node stores erasure pieces corresponding to different segments but no two erasure pieces of a specific segment can be stored on the same node.
- **Assumption 2:** Each segment $s_i, \forall i \in [1, m]$ of file F_j has equal probability of being deleted when there are A adversarial nodes present. This is true by Assumption 1, of the content distribution. Therefore, let

$$P(E_{s_i}) = P(E_{s_2}) = \cdots = P(E_{s_m}) = p^*$$

- **Assumption 3:** Following the content distribution principle in Assumption 1, $E_{s_1}, E_{s_2}, ..., E_{s_m}$ are m-independent events.

Based on the above assumptions we will now compute the probability that the file F is lost when the adversary controls A nodes. Let us define E_F to be an event when the file F is lost under the condition that there are A-adversarial nodes. The overall probability of a loss of a file given these assumptions is then:

$$P\left(E_F\right) = 1 - (1 - p^*)^m \tag{1}$$

where m is the total number of segments of file F and p^* denotes the probability with which any one segment of file F can be deleted when there are A adversarial nodes. The attack model can further be extended for multiple files in the network. Denote by E_{F_X}, the event that a single file is lost. Then as a straightforward extension of the previous expressions and using similar assumptions of independent files. We get the probability of loss of a single file among T different files in the system, $P\left(E_{F_X}\right)$, then equals:

$$P\left(E_{F_X}\right) = 1 - \left[(1 - p^*)^{\sum_{i=1}^{T} m_i}\right] \tag{2}$$

where $M = \sum_{i=1}^{T} m_i$ is the total number of segments in the system.

B. Advanced Adversarial Model: When the nodes are non-homogeneous

Next, we consider the advanced model which directly corresponds to the content distribution principle used in Storj that has two types of nodes: $vetted$ and $unvetted$ [4] . Let there be N_v-*vetted* nodes and N_u-*unvetted* nodes in the network. For the segment s_i of file F_j there are k-erasure pieces $\{p_1, ..., p_k\}$. Out of the k-erasure *pieces* of *segment* s_1: k_v-erasure *pieces* goes to the vetted *nodes* in the network (i.e., N_v-nodes) and k_u-erasure *pieces* goes to the unvetted *nodes* in the network (i.e., N_u-nodes). At a given time t an adversary can control A-nodes. We fix the A-adversarial nodes and vary how the k-erasure pieces of segment s_i are stored among the N-nodes. Consider that an adversary controls $(N_v)_A$-vetted nodes and $(N_u)_A$-unvetted nodes. k_v-erasure pieces are distributed among the vetted adversarial nodes and vetted non-adversarial nodes. Similarly, the rest of k_u-erasure pieces are distributed among the unvetted adversarial nodes and unvetted non-adversarial nodes. Let X and Y be two stochastic variables

where $X = \{0, 1, ..., k_v\}$ and $Y = \{0, 1, ..., k_u\}$. X is the number of pieces that are distributed among the vetted adversarial nodes and Y is the number of pieces that are distributed among the unvetted adversarial nodes. Given similar reasoning as the probability analysis in Section IV-A, the probability distribution of X and Y are given by:

$$P[X = x] = \frac{\left(\begin{array}{c} (N_v)_A \\ X \end{array} \right) \left(\begin{array}{c} N_v - (N_v)_A \\ k_v - X \end{array} \right)}{\left(\begin{array}{c} N_v \\ k_v \end{array} \right)} \quad (3)$$

and

$$P[Y = y] = \frac{\left(\begin{array}{c} (N_u)_A \\ Y \end{array} \right) \left(\begin{array}{c} N_u - (N_u)_A \\ k_u - Y \end{array} \right)}{\left(\begin{array}{c} N_u \\ k_u \end{array} \right)} \quad (4)$$

The event E_{s_i} remains the same as before, which implies that the segment s_i of F_j is lost given that the attacker is controlling A number of nodes. The joint probability distribution is then given by (under the assumption of independent events):

$$P(E_{s_i})_* = P[X = x, Y = y] = P[X = x] \cdot P[Y = y] \quad (5)$$

The overall probability $P(E_{s_i})_*$, for that, the attacker will be able to delete a single segment, is the sum of all events where the attacker controls $k - d + 1$ or more nodes where the erasure pieces of segment s_i is stored. This is in turn the sum of all events where the attacker has access to z nodes, where $z \geq k - d + 1$ and where $x + y = z$, i.e. where x is the number of vetted nodes controlled by the attacker added with y, which represent a number of unvetted nodes controlled by the attacker equals z. By then calculating this probability for all possible values of $z \geq k - d + 1$, we get the total probability that the attacker deletes the storage of the segment s_i. Given the previous distribution functions, (3) and (4) and by defining $P[X = x] = 0, x > k_v$ and $P[Y = y] = 0, y > k_u$, this is then equal to the following sum:

$$P(E_{s_i})_* = \sum_{i=k-d+1}^{k} \sum_{j=max(i-k_u,0)}^{i} P[X = j] \cdot P[Y = i - j] \quad (6)$$

Similar to 2, we can have

$$P\left(E_{F_X}\right)_* = 1 - \left[(1 - P(E_{s_i})_*)^M\right] \quad (7)$$

V. Experimental Setup

We have implemented the adversarial model in MATLAB and evaluated the probability of loss of a file using a cost analytic approach. In this section, we describe the evaluation procedure for the Storj system under different scenarios.

A. Adversary

The attacker under consideration can be an individual or state-sponsored adversary with a coordinated group of individuals whose aim is to cause a DDoS attack on the Storj network by taking down a set of A-nodes. The adversaries could either be present at a specific geographical location or coordinate with each other by being located at different geographical locations. We can assume that any storage node in the network has equal likeliness to come under adversarial control, this is because Storj treats all the vetted nodes equally to that of the unvetted nodes. This justifies our advanced adversarial model where any node has an equal probability of becoming an adversarial node, which directly correspond to the real DDoS threat in Storj.

B. DDoS Attack Cost Analysis

In order for an adversary to perform a DDoS attack it has to spend a certain amount of resources in the form of computational power which we term as the *cost-of-attack*. Let C be the total cost of attack and c_v, c_u be the cost incurred by the adversary/adversaries to control one vetted and one unvetted node respectively. The total cost of attack C can be incurred by an adversary alone or can be distributed among a group of coordinated adversaries. If there are N_0 number of adversaries in a group then $\sum_{i=1}^{N_0} C_i = C$, where C_i is the amount of resource spent by each of the adversaries present in the group. The goal of an adversary is to maximize the probability of loss of a file F_j by controlling an optimal number of vetted and unvetted nodes subjected to a given cost. The cost expression for the DDoS attack is therefore given by:

$$C = c_v \cdot (N_v)_A + c_u \cdot (N_u)_A \quad (8)$$

where $(N_v)_A$ and $(N_u)_A$ are the number of vetted and unvetted adversarial nodes that an attacker can control as constrained to the fixed cost C. If we put $(N_u)_A = 0$ in 8, then $max((N_v)_A) = \frac{C}{c_v}$. This means that with the specific values of (C, c_v, c_u) an adversary can control a maximum of $\frac{C}{c_v}$- vetted nodes. Similarly, when $(N_v)_A = 0$ then an adversary can control a maximum of $\frac{C}{c_u}$- unvetted nodes. In our cost equation the value of c_v is always greater than c_u. For an adversary to control a vetted node, it has to spend more resources than controlling an unvetted node since each of the vetted nodes goes through an audit process that takes around 30 days [4].

C. Experiment

We set up an experiment to evaluate the impact of a coordinated DDoS attack on the Storj network. The impact is quantified in terms of the numerical values that $P(E_{s_i})_*$ and $P\left(E_{F_X}\right)_*$ takes at every point $\{(N_v)_A, (N_u)_A\}$. We have implemented the expression for the probability of loss of a file mentioned in Section IV-B. For a given set of values of (C, c_v, c_u) the MATLAB program finds all the feasible integer solutions i.e., $\{(N_v)_A, (N_u)_A\}$ for

the linear equation 8 and the probability of loss of a file $P\left(E_{F_X}\right)_*$ is evaluated at the feasible integer solution points. For all our plots the X- axis contains the possible values of vetted adversarial nodes $(N_v)_A$ subjected to the given cost C and the Y-axis contain $P\left(E_{F_X}\right)_*$ values.

D. Parametric Values

In our experiments, we implement the parametric values obtained from the real-time Storj Network statistics [26]. As of 11^{th}-September 2023, when the experiments are conducted there are around 25500- vetted nodes in the network, and 308-unvetted nodes. Also, the total amount of stored customer data amounts to 22.3 Petabyte, and the average size of a segment is 9.39-MB. This leaves us with approximately a total of 2.375×10^9 segments in the Storj network. We consider that the cost of controlling a vetted node (c_v) is 50 times that of the unvetted ones. It is not possible to get a precise figure for the difference in cost for an attacker to take a vetted node compared to an unvetted node [27], but here for the purpose of simplicity, we use the factor 50 in our runs which gives a fair description of the significant difference in cost of controlling an unvetted nodes compare to that of a vetted one. The vetting process takes 30-days according to [4]. For the base case presented in the Section-VI, we implement $N_v = 25500, N_u = 308, c_v = 50, c_u = 1$. Regarding the parametric values of k_v and k_u, our simulations are based on the Storj erasure piece distribution parameters where 95% of the ingress traffic goes to the vetted nodes and 5% to the unvetted nodes [28]. At worst around 7.5% of erasure pieces for a segment can end up on the unvetted nodes [28]. Therefore we vary the k_v value in our simulations from 74 to 77 and k_u from 6 to 3 respectively in order to investigate how $P(E_{F_X})_*$ vary around the points for Storj distribution parameter. Also, the minimum threshold value for successful retrieval (d) is set to 29 as mentioned in [4].

VI. Results

In this section, we exploit the adversarial model that we have developed in Section IV-B for non-homogeneous nodes to access the overall performance of the Storj network under different attack scenarios. All our evaluations are based on the parametric values mentioned in Section V-D. We evaluate the extent of a coordinated DDoS attack when the adversary with a fixed resource aims to delete a file from the system. We consider the coordinated DDoS attack to be successful if for given parametric choices $P\left(E_{F_X}\right)_* \geq 0.5$, which would imply that for every second time, the attacker would succeed. Following eq. 7, one can calculate the lower bound on the probability of loss of a segment in case of a successful attack, which is 0.29178×10^{-9}. Even if the adversary can achieve a probability of loss of a segment value as small as in the order of 10^{-9}, then $P\left(E_{F_X}\right)_* \geq 0.5$ and the attack will be successful. On the other hand, we consider the system to be robust if for given parametric values $P\left(E_{F_X}\right)_* \leq 0.001$, which would mean that the adversary has to initiate at least 1000 attack trial to succeed with the DDoS attack. Similarly, following eq 7 , one can find the upper bound on the probability of loss of a segment in the case of a robust system, which is 4.213×10^{-13}. If the probability of loss of segment surpasses this value then we cannot term the Storj system to be robust. Our investigation of the robustness of the Storj network is mainly divided into three parts:

1) Under the assumption of Storj system parameters with respect to $(k_v, k_u) = \{(74,6),(75,5),(76,4),(77,3)\}$ and $d = 29$ what is the cost budget required for an attacker with high likelihood to succeed with an attack i.e., achieve $P\left(E_{F_X}\right)_* \geq 0.5$?
2) For the given Storj parameters and the attack budget found in (1) is the erasure piece distribution values (k_v, k_u) an optimal choice for all scenarios?
3) For a given cost budget, by how much does one need to increase the number of vetted or/and unvetted nodes in the network to achieve a robust system i.e., $P\left(E_{F_X}\right)_* \leq 0.001$?

For each of the above three subcases, we investigate the probability of loss of a file with the increase in the number of vetted adversarial nodes subjected to a fixed cost.

A. Questions to be investigated:

*1) **A.1. Under the assumption of Storj system parameters with respect to** $(k_v, k_u) = \{(74,6),(75,5),(76,4),(77,3)\}$ **and d = 29 what is the cost budget required for an attacker with high likelihood to succeed with an attack i.e., achieve** $P\left(E_{F_X}\right)_* \geq 0.5$**?:*** Fig 1 is considered as the base case as we experiment with the real-time parametric values i.e., $N_v = 25500$ and $N_u = 308$ as mentioned in [26]. Each of the plots is evaluated at four distinct values for k_v, k_u i.e, $(k_v, k_u) = \{(74,6),(75,5),(76,4),(77,3)\}$. Storj minimum threshold parameter $(d = 29)$ is used in this case and there is approximately a total of 2.375×10^9-segments in the network.

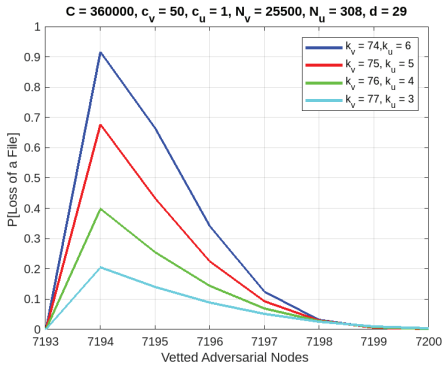

Fig. 1. Part-1: Cost Budget $C = 360000$

Conclusion:

The goal of Fig.1 is to find the attack cost budget C with which the adversary would be able to achieve $P\left(E_{F_X}\right)_* \geq 0.5$, i.e., to be able to successfully carry out the DDoS attack. We found that $C = 360000$ could be one such suitable cost budget that fulfills our requirement. From Fig.1, we can conclude that for $(k_v, k_u) = \{(74,6),(75,5)\}$, the $max|P\left(E_{F_X}\right)_*| > 0.5$. On the otherhand, we notice that for $(k_v, k_u) = \{(76,4),(77,3)\}$, the $max|P\left(E_{F_X}\right)_*| < 0.5$. Therefore, looking from the attacker's perspective for $(k_v, k_u) = \{(74,6),(75,5)\}$ the attack succeeds at the optimal points corresponding to $X = 7194$ where the probability of loss of a file attains its maximum values. Therefore for $(k_v, k_u) = \{(74,6),(75,5)\}$ the attacker with $C = 360000$ would aim to control 7194-vetted nodes and 300-unvetted nodes to achieve the maximum probability of loss of a file. The attacker would have a lower probability for successfully carrying out a DDoS attack when $(k_v, k_u) = \{(76,4),(77,3)\}$. From Fig.1 we can conclude that the Storj system is vulnerable to a DDoS attack for all choices of (k_v, k_u), but $(k_v, k_u) = \{(76,4),(77,3)\}$ results in the lower probability of successful attack in comparison to $(k_v, k_u) = \{(74,6),(75,5)\}$. Therefore in order to opt for better system security in the case of Fig.1 one should choose $(k_v, k_u) = \{(76,4),(77,3)\}$. With a difference of a factor of 50 between controlling a vetted node and an unvetted one, it is not an optimal strategy to put more pieces on the unvetted nodes as this would result in a higher value of $max|P\left(E_{F_X}\right)_*|$ as shown in Fig.1. We notice a decrease in the value of $P\left(E_{F_X}\right)_*$ beyond the optimal points as there is less budget available to control the unvetted nodes.

2) ***A.2 For the given Storj parameters and the attack budget found in A.1 is the erasure piece distribution values*** (k_v, k_u) ***an optimal choice for all scenarios?*** *:* In Fig.2, we evaluate the probability of a successful DDoS attack by varying $(k_v, k_u) = \{(74,6),(75,5),(76,4),(77,3),(78,2),(79,1),(80,0)\}$. Fig.2, resembles the network scenario as shown in [26] where there are very few unvetted nodes. We keep the number of vetted nodes constant at $N_v = 25500$ and the total budget of attack to be 360000. In Fig.3 we increase the number of unvetted nodes to 50000 in order to analyze the optimal distribution of (k_v, k_u) when there are a large number of unvetted nodes in the network.

Conclusion: The goal of Fig.2 is to analyze whether $(k_v, k_u) = (76,4)$ is an optimal choice when there a very few unvetted nodes in the network. We can conclude that from Fig.2, given the current number of very few unvetted nodes, it is not the best strategy to distribute real erasure pieces to the unvetted node at all. Rather one could adopt a different approach to check the reliability of the new unvetted nodes with test erasure pieces which does not correspond to a real segment of a file uploaded by any user. For Fig.3, where there are 50000-unvetted nodes, we can conclude that $(k_v, k_u) = (77,3)$ is an optimal choice as it has the least $max|P\left(E_{F_X}\right)_*|$ value. When there are a large number of unvetted nodes in the network, it is a good strategy to place a few i.e., $k_u = 3$ erasure pieces on the unvetted nodes which will result in a lower probability of a successful DDoS attack.

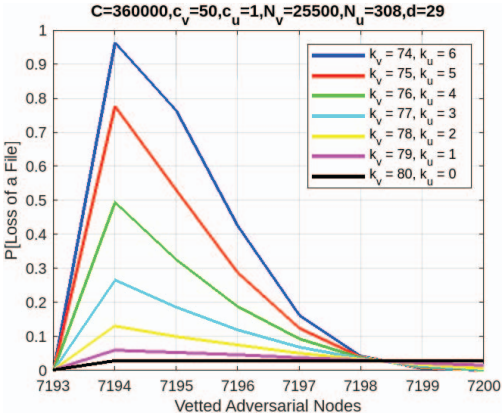

Fig. 2. Varying (k_v, k_u) for $N_u = 308$

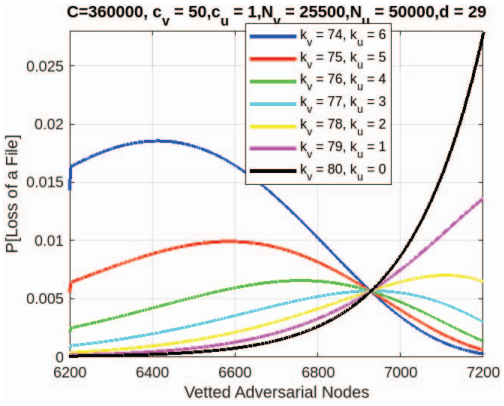

Fig. 3. Varying (k_v, k_u) for $N_u = 50000$

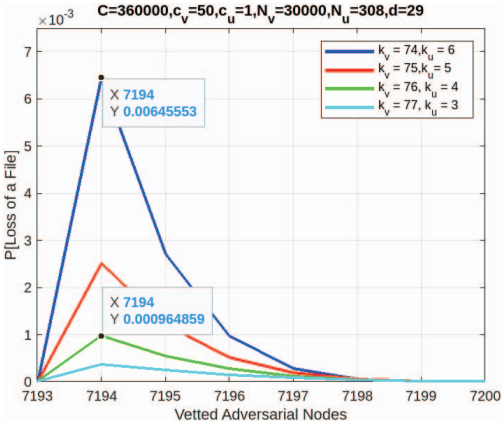

Fig. 4. Vetted nodes increased to 30000

3) ***A.3 For a given cost budget, by how much does one need to increase the number of vetted or/and unvetted nodes in the network to achieve a robust system i.e.,*** $P\left(E_{F_X}\right)_* \leq 0.001$***?:*** The goal of Fig. 4 and Fig. 5, is to find the number of vetted nodes in the network keeping the unvetted nodes fixed for which we get a robust system i.e., $P\left(E_{F_X}\right)_* \leq 0.001$. First, we increase the number of vetted nodes to 30000 in Fig.4 and to 31525 in Fig.5 respectively, while keeping the number of unvetted constant at $N_u = 308$. A total cost budget of $C = 360000$ is used and each of the plots is further evaluated at four distinct values for k_v, k_u i.e, $(k_v, k_u) = \{(74,6),(75,5),(76,4),(77,3)\}$.

Whereas, the goal of Fig.6 and Fig.7, is to find how many unvetted nodes there should be in the network keeping the vetted nodes fixed for which we get a robust system i.e., $P\left(E_{F_X}\right)_* \leq 0.001$. First, we increase the number of unvetted nodes to 100000 in Fig.6 and to 110000 in Fig.7 respectively, while keeping the number of vetted nodes constant at $N_u = 25500$. A total cost budget of $C = 360000$ is used and each of the plots is further evaluated at four distinct values for k_v, k_u i.e, $(k_v, k_u) = \{(74,6),(75,5),(76,4),(77,3)\}$.

Conclusion: From Fig.4, for $N_v = 30000$ we can conclude that the Storj system is robust when $(k_v, k_u) = \{(76,4),(77,3)\}$ as $max|P\left(E_{F_X}\right)_*| < 0.001$ at the optimal points corresponding to $X = 7194$. The system is not robust for $(k_v, k_u) = \{(74,6),(75,5)\}$ as $max|P\left(E_{F_X}\right)_*| > 0.001$ at the optimal points. Fig.5, depicts a scenario where there are 31525 vetted nodes and the Storj system proves to be robust for all choices of (k_v, k_u) as $max|P\left(E_{F_X}\right)_*| < 0.001$. An opposing observation can be made from Fig.6, and 7 regarding the optimal choice of (k_v, k_u) while achieving robustness only through increasing the number of unvetted nodes keeping the vetted nodes constant at 25500. From Fig.6, for $N_u = 100000$ we can conclude that the Storj system is robust when $(k_v, k_u) = \{(74,6),(75,5)\}$ as $max|P\left(E_{F_X}\right)_*| < 0.001$ at the optimal points. The system is not robust for $(k_v, k_u) = \{(76,4),(77,3)\}$ as $max|P\left(E_{F_X}\right)_*| > 0.001$ at the optimal points. We can make a similar conclusion from Fig.7 where the number of unvetted nodes is further increased to 110000. For the scenarios illustrated in Fig.6 and 7, we can conclude that it is difficult to achieve a robust system for the Storj erasure piece distribution value i.e., $(k_v, k_u) = (76,4)$. But, if we place a few more erasure pieces on the unvetted nodes i.e., $k_u = 6$ and $k_u = 5$ as shown in Fig.6 and 7, it is feasible to achieve a robust system just by increasing the number of unvetted nodes in the network while keeping the number of vetted nodes constant.

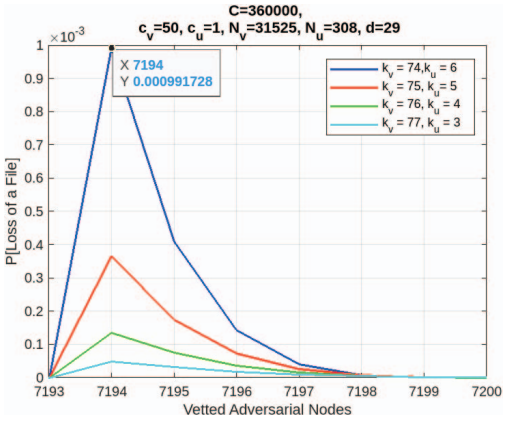

Fig. 5. Vetted nodes increased to 31525

VII. Conclusion

With the rapid emergence of blockchain-based applications, it is of real concern on how to avert the impact of large-scale DDoS attacks that can be carried out through node failure. In this paper, we analyzed the impact of a coordinated DDoS attack that can be carried out by state-sponsored adversaries on the Storj Decentralized Cloud Storage (DCS) platform. The impact is quantified in terms

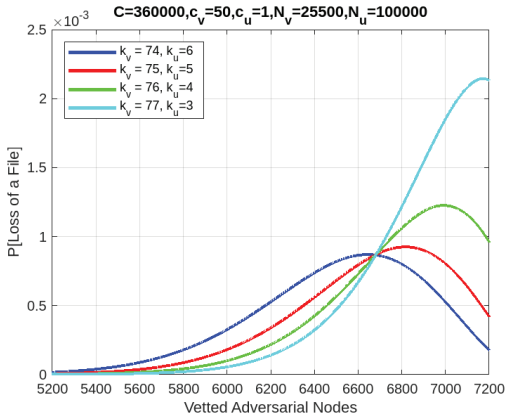

Fig. 6. Unvetted nodes increased to 100000

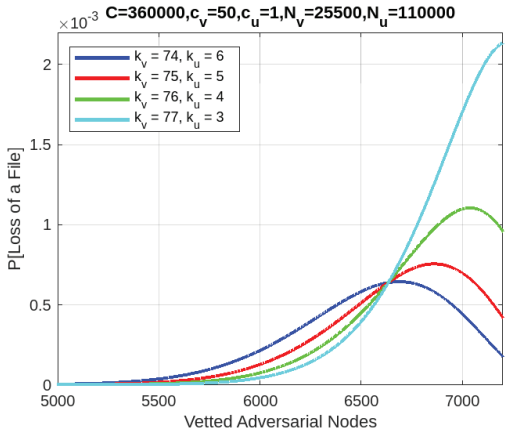

Fig. 7. Unvetted nodes increased to 110000

of the probability of successful deletion of a file from the system. We implemented the proposed statistical model for data loss in MATLAB to perform robustness analysis. Our evaluations are based on an adversarial model, where an adversary would aim to find the optimal way to distribute its resources to control a certain number of vetted and unvetted nodes. The conclusion from this work can be summarized in three folds: First, we calculated the cost budget that the adversary has to incur for a successful DDoS attack on the Storj system with a high probability of success. For $C = 360000$, the adversary could control approximately 28% of the total vetted nodes i.e., 7194-vetted nodes out of 25500 available vetted nodes, and achieve a high probability of successful DDoS attack. Secondly, we showed the effect of varying $(k_v, k_u) = \{(74, 6), (75, 5), (76, 4), (77, 3), (78, 2), (79, 1), (80, 0)\}$ on the probability of a successful DDoS attack while keeping the vetted nodes constant at 25500. We conclude that for a current scenario where there are a few unvetted nodes in the network, it is not an optimal strategy to place any real erasure piece on the unvetted nodes but rather adopt an approach where we could place pieces corresponding to a test file on the unvetted nodes. We also checked what could be an optimal choice of erasure distribution when there are 50000-unvetted nodes in the network. Our analysis showed that it is a better strategy to place a few pieces on the unvetted nodes under such a scenario to achieve a lower probability of a successful DDoS attack. Finally, for a DDoS attack scenario where the adversary has $C = 360000$, it is possible to make the system robust i.e. $P\left(E_{F_X}\right)_* \leq 0.001$, either by increasing the number of the vetted nodes to approximately 20% or by increasing the unvetted nodes in the network to at least 100000. Also, we showed that when there are 25500 vetted nodes in the network and 100000 unvetted nodes it is beneficial to put a few more erasure pieces on the unvetted nodes as it makes the system robust. For this specific scenario, it is difficult to achieve a robustness condition for the Storj erasure distribution parameter i.e., $(k_v, k_u) = (76, 4)$. All our simulations are done with a difference in the cost factor of 50 between the vetted and unvetted nodes. Nevertheless, our proposed model could used to analyze any scenario with different cost factors and similar conclusions can be drawn.

REFERENCES

[1] GlobeNewswire, https://www.globenewswire.com/news-release/2023/02/28/2616751/0/en/With-17-8-CAGR-Data-Storage-Market-Size-Worth-USD-777-98-Billion-by-2030.html, 2023.

[2] T. V. Doan, Y. Psaras, J. Ott, and V. Bajpai, "Toward decentralized cloud storage with ipfs: Opportunities, challenges, and future considerations," *IEEE Internet Computing*, vol. 26, pp. 7–15, 2022.

[3] P. Labs and J. Benet, "Filecoin: A decentralized storage network," Protocol Labs, Tech. Rep., 2017.

[4] I. Storj Lab, "Storj: A Decentralized Cloud Storage Network ;https://www.storj.io/storjv3.pdf," Tech. Rep., 2018.

[5] D. Vorick and L. Champine, "Sia: Simple decentralized storage," *Retrieved May*, vol. 8, p. 2018, 2014.

[6] L. Balduf, S. A. Henningsen, M. Florian, S. Rust, and B. Scheuermann, "Monitoring data requests in decentralized data storage systems: A case study of ipfs," *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pp. 658–668, 2021.

[7] C. Karapapas, G. Polyzos, and C. Patsakis, "What's inside a node? malicious ipfs nodes under the magnifying glass," *ArXiv*, vol. abs/2306.05541, 2023.

[8] E. Daniel and F. Tschorsch, "Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks," *IEEE Communications Surveys & Tutorials*, vol. 24, pp. 31–52, 2021.

[9] BlockchainSecurity, https://www.halborn.com/blog/post/how-blockchain-ddos-attacks-work, 2021.

[10] B. Magazine, https://blockchainmagazine.net/what-are-ddos-attack-top-10-ddos-attacks-on-a-blockchain/#: :text=Here

[11] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[12] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[13] A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0. 8.13," *Whitepaper*, 2018.

[14] K. Kapusta and G. Memmi, "Data protection by means of fragmentation in distributed storage systems," *2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*, pp. 1–8, 2015.

[15] E. Daniel and F. Tschorsch, "Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 31–52, 2022.

[16] S. D. C. di Vimercati, S. Foresti, and P. Samarati, "Protecting data and queries in cloud-based scenarios," *SN Computer Science*, vol. 4, 2023.

[17] E. Bacis, S. D. C. di Vimercati, S. Foresti, S. Paraboschi, M. Rosa, and P. Samarati, "Securing resources in decentralized cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 286–298, 2019.

[18] E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa, and P. Samarati, "Dynamic allocation for resource protection in decentralized cloud storage," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[19] X. Zhang, J. Grannis, I. Baggili, and N. L. Beebe, "Frameup: An incriminatory attack on storj: A peer to peer blockchain enabled distributed storage system," *Digital Investigation*, vol. 29, pp. 28–42, 2019.

[20] S. de Figueiredo, A. Madhusudan, V. Reniers, S. Nikova, and B. Preneel, "Exploring the storj network: a security analysis," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021, pp. 257–264.

[21] E. Egon, "Storj test network," https://github.com/storj/storj/wiki/Test-network, 2022.

[22] Y. Du, H. Duan, A. Zhou, C. Wang, M. H. Au, and Q. Wang, "Enabling secure and efficient decentralized storage auditing with blockchain," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[23] Y. Su, Y. Li, B. Yang, and Y. Ding, "Decentralized self-auditing scheme with errors localization for multi-cloud storage," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, pp. 2838–2850, 2022.

[24] R. Gaeta, "On the impact of pollution attacks on coding-based distributed storage systems," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 292–302, 2022.

[25] R. Gaeta and M. Grangetto, "Malicious node identification in coded distributed storage systems under pollution attacks," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 6, no. 3, pp. 1–27, 2021.

[26] S. Forum, https://storjstats.info/d/storj/storj-network-statistics?orgId=1, 2023.

[27] F. Discussion, https://forum.storj.io/t/cost-of-running-nodes/14232, 2023.

[28] S. D. Forum, https://forum.storj.io/t/distribution-of-erasure-pieces-of-one-segment/21082, 2023.