Journal of Computer Sciences

Original Research Paper

A Secured Blockchain-based Information-Centric Network

[1,3]Abdelrahman Sheham Abdellah, [1]Sherif Saif, [1]Hesham E. ElDeeb, [2]Emad Abd-Elrahman and [3]Mohamed Taher

[1]Cloud Computing Lab, Electronics Research Institute, Cairo, Egypt
[2]Department of Computer and Systems, National Telecommunication Institute of Egypt, Cairo, Egypt
[3]Department of Computer and Systems Engineering, Ain Shams University, Cairo, Egypt

Article history
Received: 18-11-2021
Revised: 25-02-2022
Accepted: 11-03-2022

Corresponding Author:
Abdelrahman Sheham Abdellah
Cloud Computing Lab,
Electronics Research Institute,
Cairo, Egypt
Email: abdosheham@eri.sci.eg

Abstract: In this study, we propose an Information-Centric Network (ICN) approach for the Internet as an alternative to the present host-centric architecture. The proposed approach solves present Internet challenges, where most Internet users nowadays are involved in seeking knowledge by searching through large amounts of data, independent of the data's physical locations and these users usually have requests that need prompt responses. Hence, Internet requirements have got a new shape and the whole Internet paradigm should be shifting where different network considerations are needed. In this context, ICNs can play a vital role where the host-centered architecture is replaced by a content-centered one since the content itself is the aim and not the location. However, the ICN paradigm as a substitute for traditional Internet faces some challenges in terms of security and performance. ICN needs to be protected against some threats such as Denial-of-Service attacks (DoS), hacker attacks, loss of data, data replication, and cache pollution. To accomplish this, we propose this Secured Blockchain-Based ICN (SBBICN) implementation that exploits the secure aspects of Blockchain technology such as data integrity and non-tampering to secure the ICN against the aforementioned threats. In this proposed system, we describe and develop a voting system based on a blockchain consensus algorithm to avoid a single point of failure during the verification process and we apply the system using an Ethereum smart contract to verify the effectiveness of the proposed system. The experimental results and the security analysis demonstrate the effectiveness of the SBBICN proposal when compared to other schemes in the literature.

Keywords: Information-Centric Network, Blockchain, Internet Infrastructure, Future Internet, Smart Contracts

Introduction

An encouraging popular solution to several potential Internet research proposals is the idea of the Information-Centric Network (ICN). This technology incorporates in-network caching and multi-sided connectivity via replication and interaction models that separate senders and receivers (Conti *et al*., 2020). The goal is to offer a stronger network infrastructure operation that is more resilient to degradation and disruption. The Internet paradigm, which is applied nowadays, is a host-centric based model and all Internet information transfers are conducted by developing the contact networks between sender and receiver. This Internet host-based model is well-tailored to the early use of the Internet. The usage of the World Wide Web has changed over the years. Online videos are the major contribution to networking traffic and Hyper Text Transfer Protocol (HTTP) videos are a prime example of this. In Snapchat, Instagram, E-learning, online trading, YouTube, and so on, people do not pay attention to ”where” they can get information that they are interested in, but ”what” information actually is (Chen *et al*., 2020). There are many features such as Network Address Translation (NAT), Domain Name Server (DNS), multicast, multi-homing, mobility, security for multi-homing, security for mobility, and so on. are added to the current Internet protocol stack. As a result, tracking these developments make the Internet more and more complex. From that, the host-based Transmission Control Protocol/Internet Protocol (TCP/IP) Internet is becoming too heavy to offer the best performance to the end-users (Conti *et al*., 2020). To overcome these

limitations, many science communities are motivated to create informational networking that relay on the content itself. In ICN, the host-centric architecture is substituted with a content-centric architecture, as the content is more significant than its location. By using this concept, information is not handled with an IP address but other naming schemes are used to differentiate items (Din *et al*., 2019). Furthermore, ICN provides various advantages over traditional networks such as simplified content access, distribution, security, and in-network caching (Negara and Syambas, 2020).

ICN tackles many problems of classical Internet structure such as content moved within the site, content moved to a different site, the site changed domain, source temporarily unreachable, or content permanently unavailable (Eum *et al*., 2012).

However, the ICN paradigm is still facing some challenges in terms of security and data integrity. One of these concerns arises when a publisher registers its content in ICN nodes, there is a risk of this information being tampered with. Furthermore, when a malicious ICN node refuses to forward data to other ICN nodes or users, this will lead to further delay in the network.

To overcome ICN limitations, blockchain technology can be a promising tool to solve many ICN security issues because of certain features such as decentralization, immutability, consensus-based, and timestamp-based (Zeng *et al*., 2020).

By using the blockchain model, all of the executed transactions that carry the ICN node's actions are committed to the global blockchain and verified against any unauthorized action or access (Berdik *et al*., 2021). Each blockchain entity keeps a copy of data. As a consequence, any ICN node cannot deny or refuse transactions that have been approved by the blockchain. Blockchain can achieve a global agreement for the whole sequence of content (Chen *et al*., 2021). Thus, an incompatible record/transaction will be deleted directly when it is checked. Blockchain features such as non-repudiation and non-tampering ensure a secure availability of content in ICN.

Problem Statement

Even though the ICN model is an up-and-coming solution that tackles many present Internet issues, it is still not mature enough in the security aspect (Dutta *et al*., 2021). While security is a significant aspect of the ICN paradigm, ICN still faces many issues that negatively affect its performance and security robustness as follows:

- First, most ICN models use self-certifying naming based on public-key cryptography (Nour *et al*., 2019). This type of naming scheme implies some issues such as key compromise and it needs to be resolved by a third party by public key management (Zhang *et al*., 2021)
- Second, ICN may suffer from multiple users who publish the same data content and this will cause a useless overhead on ICN nodes in terms of processing, mining (verification), and storage. Replicated data from different sources will also affect the ownership integrity and the real owner will be lost (Fotiou, 2020)
- Third, when malicious users fill the cache with unpopular content, this will cause a delay when the actual request comes. This issue is called cache pollution. Furthermore, cache pollution will impact the cache performance in ICN nodes because the replicated data will reserve many unnecessary locations in the cache (Man *et al*., 2021)
- Fourth, data in ICN also is vulnerable to being altered or deleted. Because of the importance of data integrity in publisher-subscriber networks, ICN should provide receivers with a security mechanism to verify the integrity of the data objects (Sokolov, 2021)
- Fifth, Distributed Denial of Service DDos, is one of the most common methods hackers use to compromise such a network. The DDos attack is a way that hackers make certain online services unavailable by flooding them with excessive fake traffic (Conti *et al*., 2019)
- Another issue that should be considered is access control and how we can control the process of how publishers can join ICN safely

Related Work

Blockchain has become a promising technology that can be used to ameliorate such a system. As in (Asaf *et al*., 2020), blockchain technology has been commonly used for distributed payments, fund tracking, healthcare systems, and cloud infrastructure. ICN is also a prime example of integration between blockchain technology and ICN. by using the power of blockchain in terms of decentralization, tamper-proof, and consensus, the researchers were inspired to use these remarkable features to boost ICN security.

Pan *et al*. (2020), the authors introduced a trust-information-centric network architecture fueled by blockchain and Artificial Intelligence (AI) to deal with the security issues of Beyond Fifth Generation (B5G) applications such as malevolent accidents caused by untrustworthy information in vehicle navigation and autonomous systems. They designed a scheme called Trust Coin in ICN for B5G and used the blockchain features to measure the trust of B5G nodes in a diverse and fine-grained manner, based on not only content trust but also producer trust. This framework addresses the data trust problem created by manipulation, counterfeiting, and hijacking in ICN for B5G. The framework in (Pan *et al*.,

2020) relies on the NEO platform. Da Hongfei and Erik Zhan formed NEO as Ant Shares in China in 2014 and it was rebranded "NEO" in June 2017. NEO is considered a blockchain-based network with its token with the ability to create digital assets and smart contracts.

NEO plans to use smart contracts to simplify the ownership of digital properties, with the ultimate goal of creating a distributed network-based smart economy infrastructure (Coelho *et al*., 2020). This scheme (Pan *et al*., 2020) provides a stable processing environment using Trust Coin's consortium blockchain, which incorporates the benefits of both public and private blockchains. Trust Coin system is better suited for large data processing in B5G because of its low energy consumption, low latency, and high security.

However, in SBBICN, we used a different blockchain platform which is Ethereum. As seen in Table 1 there are many key differences between Ethereum and NEO. In the blockchain field, Ethereum has a formidable power. Ethereum's benefits in terms of adoption and control are undeniable. In contrast, the market share of Ethereum is much more than NEO (Hu *et al*., 2021). Therefore, we preferred to use Ethereum as a blockchain platform in our implementation.

In 2020, a Secure Blockchain-based Access Control (SBAC) framework for the information-centric network was represented (Lyu *et al*., 2020) to ensure a service provider can securely share, verify and cancel the content. They also designed a matching access management model to gain hierarchical access and present an access token scheme based on blockchain to withstand the single point of failure and balance privacy and audit in ICN.

In (Li *et al*., 2019), the authors developed a trusted Blockchain-based ICN (BICN) architecture for content delivery. This mechanism can feed the records of behaviors on ICN nodes to the blockchain faithfully, which is the key to guaranteeing the tracing of the malicious ones. In other words, the entire process of content delivery is verified in an implicitly trusted method by the use of the excellent functionality of the blockchain to identify a malicious ICN node. In addition, they applied the architecture on a private blockchain and proves that the use of BICN can protect ICN from hijacking and jamming problems by applying a general blockchain that uses hash functions to verify each transaction and broadcast the result to all nodes. Researchers in (Li *et al*., 2019) applied their design (BICN) in a private blockchain. The private blockchain is not fully decentral listed and this is one of the key drawbacks of a private blockchain that goes against the distributed ledger technology or blockchain theory in general (Bera *et al*., 2021). In opposition to this, a public blockchain avoids single authority control and allows many parties to join the verification process which motivated us to apply our framework to a public blockchain.

The impacts of cache attacks on BICN networks were examined experimentally by the authors (Roy *et al*., 2019). They tested the cache attack in which the attacker loaded unpopular information into the cache, causing the user to download the data from the web servers. In this paper, the authors utilized the hyper ledger fabric, a blockchain framework, and a popular implementations simulation environment for blockchain projects (Chacko *et al*., 2021). Many businesses employ Hyperledger fabric, which is a private and permissioned implementation. Hyperledger Fabric has a pluggable consensus algorithm that may be customized to meet particular application needs. They looked at how long it takes to query the current state of the blockchain and request an update in the blockchain as well as all versions of the ledger.

Furthermore, as stated in (Abdellah *et al*., 2020), Blockchain Public Key Management (BC-PKM) was proposed for Named Data Networking (NDN), an architecture of ICN, to take advantage of the decentralized and tamper-proof design features of Blockchain. That paper further proved that BC-PKM could resist a variety of attacks from adversaries that compromise less than half of the public key miners.

Secured Blockchain-Based ICN (SBBICN)

In order to employ the advantages of blockchain in ICN, we develop a system that combines the security aspects of blockchain to secure the publishers, subscribers, and data as well.

The proposed Secured Blockchain-Based ICN (SBBICN) consists of four parts (Fig. 1):

- Publishers: These are the individuals or resources that provide content.
- ICN nodes: Are the network equipment that is used for managing, storing, and transmitting information
- Blockchain network: Used for the validation and the authentication processes
- Subscribers: Are the users who request data/content from the system

All of these parts act together to maintain SBBICN (Secured Blockchain-Based ICN). Generally, all types of online resources can all serve as publishers. For content delivery, on one hand, the publisher who plans to transmit its content to the consumer should first send a request message to the blockchain. In the blockchain, the publisher will be authenticated by hash functions such as the Keccak function as seen in Algorithm 2 to ensure his/her content is unique and valid. As seen in Fig. 1, after the blockchain completes the verification and authentication process, the publisher can broadcast its content to ICN nodes. Then, the content will be propagated among these nodes via its in-network caching feature. Furthermore, in SBBICN, we can add a voting

stage to allow publishers to vote for an operation such as preventing invalid content or removing suspected users and publishers. On the other hand, to fetch content, the subscriber should send a request message to the blockchain. The blockchain will check its identity and will give him/her permission to search for content. After that, the subscriber can search for such content by typing the content's name. The requested content will be provided to the subscriber if it is available by the ICN nodes (Fig. 1).

Implementation and Security Analysis

In this study, we will implement the SBBICN system to utilize blockchain technology to protect an ICN system. We will also test the effectiveness of this system against some security issues to show that this innovative technology will overcome these threats.

Our implementation is based on Ethereum smart contract and its programming language Solidity. By using smart contracts, we can have a balance (Ethers) and we will be able to submit transfers through the network. Smart contracts, however, are not operated by a user; instead, they are distributed to the network and run according to a set of instructions (Vivar *et al*., 2021). Users' accounts will then communicate with a smart contract by making transactions that cause the smart contract to perform a feature or a function. Smart contracts, like standard contracts, will define rules and have them enforced automatically by the code (Sayeed *et al*., 2020). Our SBBICN was developed, implemented, and tested using Remix IDE (http://remix.ethereum.org).

Table 1: Ethereum Vs. NEO

Evaluation criteria	Ethereum	NEO
Market adoption	Very high	High
Consensus Mechanism	(PoS)	(dBFT)
Divisibility	YES	NO
Speed	15 transactions/s	10,000 transactions/s
Censorship/regulations	Free	Has some censorship issues
Exchange support	Available on all major exchanges	Not supported yet in many main exchanges
Language support	Solidity	C, Java, and python
NO. of decentralized applications	More than 3000	Less than 100

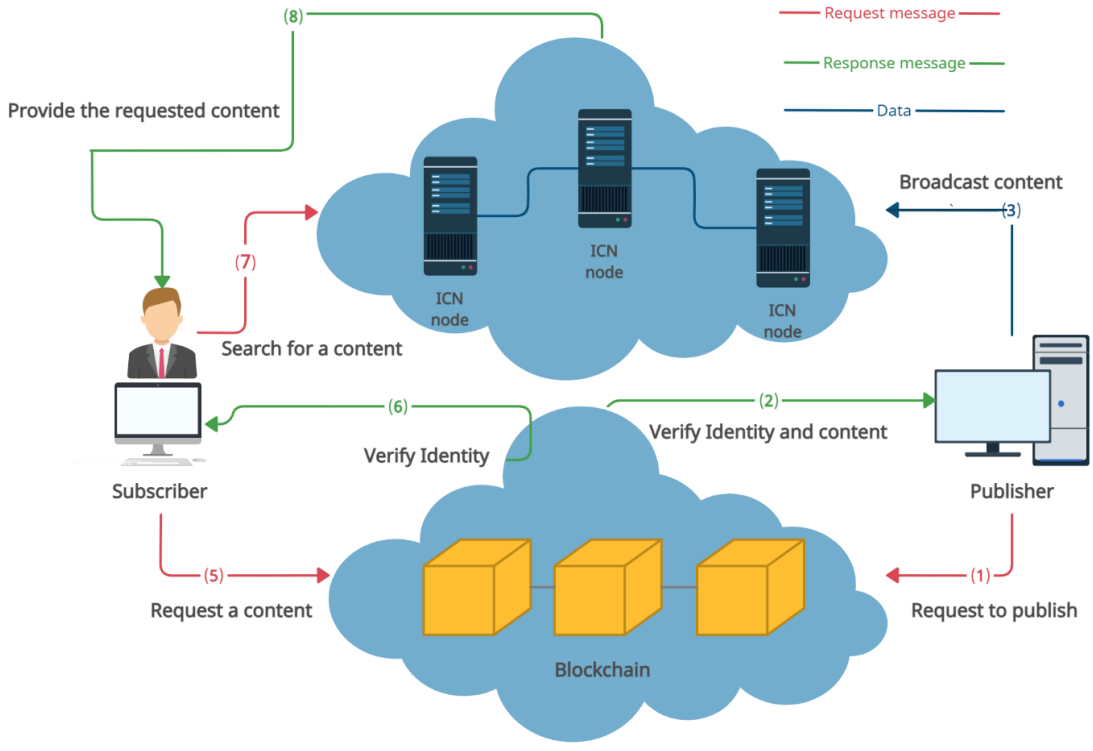

Fig. 1: Blockchain technology in ICN

Algorithm 1: Publisher Registration

```
function REGISTER(Publisher Address)
  if Stage 6 = Resgister Stage then
    Return;
  end if
  Mapping (Publisher ← address) publishers
  Publishers [Publisher Address]. voted false
end function
```

Algorithm 2: Add Unique Content

```
function ADDCONTENT (New Content, publisher Address)
  flag ← 0
  has hed Data ← Keccak (New Content)
 for i = 0 → Max OF publishers do
        for y = 0 → Max OF publishers Contents do
If (hashed Data=Keccak (Publishers[i]publishers Contents[y])))
                    {print Repeated data.. not allowed
                    flag ← 1;
                    break;}
           end for
end for
  if flag is 0 then
    Publishers [publisher Address] ← New Content
  end if
end function
```

Furthermore, we utilized Meta Mask to communicate with Remix IDE to handle account management and connect the users/publishers to the blockchain. Figure 2 describes how Meta Mask integrates with our smart contract. Meta Mask is a tool that allows users to maintain their accounts' tokens and keys in several methods, including hardware wallets while keeping them separate from the site's environment. This is far more secure than keeping user keys on a single central server or even in local storage (Arora *et al*., 2021).

There are three entities in our smart contract of SBBICN, publishers, ICN nodes, and subscribers. Each publisher (content provider) can participate in the system by calling the register function in the contract. Figure 3 demonstrates the messages sequence diagram for joining and adding new content in SBBICN.

Algorithm 3: Accept votes from the Publishers

```
function VOTE (vote, publishers Address)
"Enter your vote"
   publishers [publisehrs Address]. Vote Value ← vote
  if vote =Yes then
    Yes Counter++
  end if
  if vote = No then
    NO Counter++
  end if
end function
```

Next, we will show the pseudocode of the algorithms that were used to develop the important functions in our smart contract. We used struct data type to represent and store the publishers and their meta-data. Each publisher has a unique address, array of contents, and a voting flag.

Struct Publisher {Publisher Address, Publisher Contents [ ], Voting Flag}.

When a publisher wants to join SBBICN network, he/she should first execute the Register function (Algorithm 1).

We used the mapping function which is a solidity-based function to store data which is a key-value pair (Algorithm 1). Therefore, the key of the array will be the publishers' address instead of the default integer keys. Then, if the registration process is completed, the publisher's address will be recorded in the blockchain. When a publisher decides to add and broadcast such content in the ICN nodes, he/she should call the ADDCONTENT function (Algorithm 2). To avoid any replication and ensure that any publisher will add and broadcast unique content, we used the Keccak function to generate a unique hash number for each item that will be published and search if this hash already exists or not. Keccak function is a function that takes in arbitrary size input and performs a fixed size output (Braeken, 2020). Keccak has some properties:

- It is deterministic which means $hash(x) = h$ every time we calculate a $hash(x)$
- It consumes less computational power than other computational power (Martínez *et al*., 2022)
- It is irreversible. Therefore, given h, it is hard to find x such that $hash(x) = h$ (Rathod *et al*., 2020). As a result, hackers cannot retrieve the transaction content from its hash because the one-way feature of hashing techniques
- It can deal with and hash any type of data (strings, integers, and floats)

Algorithm 4 Calculate vote final decision

```
function CALCULATE VOTES
  if Yes Counter >= No Counter then
    decision ← Accepted Operation
publisher Content ready to be published
  end if
  if Yes Counter < No Counter then
    decision ← Rejected Operation
  end if
end function
```

If the candidate already exists in the blockchain, the smart contract will set a flag to 1 and prevent this content from

being published. On the other hand, if the hash was found in the blockchain before (the flag still has 0 value), the smart contract will add the content successfully to the Publisher struct to be broadcast via ICN nodes (Algorithm 2).

However, as our contract depends on for-loop functions to search in the publisher struct, the complexity of the ADDCONTENT function is $O(N^2)$. Therefore, we still need to enhance our algorithm to reduce the time complexity of deploying a smart contract.

Replicated Data and Data Origin

In the traditional Internet and information-centric networks, a network may suffer from multiple users who publish the same data content and this will impact the cache performance in ICN nodes because the replicated data will reserve many unnecessary locations in the cache (Man *et al*., 2021). Replicated data from different sources will also affect the ownership integrity and the real owner will be lost.

In our proposed system, SBBICN, we ensure that adding any content in the network is completed by coupling the account address (publisher address) with the content that he/she wants to publish or broadcast. Because of using the power of hash functions in the solidity language of blockchain, we can prevent any replicated data and protect the ownership integrity (Taş and Tanrıöver, 2019). Therefore, when the same publisher or any other publisher adds replicated data, the system will refuse the request as explained in Algorithm 2. Figure 4 describes the overall process in a simple flow chart.

To apply this in a real use case, suppose the following parameters:

- Publisher 1 added a unique string content called "my first file"
- As a result, the transaction is completed and the content is successfully added (Fig. 7)
- Suppose that a publisher by mistake requests to add the same previous content again. SBBICN should prevent any replicated data. Figure 6, our smart contract shows an error message for the publisher to prevent him/her from adding the replicated content

Cache Pollution/Poisoning

Most of the cache algorithms store the most recent information or the popular demanded content but what happens when a malicious user starts to request a fake content or a fake request?, this is called cache pollution (Zhou *et al*., 2020). In other words, filling the cache with unnecessary content from malicious users will replace the proper content in the cache and this will increase the response time and affect negatively the performance of ICN.

However, in an ICN-based blockchain, each transaction/request is registered with a timestamp and account address (Conti *et al*., 2019). Therefore, it is easy to detect repeated fake requests and users. When the system suspects any repeated request from the same user in a short interval time, the request will be refused. Here is an example from our implementation. When a publisher runs the ADDCONTENT function to add new data, the Meta Mask plugin will check the publisher address and check the required ethers to complete this operation. Then the blockchain smart contract, SBBICN, will perform the transaction and all the results will be recorded in a non-tampered record including (transaction hash, block hash, function hash, and the timestamp) (Fig. 5).

As a result, we can check the request rate from such a publisher to secure the system from any malicious or unnecessary requests.

Data Integrity

Traditional ICN networks may suffer from a lack of data integrity because data is vulnerable to being edited and modified or even deleted.

Using blockchain in the ICN network will protect data against any alteration or being removed because in blockchain once information (transaction) is verified, it is stored in the system in an immutable way (Conti *et al*., 2019). This process depends on that every transaction in the blockchain has a hash number and is related to a block that has a hash number and each block is coupled with the hash number of the previous block. Therefore, any modification in the content will change the hash number of the block to appear different from the already verified hash of the block and all other hashes of the other blocks will also be changed. Therefore, it becomes easy to be detected any changes by the miners (verifier nodes). To emphasize the power of connected hashed block, Next, we will apply the concept of SBBICN to different use cases. To begin with, suppose we have three transactions (Fig. 8), each transaction represents a publisher that wants to add new content to SBBICN.

Each publisher will add his content to the ICN and this content will be verified and recorded on different blocks. Now we will calculate the hash of each block, the hash will be calculated based on the account address and the content. We will use the Secure Hash Algorithm (SHA256) for this purpose. SHA256 is one of the cryptographic hash functions. The SHA256 algorithm produces a 256-bit (32-byte) hash that is nearly unique and is referred to as a one-way function (Cortez *et al*., 2020). This qualifies it for data integrity checks, challenge hash verification, anti-tamper, digital certificates, and blockchain (Lugo and Pedraza, 2020). We should also take into consideration that in the blockchain all the hashed blocks are coupled with each other. In other words, each hash depends on the previous block hash and the block data itself. Therefore, any alteration in any block will reflect a change in all block hashes (Fig. 8).

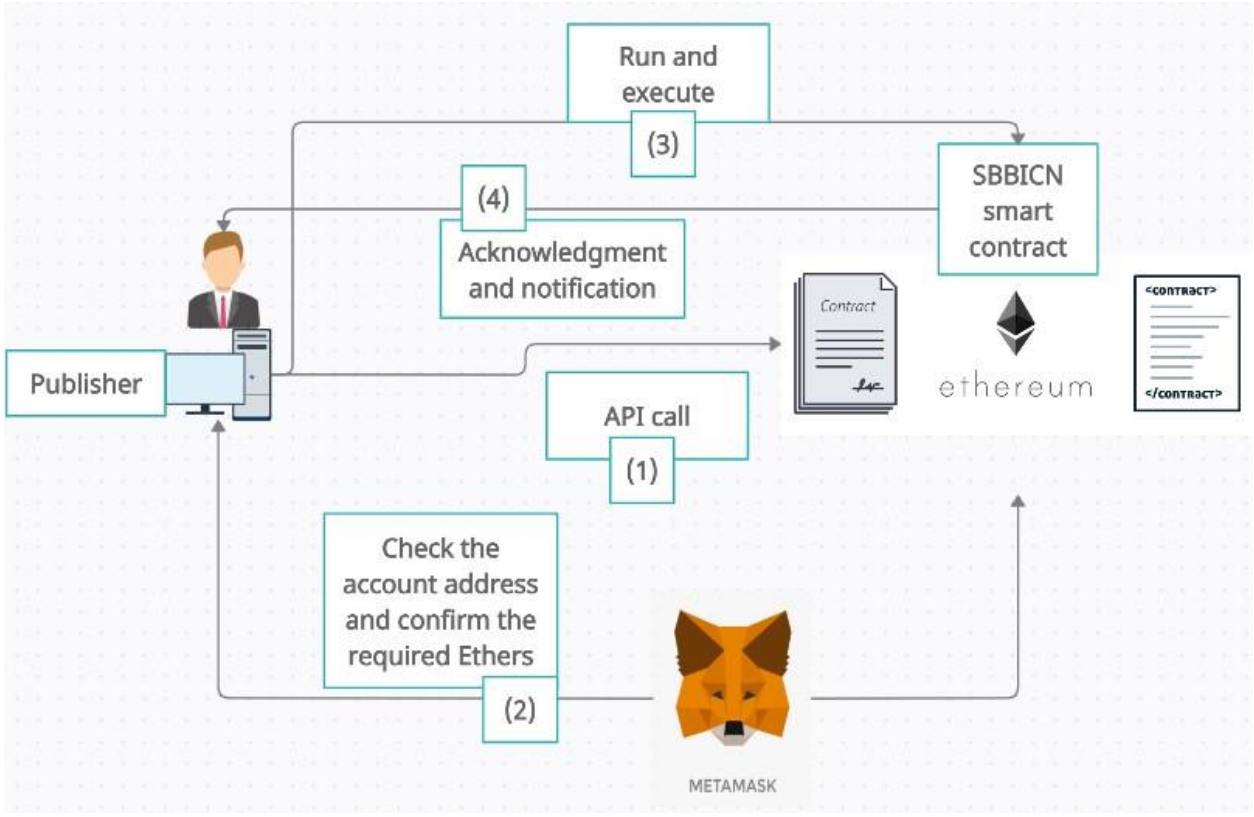

Fig. 2: How meta mask interacts with our SBBICN

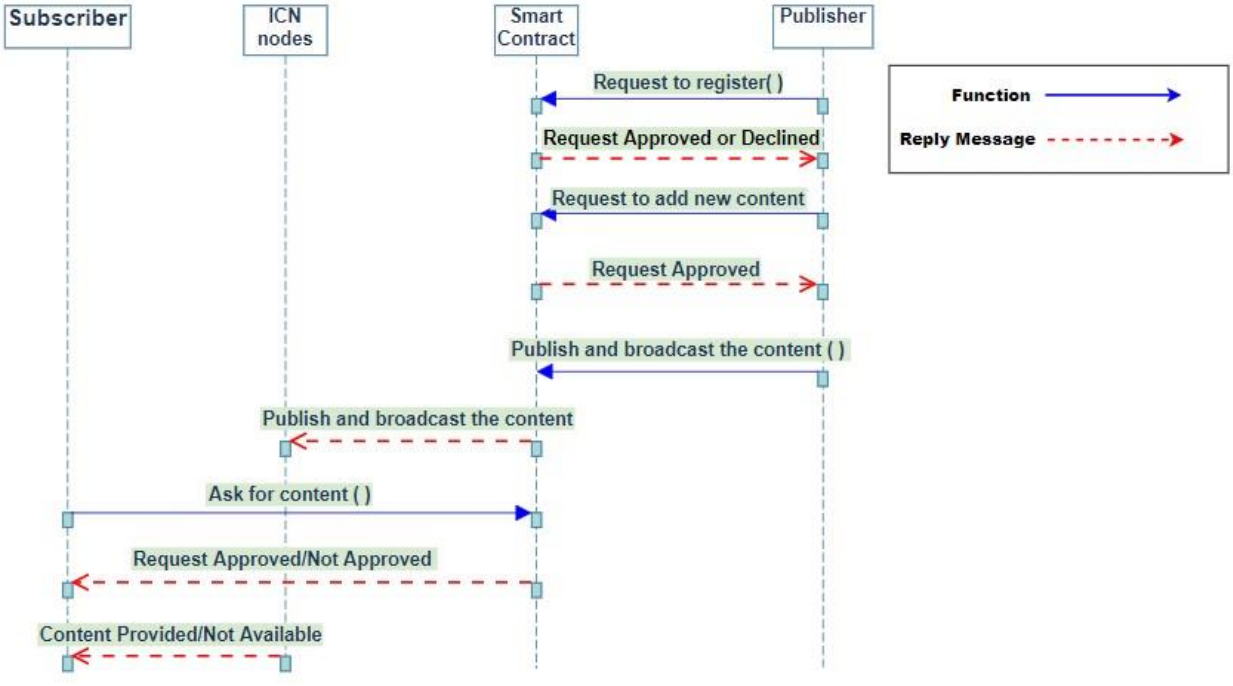

Fig. 3: Message sequence diagram of SBBICN

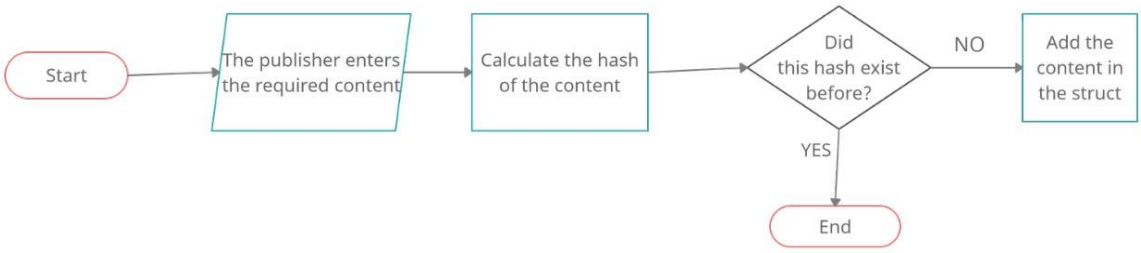

Fig. 4: Flow chart of adding content process in SBBICN

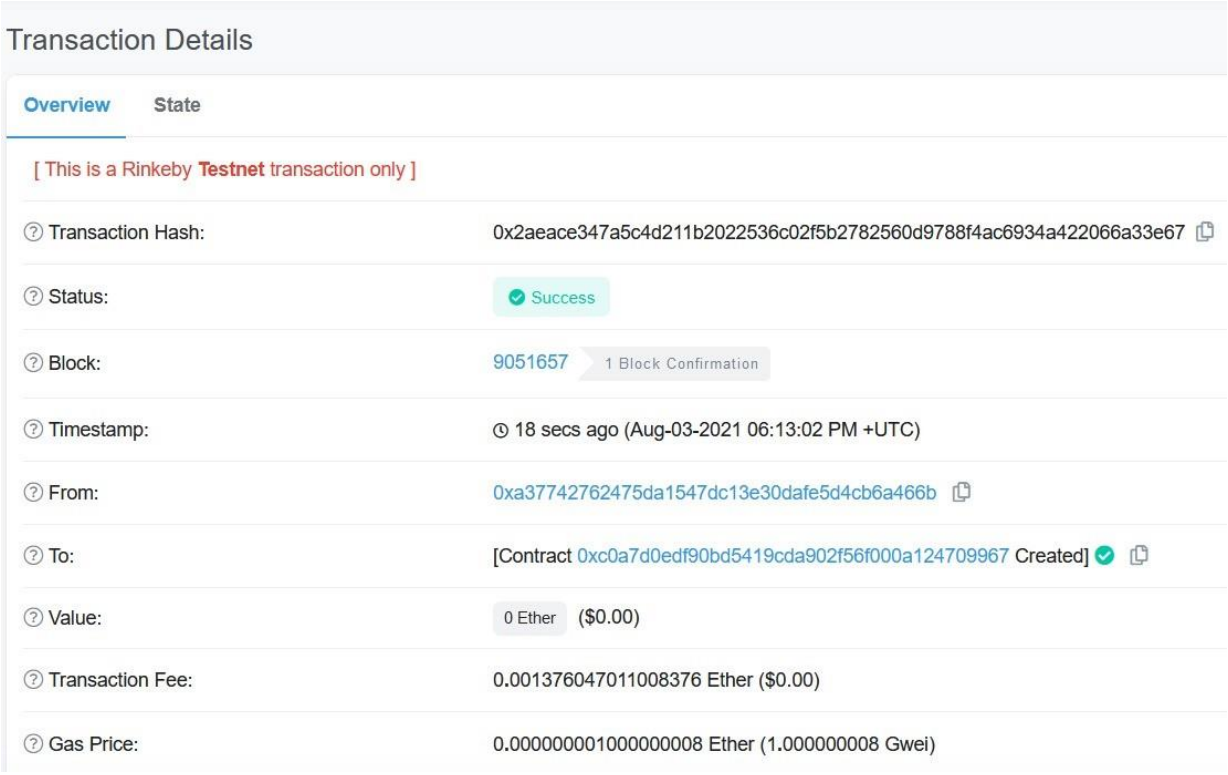

Fig. 5: Transaction details after completing an operation

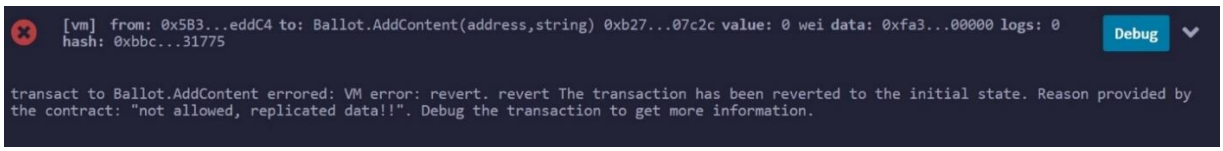

Fig. 6: An example of a request for a repeated content that is not allowed in SBBICN

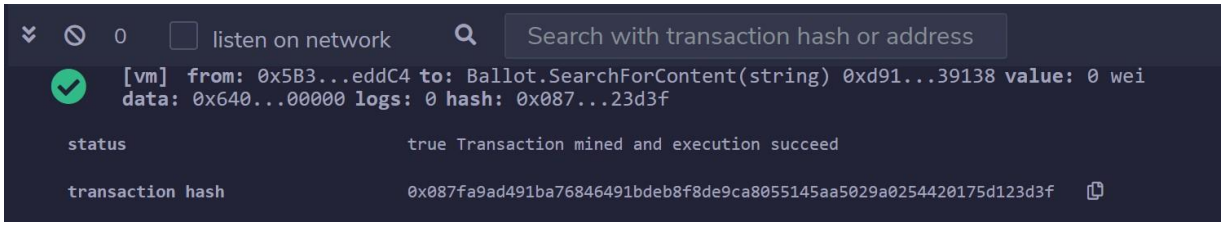

Fig. 7: An example of a successful transaction using SBBICN

Fig. 8: Hash numbers before modification

Next, we will make a little change in publisher 1 by changing the letter (d) in the data with a capital (D). Now, the publisher 1 content will be (this is publisher 1 Data) instead of (this is publisher 1 data). As a result, the overall hash numbers will change due to this small modification. In figure 9, the hash numbers of the blocks are completely changed because of a small change. Therefore, it is easy to detect any alteration in the registered data.

Access Control

In SBBICN, we add a voting mechanism as an additional layer of security in many cases such as excluding such a publisher or restricting its content.

The flow chart below (Fig. 10) represents the steps of this procedure in SBBICN. In Algorithms 3 and 4, we represent the pseudocode of our voting process.

Next, we implemented and tested this in two different use cases.

Use case 1:

- We have three publishers (A, B, and C) and there was a correct operation that was submitted to be approved by the voters.
- Publisher A and Publisher B voted with 1 (YES) (Fig. 11)
- publisher C voted with 0 (NO) (Fig. 12)

As shown in Fig. 13, the overall result of the voting process is YES (the majority approved the transaction).

Use case 2:

- We have three publishers and there was suspected operation was submitted to be approved by the voters
- publisher A and Publisher B refused the operation and voted with 0 (NO)
- Publisher C accepted this operation and voted with 1 which means YES
- In figure 14, the overall result of the voting process is NO (the majority does not approve the transaction)

After testing the voting system in SBBICN, it can be stated that our proposed system can help and enhance security.

However, (50%+1) system has some disadvantages such as:

- The scalability: Increasing the number of users or publishing will increase the complexity of collecting votes (Saad *et al*., 2019)
- (50%+1) the system needs also all involved entities to vote and participate in a prompt time not to cause any delay to make a decision (Saad *et al*., 2019)
- The is also a possibility for an organization to affect and control (50%+1) of the involved parties and this will lead to approval or refusal of such a transaction and increase the single point of control role which is not an objective in blockchain technology (Sayeed and Marco-Gisbert, 2019)

Protection Against DDos

DDos stands for Distributed Denial of Service and it is one of the most common methods that hackers use to shut down sites. DDos attacks make those web resources inaccessible by massively false traffic by hackers (Singh *et al*., 2020) (Sharafaldin *et al*., 2019). As blockchain is a decentralized technology, it can mitigate this security issue. As illustrated above, by using blockchain technology, there is no single point of verification, but multiple nodes are involved in the process of validation, voting and so on (Haider *et al*., 2020).

To be secure against DDos many factors should be considered such as the number of attacks and how many nodes the attacker can break in a short interval of time (Saad *et al*., 2019). As illustrated in the previous section, the verification process in our implementation, SBBICN, is performed by many nodes. Therefore, hacking and compromising many different nodes in a short time is hard and needs a synchronized attack and powerful computational capabilities (Behal *et al*., 2018), thanks to the decentralized nature of blockchain.

Fig. 9: Hash numbers after modification

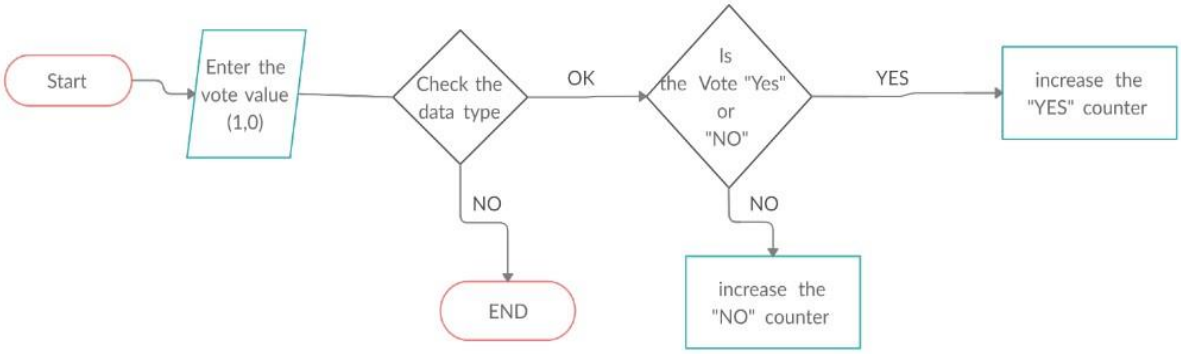

Fig. 10: Steps of (50%+1) mechanism in SBBICN

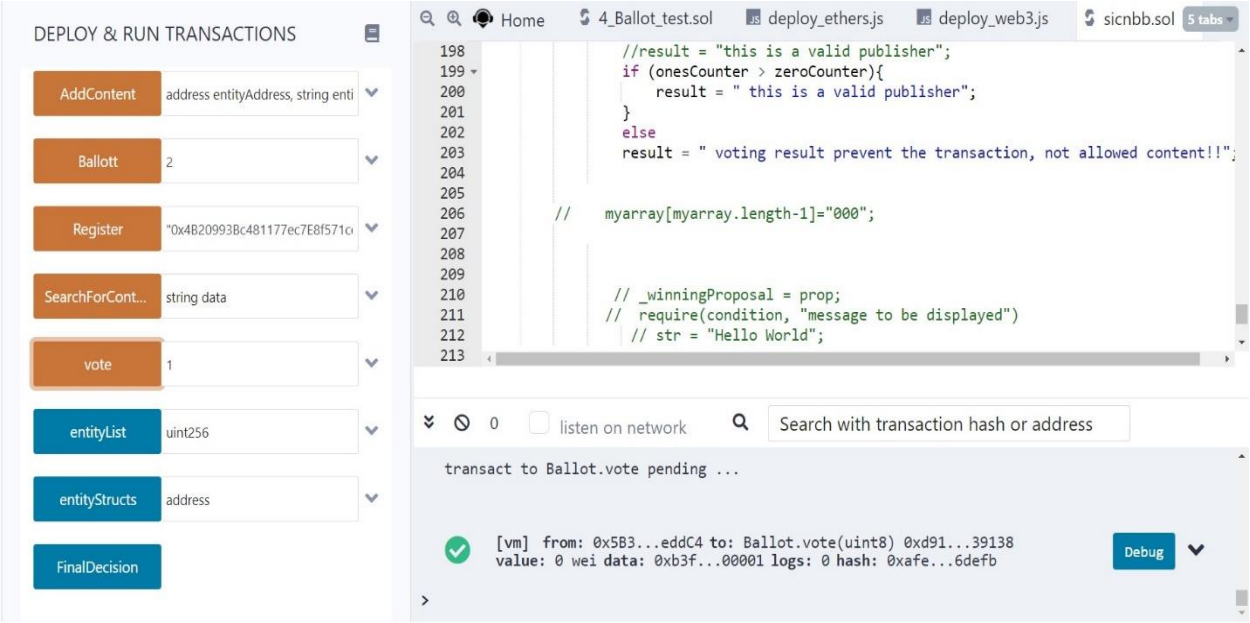

Fig. 11: Publisher voted with YES

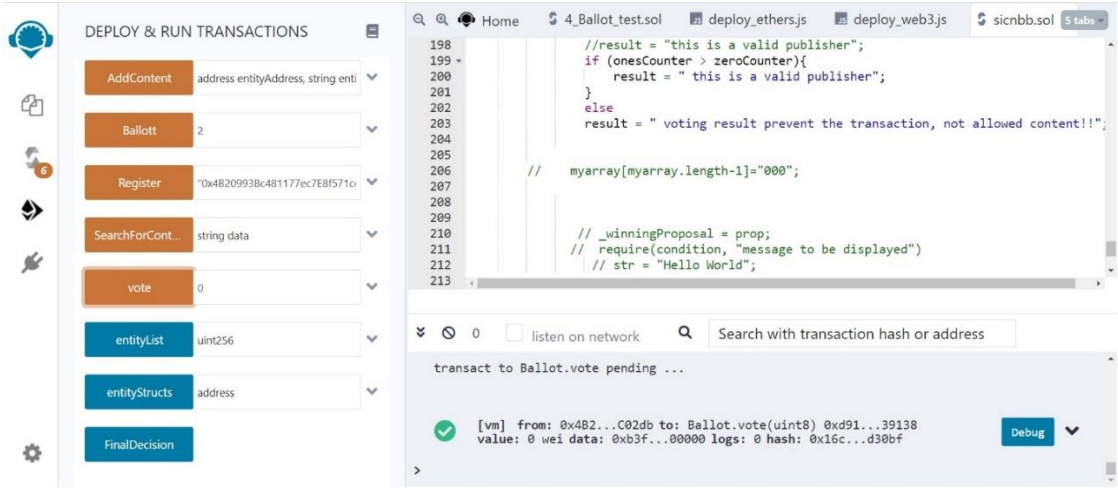

Fig. 12: Publisher voted with NO

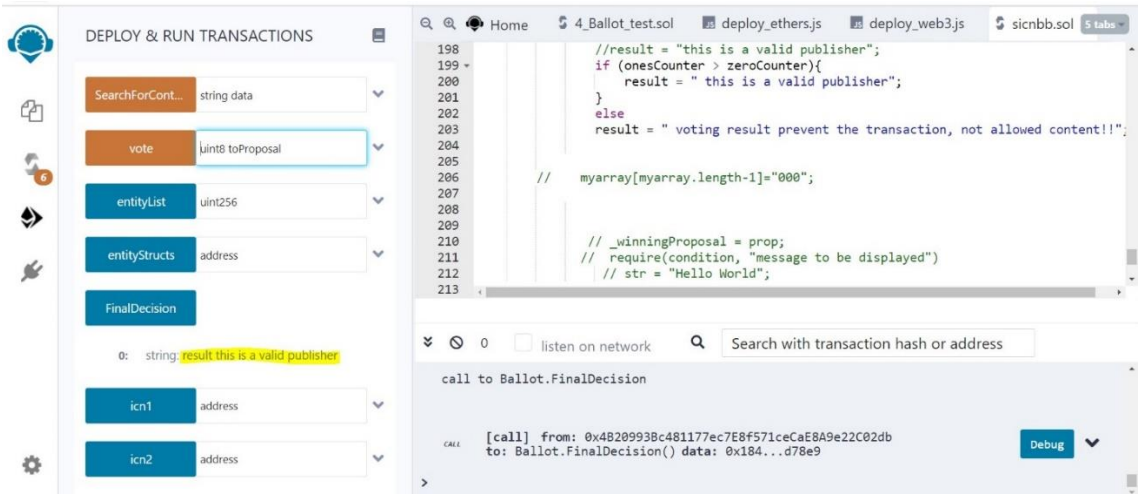

Fig. 13: The overall result of case no.1

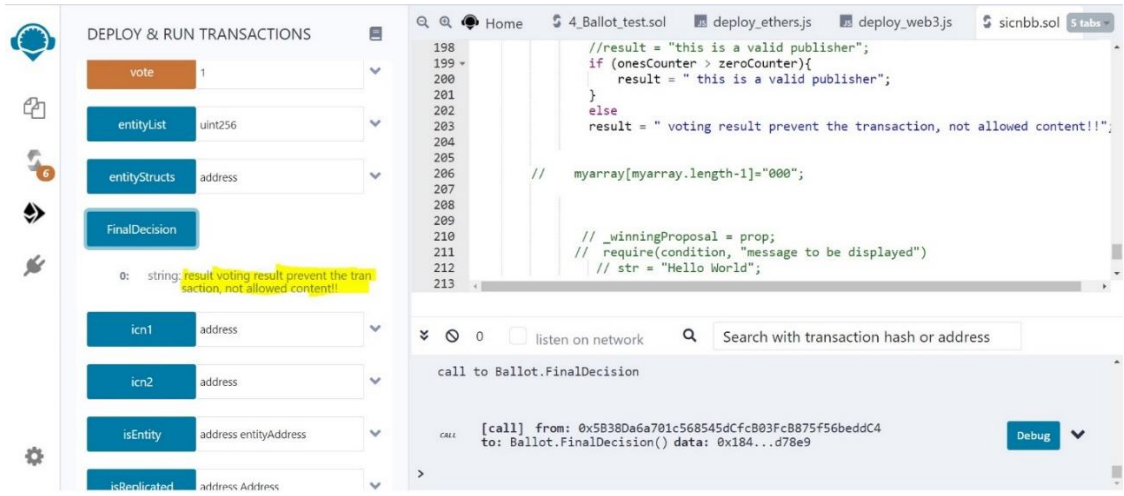

Fig. 14: The overall result of case no.2

Comparison and Performance Evaluation

Security Performance Comparison

Table 2 shows that our proposal not only can secure the system against cache pollution like in ABAC (Li *et al*., 2018), Live (Li *et al*., 2014), SBAC (Lyu *et al*., 2020), BICN (Li *et al*., 2016) and (Nour *et al*., 2021). SSBICN also is the only proposal that focuses on how to prevent the publishers (providers) from broadcasting a replicated content by comparing the combining hash number of the publisher address and the candidate data with the hash of the stored data. As a consequence, preventing any replicated content will affect positively the data integrity, cache performance, storage performance, the bandwidth and protect the data origin as well. Moreover, the degree of access control in ABAC (Li *et al*., 2018), Lightweight Integrity Verification (Live) (Li *et al*., 2014), and Blockchain-based efficient privacy-preserving and Data Sharing Scheme (BPDS) (Fan *et al*., 2018) are considered to be at a medium level because they focus only on read-only content and they do not consider the hierarchical access. They also only provide the content provider with the ability to share the content. Blockchain-Based Access Control (BBAC) (Di Francesco Maesa *et al*., 2017) and Fair Access (Ouaddah *et al*., 2017) do not test their scheme against cache pollution and DDos attacks, while they have a high access control as they achieve user anonymity, tamper-proof, anti-counterfeiting, multilevel content access as in BICN (Li *et al*., 2016), (Nour *et al*., 2021).

Moreover, in SBAC framework (Lyu *et al*., 2020) the authors created a matching-based access control model to gain hierarchical access and offer a blockchain-based access token structure to avoid a single point of failure while maintaining anonymity and audibility. Like SBAC (Lyu *et al*., 2020), our proposed framework can secure such a system from cache privacy attacks, DDos attacks, and man-in-the-middle attacks, but our implementation also can protect the system from replicated data to ensure data origin which is not included in (Lyu *et al*., 2020).

Gas Cost

The charge or pricing value, necessary to successfully conduct a transaction or execute a contract in the Ethereum blockchain platform is referred to as gas cost which is measured in Ethers (ETH) (Li, 2021). The actual price of the gas is set by supply and demand among network miners, who can refuse to execute a transaction if the gas price does not match their threshold and network users seeking processing (Donmez and Karaivanov, 2022). Table 3 shows the gas cost of deploying our smart contact and other functions in the implementation. By comparing that with SBAC (Lyu *et al*., 2020), our proposal SBBICN reduced the overall required gas cost.

For example, to deploy the smart contract of the SBBICN, we need 0.0034736 ETH while in SBAC (Lyu *et al*., 2020) the number was 0.0130169 ETH. Therefore, our proposal SBBICN reduced the cost by 26%.

Table 2: Comparison between different proposals in terms of security aspects

Proposal	Protection against cache pollution attack	Protection against DDos attack	Protection against data replication	Data integrity	Access control
ABAC (Li *et al*., 2018)	YES	NO	NO	YES	Medium
Live (Li *et al*., 2014)	YES	Limited	NO	NO	Medium
BPDS (Fan *et al*., 2018)	NO	YES	NO	YES	Medium
BBAC (Di Francesco *et al*., 2017)	NO				
Fair Access (Ouaddah *et al*., 2017)	NO	NO	YES	High	
SEAF (Xue *et al*., 2018)	NO	NO	NO	YES	High
SBAC (Lyu *et al*., 2020)	YES	YES	NO	YES	Very High
BICN (Li *et al*., 2019)	YES	Limited	NO	YES	High
TrustCoin (Pan *et al*., 2020)	NO	NO	NO	YES	Medium
(Nour *et al*., 2021)	YES	Limited	NO	YES	High
SBBICN (our proposal)	YES	YES	YES	YES	Very High

Table 3: Gas cost in SBBICN

Function	Gas cost (ETH)
Deploy the smart contract	0.0034736
Register a publisher	0.0006
Search for a content	0.00071
Vote function	0.000167

Conclusion

The integration between ICN and blockchain technology has found its path rapidly. However, ICN also has several problems that hurt its efficiency and security. In this study, we described some of these key challenges and issues concerning blockchain technology over ICN. Next, we stated some recent related work that utilizes the features of blockchain to boost the security and performance aspects of ICN.

In addition, in this study, we proposed a secured Blockchain-based Information-Centric Network, called SBBICN. This framework uses the features of hashes, non-repudiation, non-tampering, and decentralization of blockchain in a public environment (Ethereum) and smart contract to secure ICN against cache pollution, data replication, DDos attacks and ameliorate its data integrity. We also presented the results and findings of our implementation to verify its effectiveness. Finally, we compared our SBBICN with other related schemes and showed the key differences between them in terms of security analysis and evaluation results.

In our future work, we plan to investigate the scalability issues of using blockchain in ICN due to the enormous number of transactions that are expected to be recorded in the blockchain in the real world and how we can develop its transaction speed which is considered low compared to other online transaction systems such as VISA (Logu *et al*., 2022).

Acknowledgment

We want to express our thanks and gratitude to all those who helped us throughout this study.

Author's Contributions

Abdelrahman Sheham Abdellah: Conducted experiments, data-analysis and contributed to the writing.

Sherif Saif: Reviewed experiments results and contributed to the concept review and to the paper writing.

Hesham E. ElDeeb: Supervision and review.

Emad Abd-Elrahman: Shared in the conceptual design of this work.

Mohamed Taher: Supervision and review.

Ethics

This article is original and previously unpublished and contains material in any journal. The corresponding author acknowledges that the work has been reviewed and approved by all other authors and that there are no ethical concerns.

References

Abdellah, A., Saif, S. M., ElDeeb, H. E., Abd-Elrahman, E., & Taher, M. (2020). A survey of using blockchain aspects in information centric networks. In International Conference on Advanced Intelligent Systems and Informatics, pp, 292-301. doi.org/10.1007/978-3-030-58669-0_27

Arora, D., Birla, M., Gupta, M., & Tiku, M. (2021). Decentralized Incognito Limpid e-Voting System. International Journal of Innovative Research in Computer Science and Technology (IJIRCST). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3848709

Asaf, K., Rehman, R. A., & Kim, B. S. (2020). Blockchain technology in named data networks: A detailed survey. Journal of Network and Computer Applications, 171, 102840. doi.org/10.1016/j.jnca.2020.102840

Behal, S., Kumar, K., & Sachdeva, M. (2018). D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events. Journal of Network and Computer Applications, 111, 49-63. doi.org/10.1016/j.jnca.2018.03.024

Bera, B., Das,A. K., & Sutrala, A. K. (2021). Private blockchain-based access control mechanism for unauthorized uav detection and mitigation in internet of drones environment. Computer Communications, 166, 91-109. doi.org/10.1016/j.comcom.2020.12.005

Berdik, D., Otoum, S., Schmidt, N., Porter, D., & Jararweh, Y. (2021). A survey on blockchain for information systems management and security. Information Processing and Management, 58(1), 102397. doi.org/10.1016/j.ipm.2020.102397

Braeken, A. (2020). Highly efficient symmetric key based authentication and key agreement protocol using keccak. Sensors, 20(8), 2160. doi.org/10.3390/s20082160

Chacko, J. A., Mayer, R., & Jacobsen, H. A. (2021, June). Why do my blockchain transactions fail? a study of hyperledger fabric. In Proceedings of the 2021 International Conference on Management of Data (pp. 221-234). https://dl.acm.org/doi/abs/10.1145/3448016.3452823

Chen, H., Pendleton, M., Njilla, L., & Xu, S. (2020). A survey on ethereum systems security: Vulnerabilities, attacks and defenses. ACM Computing Surveys (CSUR), 53(3), 1–43. doi.org/10.1145/3391195

Chen, S., Liu, X., Yan, J., Hu, G., & Shi, Y. (2021). Processes, benefits and challenges for adoption of blockchain technologies in food supply chains: A thematic analysis. Information Systems and e-Business Management, 19(3), 909-935. https://link.springer.com/article/10.1007/s10257-020-00467-3

Coelho, I. M., Coelho, V. N., Araujo, R. P., Qiang, W. Y., & Rhodes, B. D. (2020). Challenges of pbft-inspired consensus for blockchain and enhancements over neo dbft. Future Internet, 12(8), 129. doi.org/10.3390/fi12080129

Conti, M., Gangwal, A., Hassan, M., Lal, C., & Losiouk, E. (2020). The road ahead for networking: A survey on icn-ip coexistence solutions. IEEE Communications Surveys and Tutorials, 22(3), 2104-2129. doi.org/10.1109/COMST.2020.2994526

Conti, M., Hassan, M., & Lal, C. (2019). BlockAuth: BlockChain based distributed producer authentication in ICN. Computer Networks, 164, 106888. doi.org/10.1016/j.comnet.2019.106888

Cortez, D. M. A., Sison, A. M., & Medina, R. P. (2020). Cryptographic randomness test of the modified hashing function of sha256 to address length extension attack. In Proceedings of the 2020 8th International Conference on Communications and Broadband Networking, pp, 24-28. doi.org/10.1145/3390525.3390540

Di Francesco, M., D., Mori, P., & Ricci, L. (2017). Blockchain based access control. In Chen, L. Y. and Reiser, H. P., editors, Distributed Applications and Interoperable Systems, pages 206-220, Cham. Springer International Publishing. doi.org/10.3390/app12063204.

Din, I. U., Asmat, H., & Guizani, M. (2019). A review of information centric network-based internet of things: Communication architectures, design issues and research opportunities. Multimedia Tools and Applications, 78(21), 30241–30256. doi.org/10.1007/s11042–018–6943–z

Donmez, A., & Karaivanov, A. (2022). Transaction fee economics in the Ethereum blockchain. Economic Inquiry, 60(1), 265-292. doi.org/10.1111/ecin.13025

Dutta, N., Sarma, H. K. D., Jadeja, R., Delvadia, K., & Ghinea, G. (2021). Integrating Content Communication into Real-Life Applications. In Information Centric Networks (ICN) (pp. 169-194). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-030-46736-4_9

Eum, S., Nakauchi, K., Murata, M., Shoji, Y., & Nishinaga, N. (2012, August). CATT: Potential based routing with content caching for ICN. In Proceedings of the second edition of the ICN workshop on Information-centric networking (pp. 49-54). https://dl.acm.org/doi/abs/10.1145/2342488.2342500

Fan, K., Ren, Y., Wang, Y., Li, H., & Yang, Y. (2018). Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G. IET communications, 12(5), 527-532. doi.org/10.1049/iet-com.2017.0619

Fotiou, N. (2020). Information-Centric Networking (ICN). Future Internet, 12(2), 35. https://www.mdpi.com/1999-5903/12/2/35

Haider, S., Akhunzada, A., Mustafa, I., Patel, T. B., Fernandez, A., Choo, K. K. R., & Iqbal, J. (2020). A deep cnn ensemble framework for efficient ddos attack detection in software defined networks. Ieee Access, 8, 53972–53983. doi.org/10.1109/ACCESS.2020.2976908

Hu, T., Liu, X., Chen, T., Zhang, X., Huang, X., Niu, W., Lu, J., Zhou, K., & Liu, Y. (2021). Transactionbased classification and detection approach for ethereum smart contract. Information Processing & Management, 58(2), 102462. doi.org/10.1016/j.ipm.2020.102462

Li, B., Huang, D., Wang, Z., & Zhu, Y. (2016). Attribute-based access control for ICN naming scheme. IEEE Transactions on Dependable and Secure Computing, 15(2), 194-206. https://ieeexplore.ieee.org/abstract/document/7447763

Li, C. (2021, November). Gas Estimation and Optimization for Smart Contracts on Ethereum. In 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 1082-1086). IEEE. https://ieeexplore.ieee.org/abstract/document/9678932

Li, H., Wang, K., Miyazaki, T., Xu, C., Guo, S., & Sun, Y. (2019). Trust-enhanced content delivery in blockchainbased information-centric networking. IEEE Network, 33(5), 183–189. doi.org/10.1109/MNET.2019.1800299

Li, Q., Zhang, X., Zheng, Q., Sandhu, R., & Fu, X. (2014). LIVE: Lightweight integrity verification and content access control for named data networking. IEEE Transactions on Information Forensics and Security, 10(2), 308-320. https://ieeexplore.ieee.org/abstract/document/6942259

Logu, K., Devi, T., Deepa, N., & Gayathri, N. (2022). A Real-Time Monitoring Tool for Analyzing Ethereum Digital Currency in Global Business Transaction. In Blockchain Security in Cloud Computing (pp. 167-188). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-030-70501-5_8

Lugo, L., & Pedraza, C. (2020). Performance evaluation for the hash generation phase of a democratic blockchain. International Journal of Internet Technology and Secured Transactions, 10(3), 286–303. doi.org/10.1504/IJITST.2020.107076

Lyu, Q., Qi, Y., Zhang, X., Liu, H., Wang, Q., & Zheng, N. (2020). Sbac: A secure blockchain-based access control framework for information-centric networking. Journal of Network and Computer Applications, 149, 102444. doi.org/10.1016/j.jnca.2019.102444

Man, D., Mu, Y., Guo, J., Yang, W., Lv, J., & Wang, W. (2021). Cache Pollution Detection Method Based on GBDT in Information-Centric Network. Security and Communication Networks, 2021. https://www.hindawi.com/journals/scn/2021/6658066/

Martínez, S., Gérard, S., & Cabot, J. (2022). Efficient model similarity estimation with robust hashing. Software and Systems Modeling, 21(1), 337-361. https://link.springer.com/article/10.1007/s10270-021-00915-9

Negara, R. M., & Syambas, N. R. (2020). Caching and machine learning integration methods on named data network: a survey. In 2020 14th International Conference on Telecommunication Systems, Services and Applications (TSSA, pp, 1–6. doi.org/10.1109/TSSA51342.2020.9310811

Nour, B., Mastorakis, S., Ullah, R., & Stergiou, N. (2021). Information-centric networking in wireless environments: Security risks and challenges. IEEE Wireless Communications. doi.org/10.1109/MWC.001.2000245

Nour, B., Sharif, K., Li, F., Yang, S., Moungla, H., & Wang, Y. (2019). Icn publisher-subscriber models: Challenges and group-based communication. IEEE Network, 33(6), 156-163. doi.org/10.1109/MNET.2019.1800551

Ouaddah, A., Elkalam, A. A., & Ouahman, A. A. (2017). Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In Europe and MENA cooperation advances in information and communication technologies (pp. 523-533). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-319-46568-5_53

Pan, Q., Wu, J., Li, J., Yang, W., & Guan, Z. (2020). Blockchain and ai empowered trust-information-centric network for beyond 5g. IEEE Network, 34(6), 38–45. doi.org/10.1109/MNET.021.1900608

Rathod, U., Sonkar, M., & Chandavarkar, B. R. (2020, July). An Experimental Evaluation on the Dependency between One-Way Hash Functions and Salt. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE. https://ieeexplore.ieee.org/abstract/document/9225503

Roy, S., Morais, F. J. A., Salimitari, M., & Chatterjee, M. (2019). Cache attacks on blockchain based information centric networks: An experimental evaluation. In Proceedings of the 20th International Conference on Distributed Computing and Networking, pp, 134–142. doi.org/abs/10.1145/3288599.3288640

Saad, M., Njilla, L., Kamhoua, C., Kim, J., Nyang, D., & Mohaisen, A. (2019). Mempool optimization for defending against ddos attacks in pow-based blockchain systems. In 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp, 285-292. doi.org/10.1109/BLOC.2019.8751476

Sayeed, S., & Marco-Gisbert, H. (2019). Assessing blockchain consensus and security mechanisms against the 51% attack. Applied Sciences, 9(9), 1788. doi.org/10.3390/app9091788

Sayeed, S., Marco-Gisbert, H., & Caira, T. (2020). Smart contract: Attacks and protections. IEEE Access, 8, 24416-24427. doi.org/10.1109/ACCESS.2020.2970495

Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST), pages 1–8. doi.org/10.1109/CCST.2019.8888419

Singh, R., Tanwar, S., & Sharma, T. P. (2020). Utilization of blockchain for mitigating the distributed denial of service attacks. Security and Privacy, 3(3), e96. doi.org/10.1002/spy2.96

Sokolov, M. (2021). Security and Privacy Features Supported by Different Overlay-based ICN/IP Coexistence Architectures. https://repository.tudelft.nl/islandora/object/uuid:8c4d2e1c-ff59-4ff5-b700-93f0c80acd53

Taş, R., & Tanrıöver, Ö. Ö. (2019, October). Building a decentralized application on the Ethereum blockchain. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-4). IEEE. doi.org/10.1109/ISMSIT.2019.8932806

Vivar, A. L., Orozco, A. L. S., & Villalba, L. J. G. (2021). A security framework for ethereum smart contracts. Computer Communications, 172, 119–129. doi.org/10.1016/j.comcom.2021.03.008

Xue, K., Zhang, X., Xia, Q., Wei, D. S., Yue, H., & Wu, F. (2018, April). SEAF: A secure, efficient and accountable access control framework for information centric networking. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications (pp. 2213-2221). IEEE. https://ieeexplore.ieee.org/abstract/document/8486407

Zeng, Z., Li, Y., Cao, Y., Zhao, Y., Zhong, J., Sidorov, D., & Zeng, X. (2020). Blockchain technology for information security of the energy internet: Fundamentals, features, strategy and application. Energies, 13(4), 881. doi.org/10.3390/en13040881

Zhang, G., Xie, H., Yang, Z., Tao, X., & Liu, W. (2021). Bdkm: A blockchain-based secure deduplication scheme with reliable key management. Neural Processing Letters, pp, 1-18. doi.org/10.1007/s11063–021–10450–9

Zhou, K., Zhang, Y., Huang, P., Wang, H., Ji, Y., Cheng, B., & Liu, Y. (2020). Efficient SSD cache for cloud block storage via leveraging block reuse distances. IEEE Transactions on Parallel and Distributed Systems, 31(11), 2496–2509. doi.org/10.1109/TPDS.2020.2994075