An Implementation of Fake News Prevention by Blockchain and Entropy-based Incentive Mechanism

Chien-Chih Chen
University of Waterloo
Waterloo, ON, Canada
j2255che@uwaterloo.ca

Yuxuan Du
University of Waterloo
Waterloo, ON, Canada
d9du@uwaterloo.ca

Richards Peter
University of Waterloo
Waterloo, ON, Canada
r2peter@uwaterloo.ca

Wojciech Golab
University of Waterloo
Waterloo, ON, Canada
wgolab@uwaterloo.ca

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671778

***Abstract*—Fake news is undoubtedly a significant threat to democratic countries nowadays because existing technologies can quickly and massively produce fake videos, articles, or social media messages based on the rapid development of artificial intelligence and deep learning. Therefore, human assistance is critical if current automatic fake new identification technologies desire to improve accuracy. Given this situation, prior research has proposed to add a quorum, a group of appraisers trusted by users to verify the authenticity of the information, to the fake news prevention systems. This paper proposes a stake-based incentive mechanism to diminish the negative effect of malicious behaviors on a quorum-based fake news prevention system. Moreover, we use Hyperledger Fabric, Schnorr signatures, and human appraisers to implement a practical prototype of a quorum-based fake news prevention system. Then we conduct necessary case analyses and experiments to realize how dishonest participants, crash failures, and scale impact our system. The outcomes of the case analyses and experiments show that our mechanisms are feasible and provide an analytical basis for developing fake news prevention systems.**

***Index Terms*—Authenticity, False information prevention, Malicious behaviors, Reward and punishment, Social media**

I. Introduction

In recent years, the influence of fake news has increased, and it has caused great harm to democracy, the press, and freedom of speech. For example, fake news significantly influenced the result in the 2016 U.S. presidential election and the Brexit referendum [1]. Many countries currently enact laws requiring technology companies, including Facebook and Google, to prevent fake news on their platforms [9]. Therefore, technology companies are actively combating fake news to respond to the compliance requirements from governments. For example, Facebook cooperates with news experts to identify fake news on its platform [2]. Google also has actively established policies to prevent the spread of fake news on its products [3]. Even the IEEE [10] is aware of the seriousness of fake news proliferation and has formulated a guideline to advocate human welfare.

However, due to the platforms' social media characteristics, such as Facebook, Twitter, and Instagram, the rapid delivery of information makes it challenging to prevent fake news. Modern users receive countless messages every day, so it is difficult to confirm each news item's authenticity. Moreover, fake news might come from authoritative news websites as well, such as CNN, BCC or the Wall street journal. In addition, with the development of data technology and AI, experts can quickly analyze Internet users' habits and then manipulate the public to achieve their desired results by circulating fake news [11]. Therefore, it is arduous to rely solely on AI and other technologies to identify fake news automatically.

Jaroucheh, Alissa, Buchanan, and Liu (2020) proposed a conceptual software framework that adopts blockchain and digital signature for verifying online content that involves human intervention in reviewing fake news [6]. With the help of digital signature and blockchain technology, their framework calculates online content's trust score based on an appraiser list that a user sets and the level of trust in appraisers. However, the trust score of [6] is generated by the setting of trust levels from users to appraisers. This approach lacks objectivity, so the trust score will lose its correspondence to the authenticity of the news. For instance, the trust score of the same news verified by the same group of appraisers will lead to different trust scores due to different trust levels that users set. Besides, each appraiser has various abilities to judge whether a piece of content is real or fake to result in a case in point is that an appraiser remarkably trusted by a user always produces incorrect identification results of fake news. In [6], there is no mechanism designed to avoid this situation. Instead, our system allows users to define their own trusted objects. By endorsing trusted objects, users can use the trust score calculated by credit points and the confidence of appraisers to determine the authenticity of the news. In this way, this system can effectively curb the spread of fake news. The main contribution of this article is that we complete the following improvements for the concept that was offered by [6]:

a) Offering proof of stake entropy-based incentive mechanism to diminish effects of collusion from malicious nodes: In this paper, we design a novel incentive mechanism on top of the proof of stake [18] concept to decrease the possibility of generating false information from content creators. Besides, our incentive mechanism could encourage content appraisers to contribute their effort to inspect content to provide trust scores by their credit points and confidence of a specific

Authors supported in part by Ripple and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

appraisal opinion they offer. Moreover, we adopt entropy [19] to calculate rewards and punishments of appraiser results to adjust the stakes and credit points of each appraiser. Depending on the incentive mechanism we propose, the trust score has enormous credibility and possesses the capability to eradicate ineligible appraisers. Instead, the concept in [6] is vulnerable to malicious attackers taking over a majority of verifying tasks to decrease the credibility of its trust score.

b) Providing an actual and complete implementation: We not only use Hyperledger Fabric [7] to implement the blockchain network but also apply the Schnorr signature [8] to verify whether the human appraisers approve the content sent by content producers. In contrast, [6] did not state the detail of implementing the blockchain network. In brief, our implementation transfer the concept of [6] to a real-world application. With the immutability and traceability that blockchains provide, the content creators could utilize our system to prove the authenticity of their content. Users also could retrieve signatures to realize whether they could trust target news.

c) Proposing a feasible crash failure handling and scalability mechanism: It is indispensable to a distributed system that it works without any disturbance when any node in its network fails. On the other hand, scalability is an essential requisite of a distributed system as well. Therefore, in this work, we design a mechanism to prevent crash failure with scalability. However, in [6], the authors only suggested high-level architecture and did not discuss their crash failure handling and scalability mechanism.

d) Demonstrating abundant case analyses and experiments results to facilitate future research: We provide various case analyses and experimental evaluations on incentive mechanism, crash failure prevention, and scalability of our system, while [6] did not mention any experimental data. The outcomes of analyses and experiments offer an analytical benchmark for future quorum-based fake news prevention systems.

Concerning the remainder structure of this paper, we will explain related works and background to which we refer in Section II and Section III. Afterward, our methodology and implementation will be described in Section IV, including architecture, incentive mechanism, data access layer, and business logic layer. Then the explanations of case analyses and experiments are in Section V. Finally, the conclusion is stated in Section VI.

II. Related Works

Nowadays, many content-sharing platforms, such as Medium or Reddit, allow users to edit and share all kinds of content freely. However, the audit-free content publishing mechanism can effortlessly become a platform for spreading fake news because no one, even administrators of the content-sharing forum, takes the initiative to verify content [6]. In addition, many social media platforms, such as Twitter, Facebook, or Google, have possibly become hotbeds for fake messages because users can publish content at will in accordance with the social media norms. Unfortunately, the social platforms do not offer reliable enough verification mechanisms for the authenticity of each content as well [2] [3] [6].

According to the major international economic and political crises in recent years, it is indisputable that fake news threatens national security [1] [6]. Thus, governments of various countries have formulated policies to curb the unhealthy trend of producing or spreading false information. In order to prevent the spread of fake news on content sharing or community platforms, national policies tend to require platforms to be responsible for the content. Platforms are responsible for verifying the authenticity of the content, and actions must be taken to ensure that the platform's artificial intelligence exhibits the correct behavior [6]. Although these policies are well-intentioned, they still lack the correctness of involving users in fake news verification. For instance, Facebook has its internal authenticity verification algorithm for the content, and it tries to flag fake news. However, the logic of distinguishing false news is debatable because the users do not participate in the algorithm design, and the fake news identified by the platform does not necessarily conform to the user's definition of fake news [12].

In response to government regulations, technology companies have been working hard to improve platform content quality and increase the speed and accuracy of detecting fake news [6]. For example, Google mentioned in its white paper that artificial intelligence would be used to improve the accuracy of the algorithm for detecting fake news [3]. To enhance the quality of content, Facebook hires authorities in various fields to review content and actively build an ecosystem that is comprised of professional content production organizations and individuals [2]. Nevertheless, these efforts are not enough. As [6] mentioned manual mechanisms need to be added to the artificial intelligence fake news detection system to remove blind spots of artificial intelligence. Because of recent developments in artificial intelligence and deep learning, only using algorithms to check for fake news automatically is difficult. It is unreliable and impractical to rely entirely on algorithms to prevent fake news because there are always blind spots in artificial intelligence [6].

The root cause of blind spot in artificial intelligence is that if planning to build an artificial intelligence-based fake news detection model, we must analyze false news attributes [6]. For example, Sirajudeen, Fatihah, Adamu, and Abubakar (2017) proposed a set of algorithms that can analyze the network packets of the content to collect different types of false news [4]. Another solution was provided by "Reality Defender" to analyze fake news patterns in content [5].

However, artificial intelligence models mainly adapt voice recognition, image recognition, or machine learning algorithms to detect various false news. As we mentioned, those algorithms possess inherent technical limitations to hinder them from ideally screening incorrect information. In other words, the correctness of detecting fake news of artificial models relies on massive data that people could collect. Nevertheless, not each time could we gather enough samples of what we plan to observe. Consequently, the technical limitations of underlying technologies of artificial intelligence are why the accuracy of automatic false news detection is always lower than that of manual detection [6].

Hence, as [6] mentioned, manual verification is divided into centralized (expert-based) and decentralized (crowd-sourced). Centralized is performed in platforms like PolitiFact [13] or HoaxSlayer [14]. Those platforms have a group of experts who have the background to verify the authenticity of the content. Still, as mentioned about the controversial internal censorship mechanism of Facebook [12], experts' judgment to detect fake news differs from that of the users. The decentralized method distributes the content to multiple individuals for content review, but this is back to a state of no control mechanism. Furthermore, it is impossible to confirm the reviewer's educational background, whether it is fair without any partiality, and whether he/she has malicious motives [6].

[6] also allows user to define their appraising quorum, a group of verifiers, to assist users in verifying the authenticity of the content. Although one appraiser might have blind spots according to its professionalism, the probability of misjudging news could be dwindled based on the opinion of crowds. In addition, being decentralized, immutability and traceability, it is adequate for blockchains to be utilized to form an anti-counterfeiting mechanism. Accordingly, [6] recommends keeping appraising results of fake news via blockchains.

III. BACKGROUND

Hyperledger Fabric is one of the projects under the Linux foundation's Hyperledger umbrella. It is one of the popular private and permissioned Distributed Ledger Technologies (DLT) that exist today. Hyperledger Fabric [7] states that it was designed for solving several enterprise use cases that required permissioned networks, manageable identities, high transaction throughput, and low latency of transaction confirmation while preserving privacy and confidentiality requirements of transactions and data associated with the transactions. Relying on its design, Hyperledger Fabric does not require a native currency such as Bitcoin to perform its normal operations, and it eliminates the cryptographic mining process, which is usually performed on permissionless blockchains.

On the other hand, Hyperledger Fabric also possesses powerful strength to support allowing developers to write Smart Contracts (also called Chaincode) in various programming languages using the corresponding Software Development Kit (SDK) and then deploy Smart Contracts on the network. These Smart Contracts can be invoked using the command line interface of Hyperledger Fabric or through the client code developed using the SDK. In its internal design, Hyperledger Fabric lets organizations, also known as *member*, form consortiums on a high level, and these consortiums are responsible for owning and maintaining the Distributed Ledger within their network. Further, Hyperledger Fabric allows organizations to deploy smart contracts to perform transactions within the network without a central authority. These smart contracts are packaged into a chaincode in Hyperledger Fabric [7].

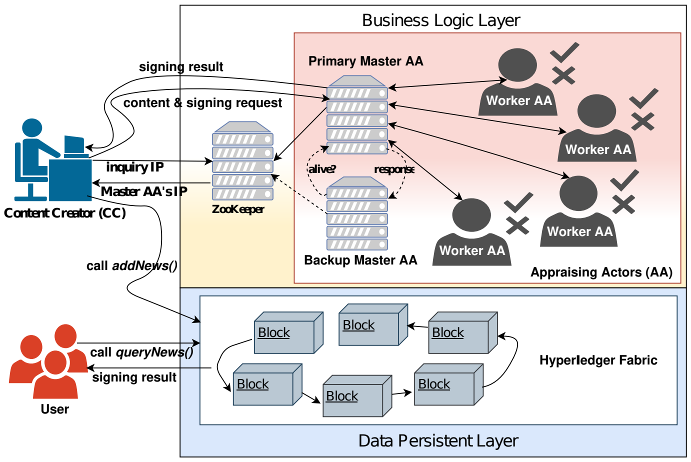

Fig. 1. One of the content creators (CCs) and users. Two layers: Business Logic Layer and Data Persistent Layer that contain their components

Each participating organization can execute smart contracts implemented by chaincodes and perform transactions with the distributed ledger.

IV. METHODOLOGY AND IMPLEMENTATION

A. Architecture

As the first contribution we mentioned in Section I, we use Hyperledger Fabric as our underlying Blockchain network infrastructure that [6] did not state in their work. Moreover, for enhancing the performance, we use the Schnorr signature instead of the two-round trip collective signature that takes more time to finish the signing process of which [6] adopted. Besides, as shown in Fig. 1, we divided [6] 's architecture into two main layers, Data Persistent Layer and Business Logic Layer, which we will describe in this section. According to this paper's case analysis and experiment results, we believe our system could fully support the most consequential function that a quorum-based fake news prevention system requires: providing objective trust scores to content. Next, we will introduce the implementation detail and main system flow of two critical parts: the Business Logic Layer and Data Persistent Layer in our quorum-based fake news prevention system.

1) Business Logic Layer: Client applications: As the name implies, the business logic layer (BLL) includes the logic for fake news verification in this layer. In this section, we will describe all of the components in the BLL.

a) Content Creator (CC): CC is responsible for creating news content, broadcasting the content to AA and user nodes, obtaining the Schnorr signature, and at last sending a request to the blockchain network to write the content hash and collective signature to Hyperledger Fabric. Besides, **CC must pay the rewards** that our system incentive appraisers as a fee for asking appraising its content. The approach of calculating rewards and punishments will be introduced in Section IV.

b) Appraising Actors (AA): In our system, the quorum is comprised of appraising nodes with two Master AA nodes and four Worker AA nodes. In the real world, appraising behaviors should be executed by humans. However, the primary purpose of this paper is to build a prototype to prove a quorum-based fake news prevention system. Thus, we simulate human behaviors via assigning different appraising results to AAs.

Master AA's responsibilities are making asynchronous RPC calls to worker AA nodes when CC invokes them. Then wait for all the worker AA's feedback to sign the content and send the result back to CC. On the other hand, Worker AA receives hash content sent by CC, reviewing content, and signing after offering their opinion. Our system simulates behaviors that human appraisers review content and send their opinion on approving or rejecting content.

As shown in Fig. 1, there is one primary Master AA and one backup master AA in the BLL. A master AA that is executed first will create its znode to put its IP address and port number in the znode, and then set itself as the primary Master AA node in the ZooKeeper, like the solid arrow line from the primary Master node to the ZooKeeper in Fig. 1. Then worker AAs and the CC could get the primary Master AA node's IP address and port number to start communicating by sending an inquiry to the ZooKeeper. The double-arrow lines between the primary Master AA and worker AAs in Fig. 1 represent that the worker AAs already know the primary Master AA's IP address and port number, so the primary Master AA and worker AAs could begin to send RPC calls to each other.

Furthermore, we let the backup master AA send an RPC call to verify whether the primary Master AA is still alive or not every 50 ms. Based on a timeout time: 100 ms, once the backup master AA cannot receive the response from the primary Master AA, the backup master AA will consider the primary Master AA is already crashed, and then the backup master AA will take over the responsibility of dispatching signing tasks to worker AAs. The double-arrow dot lines in Fig. 1 between the ZooKeeper, primary Master AA, and backup master AA describe how the backup master AA set itself to the primary Master AA. At first, the master AA that is executed when the other master AA already exists will be the backup master AA in our system. Moreover, the backup master AA will only build its znode under the same path as the primary Master AA rather than setting it as the primary Master AA in the ZooKeeper. After getting the primary Master AA's IP address and port number, the backup master AA will regularly check whether the primary Master AA is alive or not. Once the primary Master AA fails, the backup master AA will set it as the primary Master AA.

c) User: Receives the content from CC and uses the hash value of the content as a querying parameter to retrieve the digital signature stored in the blockchain. After obtaining the signature concatenation string from the blockchain, the user node will interpret the signature concatenation string to which worker AAs agree with the content. Moreover, users could also realize each appraiser's confidence, credit points, trust scores in their decision regarding the authenticity of the content from the signature concatenating string. In Section IV. B, we will explain confidence, credit points, and trust scores more thoroughly.

All the CC, AA, blockchain, and user nodes comprise our system. Each CC, AA, and blockchain node is located on different servers but in the same private network. The experiment result we will introduce in Section V illustrates that our design could prevent crash failure and possess scalability for multiple appraisers. Moreover, providing that proof of stake incentive mechanism that we propose in Section IV. B of this paper, we could raise the motivations of CC and AAs to contribute their best on conducting their tasks with benign behaviors.

2) Data Persistent Layer: Hyperledger Fabric: We named it a Data Persistent Layer (DPL) because it receives data and provides functionalities to record data on the Hyperledger Fabric network. Also, DPL provides APIs for users to query specific signatures via the content's hash. The DPL's APIs that are implemented by smart contract will be introduced in Section IV D.

B. Proof of Stake Entropy-Based Incentive Mechanism

Based on the concept of proof of stake [18], any entity who desires to become a content creator (CC) or appraiser actor (AA) in our fake new prevention system must hold a certain amount of digital tokens as their stakes. At present, we did not integrate any specific digital token in our system because our priority goal in this paper is to build a pilot system to apply an entropy-based incentive mechanism to a quorum-based fake new prevention system. However, any digital token, like bitcoin or ETH, could be integrated into our quorum-based fake news prevention system for paying rewards or punishments for each appraising outcome.

We define ***credit points*** to represent the reputation of CC and AA in our system. Anyone who desires to be a content creator (CC) or an appraiser (AA) must pay a certain amount of digital token to get their initial credit points. According to the core spirit of proof of stake, credit points could motivate that each CC or AA will not be a malicious node because the more they spend, the higher their credit points. Moreover, we design a reward and punishment process to augment the willingness of CC and AA to contribute to the quorum-based fake new prevention system with ethical behaviors. Fig. 2 shows the main flow of calculating and adjusting credit points of CCs and AAs. Besides, we also define ***trust score*** that is calculated by credit points of CCs and AAs to indicate how many percentages that a piece of news is authentic or fake. Now, we will introduce the proof of stake entropy-based incentive mechanism that we propose in Fig. 2.

1) Stake Raising Process: Stake raising process is that our system sets a period to let CCs or AAs increase their credit points, which is calculated by stakes, via paying digital tokens by themselves. Another case to initiate the stake raising process is CCs or AAs could also request a new round of stake raising process whenever their credit points are equal to zero. ***Appraisers***, also called AAs, are the ones who approve or decline the digital content. In our system, we treat CCs as one type of appraiser. However, CC only could support its content

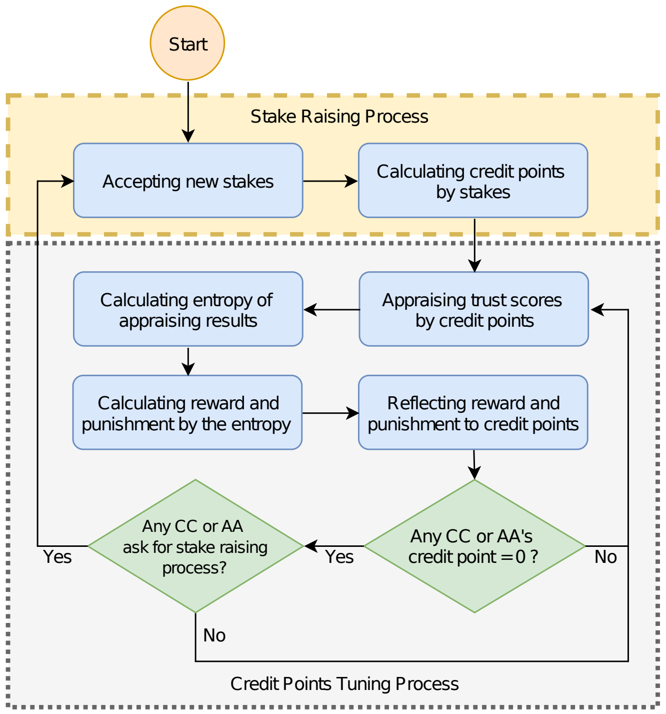
Start
Stake Raising Process
Accepting new stakes
Calculating credit points by stakes
Calculating entropy of appraising results
Appraising trust scores by credit points
Calculating reward and punishment by the entropy
Reflecting reward and punishment to credit points
Any CC or AA ask for stake raising process?
Any CC or AA's credit point = 0 ?
Yes
Yes
No
No
Credit Points Tuning Process

Fig. 2. Main flow of the proof of stake entropy-based incentive mechanism

because they should be fully confident that the content they generate is one hundred percent authentic. Although CC may also be a malicious node and send fake content to AAs and users, our proof of stake entropy-based incentive mechanism could combat malicious CCs through severe punishments for dishonest behavior in our system.

Next, we define the ***credit point of appraiser (CPoA)*** in (1), where *n* is the total number of entities that desire to increase credit points, *i* is the number of an individual entity, $1 \leq i \leq n$, and S_i is the digital tokens that an $entity_i$ would like spend to be an appraiser. Moreover, the stake an entity could spend will be limited by an upper bound and a lower bound that our system sets.

$$CPoA_i = \frac{S_i}{\sum_{i=1}^{n} S_i} \tag{1}$$

Example 1: Assume that there are five entities, $entity_1$ would like to be a CC in our system as well as $entity_2$ to $entity_5$ plan to join our system to be AAs by holding stakes that our system recognizes. Their stakes are 5000, 1000, 2000, 10000, and 3000 relatively. Therefore, we could calculate their initial ***CPoA*** are 0.24, 0.05, 0.10, 0.48, and 0.14 respectively, where we round results of (1) to two decimal digits.

2) Credit Points Tuning Process: Our system utilizes ***trust score*** that is calculated by credit points of CC and AA to indicate whether a piece of news is fake or not to users. The credit point of each CC or AA will be increased or diminished based on the outcomes of their jobs: CCs will get rewards if they produce authentic content based on evaluations from AAs. Similarly, AAs will also receive rewards if the side, fake or authentic, they support has a more significant trust score than the other side. Having introduced an overview of the credit point tuning process, we will now discuss each step that the credit point tuning process contains.

a) Appraising trust scores by credit points: We calculate two different trust scores in our system. First, we define the ***Score of authentic (SoA)*** in (2), which means how much possibility that a content is real, where *c* means the content that is currently appraised by AAs, a_i is an individual appraiser including the CC, ***Conf***$_i$, $0 < Conf_i \leq 1$, represents how much confidence the appraiser has in its judgement. Nevertheless, each appraiser **only could put their *Conf*$_i$ on supporting either authentic or fake rather than endorsing both sides.** Besides, although being one type of appraiser, the CC could only give 1.0 to its *Conf* because, as a content creator, the CC should trust its content one hundred percent. Moreover, A_{app} is the group of appraisers who approve specific information, and *n* is the total number of appraisers in A_{app}. If one appraiser considers a piece of news is authentic, the appraiser should approve it, or appraisers could reject content as long as they believe they receive false information.

$$SoA(c) = \sum_{i=1}^{n} CPoA_i \times Conf_i, \forall a_i \in A_{app} \tag{2}$$

The second type of trust score we define, on the other hand, is the ***Score of Fake (SoF)*** in (3). *SoF* represents how much possibility that a content is fake, where a_j means an individual appraiser including the CC as well, A_{rej} is the group of appraisers who reject specific information, and *n* is the total number of appraisers in A_{rej}. Besides, like *SoA* of (2), the *c* in (3) represents a particular content that is being appraised.

$$SoF(c) = \sum_{j=1}^{n} CPoA_j \times Conf_j, \forall a_j \in A_{rej} \tag{3}$$

Example 2: The credit point of the CC is 0.24, and the others four appraisers' credit points are: 0.05, 0.10, 0.48, and 0.14 according to Example 1. A case in point is that if the CC produces a content (c_1) and then sends it to the four appraisers that we mentioned in Example 1. Suppose that a_1 and a_3 approve c_1 with $Conf_{a1}$ = 0.7 and $Conf_{a3}$ = 0.8 respectively. On the contrary, a_2 and a_4 reject c_1 on the basis of $Conf_{a2}$ = 0.8 and $Conf_{a4}$ = 0.7. We could obtain the ***SoA*** and ***SoF*** of c_1 that is rounded to second decimal place based on (2) and (3):

$$SoA(c_1) = 0.24 \times 1.0 + 0.05 \times 0.7 + 0.48 \times 0.8 = 0.66$$
$$SoF(c_1) = 0.10 \times 0.8 + 0.14 \times 0.7 = 0.18$$

When users query trust scores from the blockchain, they will realize that the c_1 has been appraised on 0.66 real and 0.18 fake instead, like Example 2. To sum up, the appraising result: *SoA* and *SoF* play a pivotal role in calculating rewards or punishments to adjust credit points of CCs and AAs in our system. What follows is an account of how our system tunes credit points by appraising results.

b) Calculating entropy of appraising results: Nowadays, Shannon entropy [19], which is defined in (4) [20], is one of the most widely used methodologies to measure disorder [22].

$$Entropy(S) = -\sum_{i=1}^{k} p_i \log_2 p_i \tag{4}$$

In (4), we define S is a set that contains the appraising results of all appraisers. Also, k = 2 because there are only two classes of appraising results, either ***Authentic*** or ***Fake***, in our system. Moreover, p_i means the fraction of appraising results in class *i*. In particular, we define p_1 is the proportion of appraising results that are equal to ***Authentic*** in *S* while p_2 is the proportion of appraising results that are equal to ***fake*** in *S*. Consequently, *Entropy(S)* = 0 if all appraiser in our system have the same opinion to consider a certain content is real or fake. In this case, *Entropy(S)* = 0 means either p_1 = 1 or p_2 = 1. On the other hand, *Entropy(S)* = 1 if both p_1 and p_2 are equal to 0.5, which means the viewpoints on authenticity of a content are thoroughly divergent.

Example 3: According to Example 2, we know that CC, a_1 and a_3 approve the c_1. Conversely, the a_2 and a_4 reject the c_1. Based on the distribution of appraising values, we could use (4) to figure out the p_1 = 0.6 and the p_2 = 0.4. Hence, the *Entropy(S)* = 0.97. Note that we round *Entropy(S)* to the second decimal place.

c) Calculating reward and punishment using entropy: In our system, we compare to the *SoA* and *SoF*, and then if *SoA* $>$ *SoF*, our system will give reward to appraisers who approve a certain content (c_1) and punish other appraisers that reject c_1 and vice versa. Furthermore, the CC would be punished if its content has been appraised as false information

It is generally complicated to design a mechanism to decide a certain number of rewards or punishments in a blockchain. Our calculation approach of reward and punishment is inspired by the Ethereum [21]. We set ***Basic Reward (BR)*** of each content to ***total stakes*** $\times$ ***0.1%***. On the other hand, the ***Basic Punishment (BP)*** would be ***total stakes*** $\times$ ***10%***, which is higher than ***BR*** because we desire to raise the cost of producing malicious behaviors. Further, our design offers appraisers receive more rewards and punishment if the *Entropy(S)* is smaller because lower *Entropy(S)* means the opinions of appraisers are highly consistent. In a nutshell, we use *Entropy(S)* to design our entropy-based incentive mechanism to reward the presumptively correct appraisers and punish the wrong ones. The appraisers who are presumptively correct are the ones who vote for the winning standpoint of higher scores between *SoA* and *SoF*.

In light of this thought, we define our ***Reward of Content (RoC)*** and ***Punishment of Content (PoC)*** in (5). Based on our incentive mechanism, if *SoA* $>$ *SoF*, appraisers who vote to agree the content will receive the *RoC*, whereas the *PoC* will locate to appraisers who approve contents when *SoF* $>$ *SoA*. An extreme case that is worthy to mention is that when the *SoA*(c_1) $=$ *SoF*(c_1), users still could get the trust scores but appraisers will not receive any rewards or punishments. Although we set the ***BR*** and ***BP*** as a fixed ratio of total stakes, in fact, the ***BR*** and ***BP*** could be calculated dynamically based on the ratio between the total stakes of the numbers of appraisers like Ethereum to maintain reasonable ***BR*** and ***BP***.

$$\begin{aligned} RoC &= (1 - Entropy(S)) \times BR \\ PoC &= (1 - Entropy(S)) \times BP \end{aligned} \tag{5}$$

Example 4: Based on Example 1, we know that total stakes is 21,000. Thus, the ***BR*** would be 21,000 $\times$ 0.1% and the ***BP*** would be 21,000 $\times$ 10%. Besides, we know the *Entropy(S)* = 0.97 according to Example 3 as well. As a result. We could get the ***RoC*** = 0.63 and ***PoC*** = 63 that are rounded from the result of (5) to the second decimal places. Also for the c_1 in Example 2, the *SoA*(c_1) $>$ *SoF*(c_1). Therefore, our system will reward appraisers who approve c_1 and punish appraises who reject c_1.

d) Reflecting reward and punishment to credit points: After getting the rewards and punishment of a certain content, our system will add or reduce credit points of appraisers based on how much percentage they contributed the *SoA* or *SoF*. Hence, we define the ***Ratio of SoA (RSoA)*** and ***Ratio of SoF (RSoF)*** in (6). Note the ***RSoA*** and ***RSoF*** are rounded to the second decimal place.

$$\begin{aligned} RSoA_i &= \frac{CPoA_i \times Conf_i}{SoA}, \forall a_i \in A_{app} \\ RSoF_j &= \frac{CPoF_j \times Conf_j}{SoF}, \forall a_j \in A_{rej} \end{aligned} \tag{6}$$

For instance, in Example 2, we know that the *SoA*(c_1) = 0.66 and the *CPoA*$_1 \times$*Conf*$_1$ = 0.24. As a result, the $RSoA_1$ would be 0.36 (rounded to second decimal). Following (6), we could calculate all ***RSoA*** and ***RSoF*** to all appraisers. Thus, for c_1 in Example 4, we will reward stakes of 0.23, 0.03, and 0.37 (***RoC*** $\times$ ***RSoA***$_i$) to *CC*, a_1 and a_3 respectively, but slash 27.72 and 34.02 (***PoC*** $\times$ ***RSoF***$_j$) from the stakes of a_2 and a_4 relatively as punishments. Besides, we rounded (***RoC***$\times$***RSoA***$_i$) and (***PoC*** $\times$ ***RSoF***$_j$) to second decimal place as well. Finally, we recalculate credit points according to modified stakes after each round of appraisers. The updated credit point in this example would be: CC = 0.24, a_1 = 0.05, a_2 = 0.09, a_3 = 0.48 and a_4 = 0.14.

e) Inspecting whether any credit point of CC or AA equal to zero: After recalculating credit points based on updated stakes after each round of appeasement, we will inspect whether an appraiser has zero credit points. Our system will notify appraisers whose credit points are equal to zero. Otherwise, the status of our system will return to accept another round of appraising.

f) Checking whether any CC or AA ask for a new stake raising process: Any entity that desires to be an appraiser must have a non-zero credit point. So, appraisers whose credit point equals zero could ask for another stake raising process to increase their stakes to earn extra credit points, or they would not be allowed to be appraisers.

Until now, we have already introduced the entropy-based incentive mechanism that we proposed in this paper. Furthermore, our system will let each appraiser use the Schnorr signature to encrypt its decision and then send its encrypted appraising result directly to the blockchain to affirm that the individual appraiser will conduct its appraising task independently. When the CC finishes adding the news to the blockchain, our system will decrypt the appraising results. Each CC must add the news to the blockchain when its content has already been appraised, or the CC will get the ***BP***. This regulation could prevent a situation where CCs, for deceiving purposes to earn ***BR***, only save content that was appraised authentic.

C. Implementation of BLL: CC, AA and User

We present our implementation of Client Applications: CC, AA, and User nodes in the following paragraphs.

1) CC–Create contents: In a real-world scenario, the content should be an article, an image, or a video piece. However, any article, video, or image is encoded into a binary string to be transferred to another endpoint. Therefore, we generated content with fixed characters. Another advantage of this approach is that it is easier to control the content size, which is an essential parameter in the system throughput testing. To get the accurate size of the content we generated, we transform the string to a bytes array, and the size of the bytes array is the actual size of the content. A unique id and hash value are generated for each piece of content and are sent out along with the content to the primary Master AA node.

2) CC–Initiating the signing process: The CC plays the role of initiating the signing process and storing signatures from AAs to the blockchain. The CC exerts RPC calls (Thrift) to send a content(message) to AAs and receives signatures from AAs. As we mentioned before, our internal architecture of AA nodes' cluster comprises two master AA nodes; one is the primary Master AA, the other is Backup Master AA, and four worker AA nodes (see Fig. 1). We adopt the ZooKeeper, which is a centralized node for maintaining configuration information, coordinating services, and providing distributed synchronization [15], to achieve the fault tolerance mechanism of AA nodes.

As shown in Fig. 1, the CC will contact the ZooKeeper to get the available primary Master AA node's IP address and port. After getting the primary Master AA nodes' connection information, the CC will send an RPC call to trigger the primary Master AA to issue asynchronous RPC calls to worker AA nodes responsible for signing tasks. Worker AA nodes will sign the content passed from master AA nodes and return the result upon receiving the RPC call. On top of that, for maintaining a healthy connection between Worker AA nodes to the primary Master AA node, we let Worker AA nodes send a heartbeat to ZooKeeper to get the address of an existing primary Master AA node. The double arrow that connects worker AAs to the primary Master AA represents that the master AA sends RPC calls to assign a signing task to worker AAs, and the worker AAs reply the result after completing sign tasks.

3) AA–Executing signing process: With the help of the Schnorr algorithm, worker AA nodes sign the content and return the signature. Along with the signature, a flag parameter is also passed, which indicates whether the worker AA nodes endorse the content or not. After the master AA node receives all the signatures, it concatenates signatures and sends them back to the client.

4) CC–Storing content to Hyperledger Fabric: As mentioned before, we adopt the Chaincode to design two smart contracts to achieve the DPL's functionalities. These two smart contracts are *addNews()* and *queryNews()*. As shown in Fig. 1, the CC will call *addNews()* function to store the id, content, and signature concatenation string to the blockchain after the CC gets the signatures from worker AAs.

5) User–querying content from Hyperledger Fabric: Similarly, The other smart contract: *queryNews()* is used by the user node to query a specific message's signing result. In Fig. 1, we describe the flow of how the user node gets the signature result. The user node calls *queryNews()* to inquiry the blockchain by sending the hash value of a message. Then the user node could retrieve a signature concatenation string if the message exists in the blockchain. Afterward, the user will parse the signature concatenation string and examine which worker AAs approved the content.

D. Implementation of DPL: HyperLedger Fabric node

We mainly refer to the configuration suggestions from the HyperLedger Fabric [7] to configure our blockchain network. This section will introduce the implementation details of the Hyperledger Fabric network used in this work in three main points.

Firstly, we deployed the Hyperledger Fabric network on only one server. Our Hyperledger Fabric contains two Peer Organizations and an Orderer Organization. Besides, each organization has its own Certificate Authority. Moreover, Peer Organizations have two users – Admin and User. On the other hand, Orderer Organization has a single Orderer and only one user: Admin.

Secondly, we use Fabric-CA Server in order to generate the crypto materials. The network used in this work spawns three Fabric-CA Server Docker containers, one for each peer organization and one for the orderer organization. After starting the Fabric-CA Server containers, the network enrolls a CA Admin using Fabric-CA Client and then registers and enrolls every entity using that CA Admin. Fabric-CA Server will generate self-signed certificates and also a secret key. Once the crypto materials are generated, they are moved to the entities' appropriate directory structure for use. Finally, this work also uses an application channel to communicate between peer and orderer organizations.

Lastly, concerning the Chaincode design, the transaction that needs to be recorded in the distributed ledger includes the content created by the Content Creator and the signature collected from the Appraisal Actors by the Content Creator. Although we use Java to create objects to support our application's smart contract, we believe our main workflow in Fig. 1 could be implemented in any other programming language supported by the Hyperledger Fabric. [7].

V. Cases Analysis and Experiments

Our main contributions are to design an entropy-based incentive mechanism and conduct various experiments to verify the feasibility of a quorum-based fake new prevention system. This section first analyzes our entropy-based incentive mechanism to demonstrate we could effectively resist the 51% and Sybil attacks. Then we discuss the experiment results of crash failure prevention and scalability of our system. Our analysis and experiment results assure that our work's incentive mechanism and system architecture are robust and feasible. In other words, this paper provides the essential basis for strategic planning for further development in quorum-based fake news prevention systems.

A. Case Analysis of Proposed Incentive Mechanism

It is essential to a quorum-based to possess an effective incentive mechanism to give impetus to appraisers for completing content evaluations reliably. Our proof of stake entropy-based incentive mechanism raises punishment to dishonest CCs or AAs to deter malicious behaviors, which represents appraisers who offer an opposite opinion of the authenticity of news in our system. Table I presents all possible scenarios of our incentive mechanism in Fig. 2. We define two primary types of cases in our system. The first one is ***Regular Cases*** that means the *majority of AAs (Majority)*, appraisers who vote for the winning standpoint of higher scores between SoA and SoF, identify the authenticity of content correctly, such as the cases that we marked in gray in Table I. In contrast, the *minority of AAs (Minority)* means the appraisers that hold a different opinion to the *Majority*. The second type is the cases that have no background color that represents the ***Exotic Cases*** which indicate the *Majority* fail to judge the authenticity of a piece of news. Note that the CC does not be counted in the *Majority*, although the CC is one type of appraiser in our system. Furthermore, the ***N/A*** in Table I states that all appraisers have the same viewpoints of appraising. Next, we will dive into discussion of ***Regular Cases*** and ***Exotic Cases***.

TABLE I
Reward and Punishment in different scenarios

Case No.	Viewpoints of Content			Reward (+) or Punish (-)		
	CC	*Majority*	*Minority*	*CC*	*Majority*	*Minority*
1	REAL	REAL	N/A	+	+	N/A
2	REAL	REAL	FAKE	+	+	-
3	REAL	FAKE	N/A	-	+	N/A
4	REAL	FAKE	REAL	-	+	-
5	FAKE	REAL	N/A	-	+	N/A
6	FAKE	REAL	FAKE	-	+	-
7	FAKE	FAKE	N/A	-	+	N/A
8	FAKE	FAKE	REAL	-	+	-

1) Regular Cases: As all the cases that possess a gray background in Table I indicate, our incentive mechanism could motivate appraisers to correctly specify the authenticity of information because our system punishes dishonest appraising and rewarding reliable entities. On the other hand, each appraiser would enhance their ability to judge the authenticity of news because of the high cost of judging news falsely. Two notable examples are case seven and case eight that the CC provides fake news, but our system could identify fake news successfully based on viewpoints of the *Majority*.

2) Exotic Cases: A critical point is shown by exotic cases that only malicious nodes could be rewarded, whereas other benign appraisers are punished. The root cause of exotic cases is that several AAs unite together to commit deception, but these exotic cases will hardly happen in our system. Our incentive mechanism facilitates that it is almost impossible for a hacker to control more than 51% of stakeholders in our system, much less to collect all stakes in our system. It is also arduous even to collect more than 51% stakes in our system. Also, our system could resist the Sybil attack because it would be a considerable cost if one entity desires to possess a significant impact of trust scores in our system. However, one possibility of our system, like all other incentive-based blockchain systems that have just been initiated, is that if the amount of stake that we hold is tiny, our system would be vulnerable to 51% attacks. Besides, by calculating the *PoC* via entropy in (5), our system can know the degree of association of malicious users and adjust the amount of punishment. In other words, malicious users will face higher penalties when they consider collaborating with other users to engage in deceptive behavior. To sum up, according to the analysis result, our incentive mechanism indeed owns the ability to motivate the CCs to provide dependable news and induce AAs to contrive to judge fake news honestly.

B. Sensitive analysis of the proof of stake entropy-based incentive mechanism

The most significant outcomes in our incentive mechanism are: *trust scores:* ***SoA*** *or* ***SoF***, *reward:* ***RoC*** and *punishment:* ***PoC***. In this section, we will conduct two sensitive analyses to realize variations of variables to trust scores, rewards and punishments.

1) Confidence and credit points to trust scores: In Fig. 3, we set an experiment that only has one CC and one AA. Therefore, the summation of credit points of CC and AA would be 1.0, and we could calculate *SoA* or *SoF* by (2) or (3) in Fig. 3. rely on different pairs of credit points and the confidence of the AA. An obvious pattern is shown in Fig 3. That is, if AA's credit point is getting higher, the variation of its confidence will affect the *SoA*/*SoF* more. This phenomenon is reasonable because any entity that owns tremendous credit

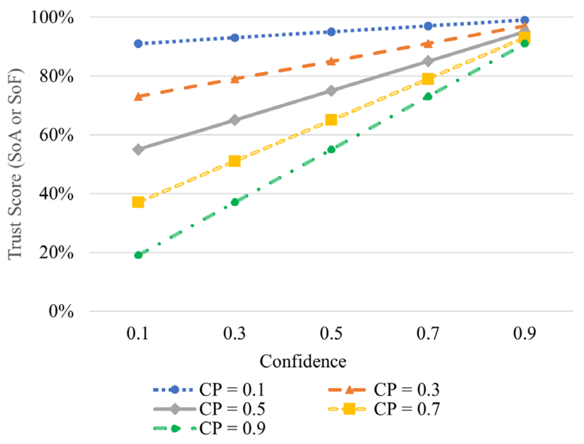

Fig. 3. Sensitivity analysis of confidence and credit points of an AA to SoA/SoF

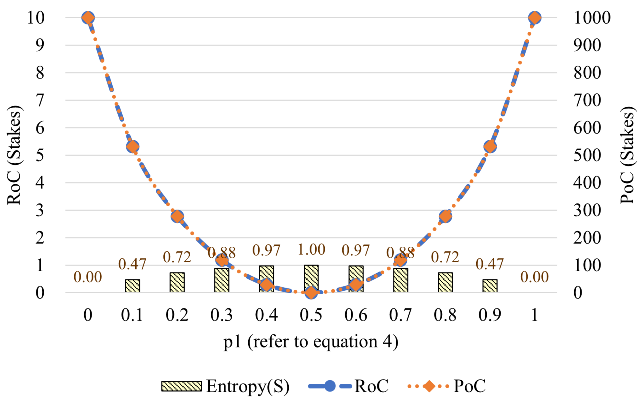

Fig. 4. Sensitivity analysis of p1 to RoC and PoC

points would possess more power to affect the trust score based on the calculating approach of trust score.

2) *Entropy to rewards and punishments:* In Fig. 4, we set the total stakes to 10,000, so the *BR* = 10 and the *PR* = 1,000. An interesting case is, based on (4), $p_1 = 0.5$ means the *Entropy(S)* = 1. Therefore, we will not put any rewards or punishments on AAs. Besides, $p_1 > 0.5$ represents more than half of appraisers approve the content, and then our system will reward appraisers who endorse the content and punish others who reject the content. On the other hand, if $p_1 < 0.5$, those appraisers who reject the content could earn rewards, and other appraisers who approve the content will receive punishments. Another attractive point in this example is when p_1 is closer to 0 or 1, the *Entropy(S)* is smaller, so the reward and punishment will increase accordingly. Conversely, when p_1 is closer to 0.5, it means that the ratio of *Majority* to *Minority* is getting small to let the reward and punishment decrease with the rising of *Entropy(S)*. The phenomenon in Fig. 4 shows that our incentive mechanism could encourage benign appraising behaviors and slash severely malicious nodes that collaborate with each other. Also, the *RoC* is always lower than *PoC* because we desire to cease malicious actions via tremendous punishments.

C. *Interactions between CC and AAs*

Essentially, message size significantly affects the performance of distributed systems, including latency and throughput. In this paper, we design six different sizes of content: 2^3 KB (8 KB), 2^5 KB (32 KB), 2^7 KB (128 KB), 2^9 KB (512 KB), 2^{11} KB (2 MB), and 2^{13} KB (8 MB) to observe the effect of different sizes of contents for the five experiments we will discuss in this section. Moreover, considering prevailing use cases of social media, people always send short text messages to others, like a tweet or a post on Facebook. According to users' preferences to make messages as brief as possible when using social media, we set the maximum message size to 8 MB. If people desire to transmit long videos or huge videos on our system, they could put links in their messages to enable others to download their content. Therefore, we believe 8 MB is a suitable upper limitation of our system.

In experiments I to III, we let the CC send messages to the Master AA node for one minute five times to record the throughput and latency. Then we summarized the average throughput and latency in each message size. For experiments I, II, and III, we define our throughput and latency as below.

a) Latency: The average waiting time is required for the whole cycle from CC sending a message to AAs until CC receiving the response message sent back from AAs.

b) Throughput: We define throughput as the average number of requests sent by CC that an AA node could process per second. For getting more accurate figures of experiment result in a single thread with synchronous requests, we use only one client of CC to send requests one by one when the previous request ends.

Note that we assume all workers agree on the content that our CC sent in experiments I to III with one hundred percent confidence (*Conf*) because the main goals of these experiments focus on measuring how the communications of different nodes affect the throughput and latency in our system. Also, human appraisers in the real world might take a few hours to several days to evaluate content.

1) *Experiment I: Normal Master AA and Worker AA nodes:* In the first experiment, we plan to realize the optimal performance that our system could achieve. As a result, we leave no failures on any AA nodes and only one master AA and worker AA node in this experiment. As shown in Fig. 5, our throughput for 8 KB contents is 602 messages per minute and 63 messages per second for 8 MB contents. As we can see, the throughput drops slightly from 8 KB to 512 KB and steeply falls when the size reaches 2 MB. This phenomenon meets our expectations because CC and AAs need more time to transmit larger messages to each other and process the content. Therefore, the latency increase and the throughput decrease result from the augmentation of message size.

2) *Experiment II: Scalability of Worker AA nodes:* We focus on the Worker AA nodes regarding our system's scalability because the number of worker AA nodes may be tremendous in the real world. Therefore, it is worthy of understanding the impact of different numbers of worker AA on our system's performance to increase the number of worker AAs in our

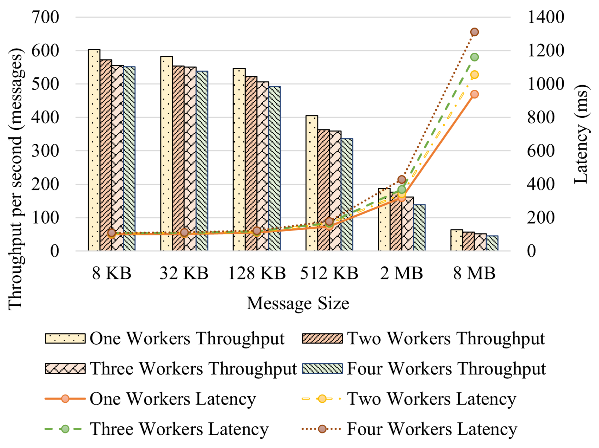

Fig. 5. Throughput and latency of scalability experiments

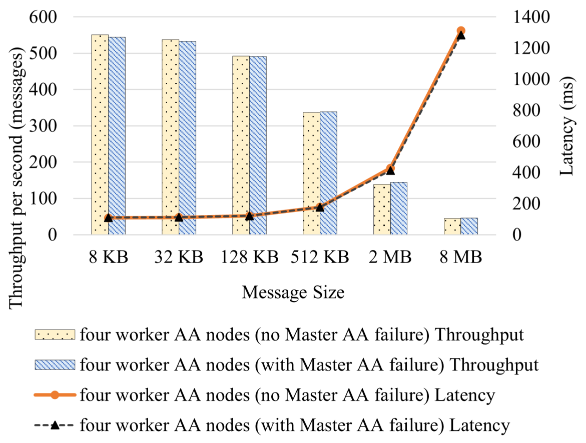

Fig. 6. Throughput and latency when the Master AA fails

system. Our experiment result of scalability is shown in Fig. 5. We could observe that, in each message size, the latency increases slightly with the augmenting numbers of Worker AA nodes. This increase in latency is because, in our system, Master AA will confirm whether all workers AA have completed their work or not at every two ms interval. Therefore, the total latency will be determined by the slowest Worker AA. Moreover, Fig. 5 displays that the latency gap is not tiny between different numbers of Worker AA nodes in each message size, which means the increase of Worker AA node will not lead to the latency growth. However, further work to scale our system to a hundred or thousand Worker AA nodes would be necessary to verify the scalability limitations of Worker AA nodes.

3) Experiment III: Crash Failure Handling of Master AA nodes: Providing that there are no Master AA nodes, our system could not normally work because the Master AA node is responsible for receiving CC's content and assigning signing tasks to Worker AA nodes. Hence, we offer another critical contribution to design a fault tolerance mechanism to prevent Master AA node failures. Fig. 6 states a comparison of latency and throughput of our system without Master AA failures and when encountering Master AA failures. This experiment executes the Primary Master AA and lets all four Worker AA nodes connect to the primary Master AA nodes. Then CC sends messages to the primary Master AA nodes, and we stop the Master AA nodes at the 10, 20, 30, 40, and 50 seconds within one minute that the CC sends messages to the Master AA nodes. As Fig 6 shows, our system could still operate normally even during the primary Master AA failures.

We adopt ZooKeeper to implement the fault tolerance mechanism. When the primary Master AA fails, the CC will get the backup Master Worker node's address, and then the CC could send messages to the backup Master AA node to make the signing process continue. Fig. 6 states that the failure of the Master AA node produces trivial influence on our system because we could observe the average throughput of the normal case (no Master AA node fails) and abnormal conditions (the primary Master AA node fails) are close. Also, the two latency lines of both scenarios approximately overlap. This result demonstrates that our mechanism is highly feasible because, based on this result, we even could ignore the impact of a failure of the primary Master AA node. We could be confident to ignore the influence of the failures of the primary Master AA because we do not execute any extra works when the ZooKeeper transfers the signing tasks for the primary Master AA Node to the backup Master AA node.

D. Interactions between CC/User to the blockchain

1) Experiment IV: Time taken when CC and User node save/retrieve data to/from the blockchain: Before sending data to the blockchain node, CC will use the BKDR hash function [17] to convert the content to a hash value. The CC then sends the hash value to the blockchain nodes. As a result, in this experiment, all the content sizes are the same. In other words, The smart contract: *addNews()* and *queryNews()* are executed once per message size and calculate the average time taken between all different message sizes. The result indicates that it took CCs on average 13.6 ms to finish the *addNews()* operation, and took Users 11.3 ms in average to complete the *queryNews()* function. The reason for the difference between the time taken to process *addNews()* and *queryNews()* is that when Hyperledger Fabric adds a new transaction to a block, the Hyperledger Fabric needs to update the CouchDB and the blocks. Moreover, the performance of executing smart contracts of our system is similar to IBM's performance benchmark [16].

VI. Conclusion and Future work

This paper offers four principal contributions:

- We propose a proof of stake entropy-based incentive mechanism that could diminish the negative impact of malicious behavior to enhance the reliability of a fake news prevention system.
- We practically implement a blockchain via Hyperledger Fabric to prove the feasibility of our design of a quorum-based fake news prevention system.

- We design mechanisms of crash failure prevention and scalability to enhance the system's robustness.
- We provide various experiment results of system performance for further reference.

Several essential case analyses are provided in this paper to demonstrate that the proof of stake entropy-based incentive mechanism we proposed is effective in decreasing the negative effect of malicious participants on our system. In addition, the experimental results show that, as we proposed in this work, it is feasible to use Hyperledger Fabric to implement the blockchain network and ZooKeeper to design the mechanism of tolerating crash failure and supporting scalability. Last but not least, the outcomes of analyses and experiments in this work can be used as a significant reference for future research.

Nonetheless, we recognize that there are still two deficiencies in the current implementation, as explained below, and we will gradually strengthen our system to conquer those weaknesses in the future.

- There are still two centralized components, ZooKeeper and Master AAs, in our system. It would be better to decentralize these two components.
- We will let appraisers evaluate each receiving content's novelty. This approach could resist one possible threat: the CCs might grab rewards by sending defrauding or similar content to AAs.

References

[1] X. Zhou and R. Zafarani, “A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities,” ACM Computing Surveys, vol. 53, issue 5, article no.: 109, pp 1–40, October 2020

[2] A. Guess, J. Nagler, and J. Tucker, “Less than you think: Prevalence and predictors of fake news dissemination on Facebook,” Science Advances, vol. 5, no. 1, eaau4586, January 2019.

[3] Google Inc, ”How google fights disinformation,” Google Inc., CA., USA, White paper, February 2019, Accessed: August 15, 2021. [Online]. Available: https://blog.google/documents/37/How_Google_Fights_Disinformation

[4] S.M. Sirajudee, N. Fatihah, A. Adamu, and I. Abubakar, “Online fake news detection algorithm”, Journal of Theoretical and Applied Information Technology, vol. 95, no. 17, pp. 4114-4122, 2017

[5] AI Foundation. ”Reality Defender 2020,” Reality Defender, https://rd2020.org/index.html (accessed August 16, 2021).

[6] Z. Jaroucheh, M. Alissa, W. J. Buchanan and X. Liu, ”TRUSTD: Combat Fake Content using Blockchain and Collective Signature Technologies,” 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 2020, pp. 1235-1240

[7] E.Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotte, C.Stathakopoulou, M. Vukolić, S. W. Cocco, J. Yellick, ”Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” EuroSys ’18: Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 2018, article no. 30, pages 1-15

[8] C. P. Schnorr, ”Efficient Identification and Signatures for Smart Cards,” CRYPTO 1989: Advances in Cryptology—CRYPTO’ 89 Proceedings, Santa Barbara, CA, USA, 1989, pp 239-252

[9] G. Haciyakupoglu, J. Y. Hui, V. S. Suguna, D. Leong, M. F. B. A. Rahman, ”Countering fake news: A survey of recent global initiatives,” S. Rajaratnam School of International Studies, Nanyang Technological University, Jurong West, Singapore, 2018, Accessed: August 16. 2021, [Online]. Available:https://think-asia.org/bitstream/handle/11540/8063/PR180307_Countering-Fake-News.pdf?sequence=1

[10] Chatila and J. C. Havens, ”The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems,” in Robitics and Well-Bing, Intelligent Systems, Control and Automation: Science and Engineering, vol. 95, M. I. A. Ferreira, J. S. Sequeira, G. S. Virk, M. Osman. Tokhi, E. E, Ladar, Eds., Springer, Cham.

[11] J. Isaak and M. J. Hanna, ”User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection,” in Computer, vol. 51, no. 8, pp. 56-59, August 2018

[12] B. Ross, A.-K. Jung, J. Heisel and S. Stieglitz (2018), “Fake News on Social Media: The (In) Effectiveness of Warning Messages”, 39th International Conference on Information Systems (ICIS). San Francisco, CA, USA, vol. 39.

[13] Poynter Institute, “PolitiFact,” PolitiFact, https://www.politifact.com/ (accessed August 17, 2021)

[14] B. Adair, M. Stencel, C. Guess, E. Ryan, J. Luther, A. Royal, “Fact checking,” Duke Reports’ LAB, https://reporterslab.org/fact-checking/ (accessed August 17, 2021)

[15] The Apache software foundation, “Apache ZooKeeper,” Apache ZooKeeper, https://zookeeper.apache.org/ (accessed August 17, 2021)

[16] P. Thakkar, S. Nathan, and B. Viswanathan, “Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform,” in IEEE MASCOTS, 2018.

[17] B.W. Kernighan and D.M. Ritchie, The C Programming Language, Prentice Hall Professional Technical Reference, 1988, ISBN:0131103709.

[18] S. King and S. Nadal, “Ppcoin: Peer-to-peer crypto-currency with proofof-stake,” Self-Published Paper, August, vol. 19, 2012.

[19] C. E. Shannon, ”A mathematical theory of communication.” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 5, no. 1, 3–55, 2001.

[20] A. Silberschatz, H. F. Korth, S. Sudarshan, ”Database System Concepts.”, 6th ed. New York, McGraw-Hill, 2010, pp. 897-898.

[21] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, ”A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks.” IEEE Access, vol. 7, pp. 22328-22370, January 2019.

[22] A. Lesne, “Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics,” Mathematical Structures in Computer Science, article no.: e240311, vol. 24, no. 3, June 2014,