electronics

MDPI

Article

BLOCIS: Blockchain-Based Cyber Threat Intelligence Sharing Framework for Sybil-Resistance

Seonghyeon Gong and Changhoon Lee *

Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea; gongsh@seoultech.ac.kr

* Correspondence: chlee@seoultech.ac.kr

Received: 3 January 2020; Accepted: 18 March 2020; Published: 21 March 2020

Abstract: The convergence of fifth-generation (5G) communication and the Internet-of-Things (IoT) has dramatically increased the diversity and complexity of the network. This change diversifies the attacker's attack vectors, increasing the impact and damage of cyber threats. Cyber threat intelligence (CTI) technology is a proof-based security system which responds to these advanced cyber threats proactively by analyzing and sharing security-related data. However, the performance of CTI systems can be significantly compromised by creating and disseminating improper security policies if an attacker intentionally injects malicious data into the system. In this paper, we propose a blockchain-based CTI framework that improves confidence in the source and content of the data and can quickly detect and eliminate inaccurate data for resistance to a Sybil attack. The proposed framework collects CTI by a procedure validated through smart contracts and stores information about the metainformation of data in a blockchain network. The proposed system ensures the validity and reliability of CTI data by ensuring traceability to the data source and proposes a system model that can efficiently operate and manage CTI data in compliance with the de facto standard. We present the simulation results to prove the effectiveness and Sybil-resistance of the proposed framework in terms of reliability and cost to attackers.

Keywords: cyber threat intelligence; blockchain; smart contract

1. Introduction

The rapid development of communication and data analysis technology has caused various paradigm changes in the area of networks. The commercialization of fifth-generation (5G) communication and the growth of the Internet-of-Things (IoT) have connected various devices to the network, and edge and cloud computing technologies have enabled high-level services such as smart cities and SCADA networks [1]. These changes have dramatically increased the size and diversity of the entire network, creating a variety of added value along with large amounts of data collected from various sources [2].

However, the increase in connectivity and diversity among devices constituting the network have caused various problems in terms of information security [3]. The variety of networks has increased the types and numbers of vulnerabilities, and this has resulted in the expansion of attackers' attack vectors [4]. Attackers can use advanced attack vectors to perform more intelligent and targeted attacks. In particular, the incidence of threats, such as the advanced persistent threat (APT), which carry out long-term attacks for specific purposes, is continuously increasing [5]. These advanced threats continuously collect information about the specific targets over a long time and use targeted attack techniques that exploit various vulnerabilities to maximize the ability to attack. This type of attack is more difficult to detect on other nodes and takes much more time to determine if a breach has occurred [6]. In addition, as a result of the diversity of the network, many new vulnerabilities can

emerge, resulting in zero-day attacks using these unknown vulnerabilities. Zero-day [7] attacks are usually severe because they can cause lasting damage until security patches become available. It is also challenging to detect a zero-day attack because it uses unknown attack patterns [8].

Existing security and incident response systems, represented by firewalls, intrusion detection/prevention systems (IDS/IPS), security information and event management (SIEM) systems, are not sufficient to respond to unknown attack patterns. To cope with and predict the advanced threats that use new attack vectors and patterns, large amounts of data are essential for in-depth analysis, such as machine learning or deep learning. However, because the types, techniques, and victims of the attacks are quite different, the types of observed data are very diverse. In particular, it is challenging to collect a large amount of security-related data in a feasible form. Thus, there is a need to integrate differentiated security systems and share threat-related information in a usable form to raise the level of understanding of cyber threats and establish active and effective countermeasures. The cyber threat intelligence (CTI) system is a threat analysis and information sharing system for improving the understanding of cyber threats and proactively responding to them. CTI systems enhance the understanding of cyber threats by reorganizing and analyzing threat-related data into a formalized form. Additionally, the core of the CTI system is to maximize the threat response capability of each node by sharing information. This approach enables profiling of attack types and patterns, attackers, and attack groups, thereby predicting potential threats and responding proactively.

However, CTI systems also face the challenge of collecting the amount of data required to analyze and share. To collect a large amount of data, not only an internal data collector but also open source intelligence (OSINT) and various data-collection channels are additionally used. However, the data collected from such sources may be inaccurate or malicious. Because the CTI system forms reputation information for a specific network node, an attacker can perform a Sybil attack that spreads a large number of malicious data to isolate a specific node and undermine the availability of the network. Thus, resistance to Sybil attacks is a security requirement that must be considered in the operation of the CTI system.

This study proposes a blockchain-based open CTI framework that can verify the validity of data by giving traceability, integrity, and Sybil-resistance. The proposed framework consists of contributors which collect and share threat-related data, consumers which consume such data, and feeds that provide CTI data sharing services. The proposed framework allows data collection through contributors to maximize the ability to collect threat-related data, while at the same time providing a mechanism to prevent Sybil attacks from malicious contributors. An attacker may perform an attack that damages reputation information of a specific node using malicious contributors and miners. The proposed framework includes a mechanism to validate the data provided by contributors to prevent the continuous distribution of data by malicious contributors. The data verification performed by the CTI feed degrades the malicious contributor's data dissemination capability by evaluating the data contributor's reliability. The framework also increases the mining costs of malicious miners by undermining the ability of malicious contributors to lose their deposits. This mechanism allows the CTI system to block the malicious data injection automatically.

This paper proposes a blockchain-based open CTI framework for collecting reliable data from various channels and proposes a design method to implement the framework. Section 2 introduces the related works and security considerations of the CTI system, and Section 3 describes the system model. Section 4 illustrates the proposed framework in detail for each layer, and we propose the detailed implementations of our proposal in Section 5. In Section 6, the simulation results of the proposed framework are presented in terms of the proposed security considerations, and Section 7 concludes this research.

2. Related Work

In this section, we discuss the background, related research, and security considerations related to CTI technology.

2.1. Background

In this section, we describe the basic concepts and principles of CTI and the characteristics of Sybil attacks that target CTI systems.

2.1.1. CTI System

The CTI system is an evidence-based intelligence threat detection and prevention system [9]. The final goal of CTI is to have the capability of a preemptive response to cyber threats, such as the advanced persistent threat (APT) and zero-day attacks, and to profile the attackers and groups of attackers. The CTI system collects threat-related data from various channels, analyzes, and shares the information with other systems. The CTI system analyzes network logs, system logs, firewall logs, traffic, reputation information of network resources, and information collected from a security information and event management (SIEM). The CTI system extracts information such as attack patterns, identifiers, malware, attackers, and tactic-technique-procedures (TTP) from various types of data, and expresses them as entities to analyze the association between them. Structured threat information expression (STIX) [10] is the most commonly used CTI data expression language, and TAXII is the data communication protocol for exchanging data expressed as STIX. The analyzed CTI data is expressed in the STIX language and then interchanged through the TAXII protocol [10]. Fast and efficient sharing of data is essential to mitigate cyber threats proactively. If the malicious behavior of the attacker is observed, the CTI system generates information associated with cyberthreats by combining collected and preidentified information. This information includes the type and procedure of attack and the course of action. By sharing this information with other nodes participating in the CTI system, the information on the cyber threat can be spread quickly to other nodes. Each node establishes and updates its security policy using this information. The CTI system also performs profiling of attackers and groups of attackers to thwart zero-day attacks. By stereotyping the behavior patterns of attackers and malware, the system can predict future patterns of attacks and quickly establish countermeasures against them. Figure 1 shows the system model of the traditional CTI system.

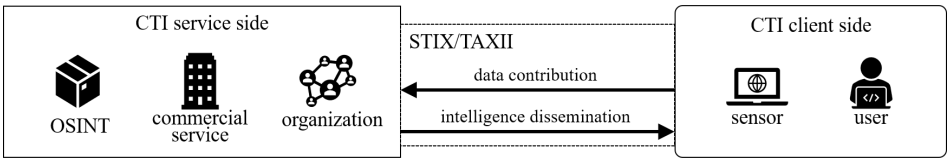

Figure 1. The system model of traditional cyber threat intelligence system.

2.1.2. Reputation Information

Reputation is one of the representative methods to perform practical functions using CTI. Reputation information allows you to determine the malicious behavior of a network node based on a unique identifier (such as IP, domain, hash). Nodes that detect malicious attacks or threats generate reputation information about the source of the attack and share it with other nodes. CTI system using reputation information can easily convert the received reputation information into a Snort rule or Yara rule to form a practical security rule [11]. Reputation information is, therefore, one of the most feasible and efficient applications of the CTI system.

2.1.3. Sybil Attack on CTI System

A security policy, which is generated based on reputation information in the CTI system, performs the security function by blocking the resource on the network. If incorrect data are input to the CTI system, the system generates false reputation information, which allows innocent nodes to be blocked from the network. Using this, an attacker can perform a series of Sybil attacks that produce malicious CTI data to manipulate the data so that the CTI system generates false reputation information.

As the first step in a Sybil attack, an attacker can inject malicious data through a naive data-collection point. For example, *ThreatCrowd.org* [12], one of the well-known OSINT CTI services, uses voting results as one of the indicators of reputation information for a particular domain. If CTI service users detect malicious activity in a particular domain, they can vote to notify the CTI service that the domain is malicious. Some CTI services, including *ThreatCrowd.org*, have no restrictions on who can vote, and an attacker can exploit these vulnerabilities to compromise the domain's reputation, as shown in Figure 2. The left screen of Figure 2 shows the query result of the reputation on the 'seoultech.ac.kr' domain. An attacker can contribute malicious data to compromise the reputation of target domain by impersonating multiple nodes or users. The right screen of Figure 2 shows the results of an experimental attack that lowered the reputation of the 'seoultech.ac.kr' domain after spoofing the IP using the tor browser.

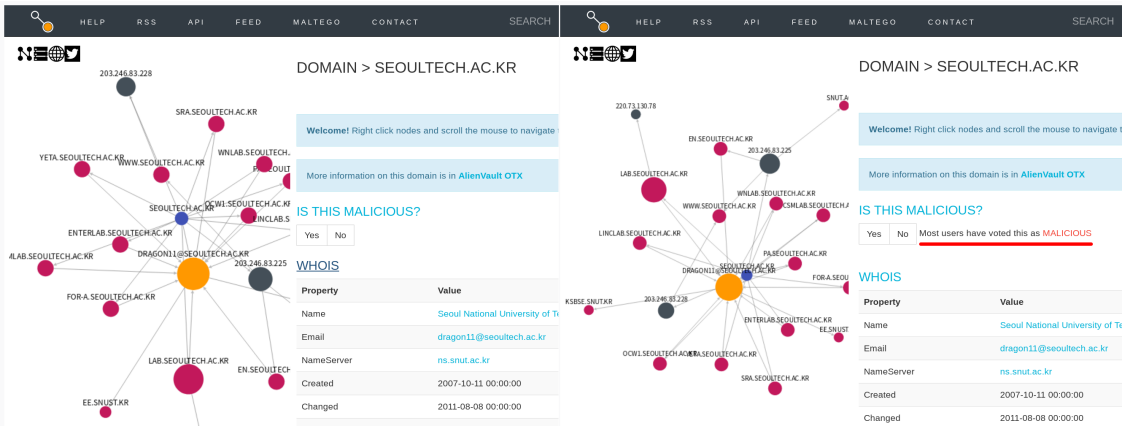

Figure 2. Change in a reputation of the 'seoultech.ac.kr' domain before (left) and after (right) a Sybil attack in the OSINT CTI service (*threatcrowd.org* [12]). The right screen shows that an attacker has created a bad reputation (underlined in red) for that domain through a Sybil attack.

OSINT CTI services cross-reference each other's data to collect data effectively. For example, data from ThreatCrowd.org is also used by other CTI services, such as *ThreatMiner.org* [13], and other OSINT data-collection and analysis tools, such as *Maltego* [14]. This approach to efficiently collect massive amounts of data can be an effective means to spread a malicious reputation. An attacker can use these data propagation paths in a Sybil attack by using a variety of data contribution paths and using a proper cost to impersonate a valid user.

2.2. State of the Art

CTI systems formalize and classify cyber threat patterns to increase the understanding of cyber threats. In [15], the types of CTI data in terms of data sharing are categorized. The CTI system can use the kill-chain model [16] as a threat response technique that formalizes cyber threat stages and uses optimal countermeasures for each attack phase. In [17–19], the types of CTI data based on the kill-chain model are classified. Furthermore, to identify cyber threats from the data, [20] proposed a threat analysis method based on the data. In [9], they propose a classification model of technologies for CTI data exchange.

The issue of how to represent threat-related data is an essential issue for CTI's practical use. Indicators of compromise (IoC) is the main index for CTI systems to represent cyber threats. Any information that can identify the target on the network and system, such as the hash value of the malware, the file name, the IP address used to distribute the malware, the domain name, and the URL, can be used as an indicator of IoC. STIX [10] is a threat-related data representation language and is currently used as a de facto standard. In [21], they propose an extension of the language model to improve STIX's ability to express threat data. In [22], they analyzed the data exchange format using ontology to enhance the capabilities of cyber threat information sharing standard technology.

Data-collection methods and collection channels are also important factors in operating a CTI system. In [23–27], they studied how to collect, refine, and operate CTI data from the Darknet and hacker forums. In [28,29], they propose a methodology for collecting CTI data from public source information and sharing them. Since the data collected from these sources exist in natural language or a similar form, not in a standardized form, a method of extracting the context of information from this type of data is required. In [29–33], they propose methods to collect and analyze CTI data using natural language processing and semantic analysis techniques.

When the data is collected from various data channels, the validity and reliability of the data must be considered. In our previous study [34], we proposed a model for determining the reliability of data collected from open source intelligence (OSINT). In [35], they show that CTI data analysis using machine learning technology may be vulnerable to data poisoning attacks. CTI systems also use a data-sharing framework for efficient operation. In [36–39], they propose a cyber threat information sharing framework, and [40,41] propose a CTI sharing framework through the blockchain.

2.3. Security Consideration

This subsection describes essential security considerations to design a CTI framework: traceability, Sybil-resistance, and privacy.

2.3.1. Traceability

Traceability, as an essential element to verify the context and validity of the data, means the tracing of data from generation to the analysis process and applications [34]. For the traceability of CTI data, metainformation of data-collection channels, environmental characteristics, and threat information should be configured together with enough strength of integrity fo the data. Metainformation about a collection channel could be a quantitative scale indicating the reliability of the channel. This composition prevents an attacker from continuously disseminating malicious data and allows tracking of data to detect abnormal behavior. In addition, it is possible to determine the importance of the information through the environmental characteristics of the data-collection channel, thereby determining the priority of the data processing process. For example, the context and importance of observed data about malware between the terminal node and the central server are significantly different. Therefore, CTI information should provide the context for cyber threats, including the environmental characteristics of the data channel.

2.3.2. Sybil-Resistance

Sybil attacks are attacks where an attacker configures multiple nodes for a specific purpose, disguising the attacker's action from actions of the crowd. This attack, which can occur in networks such as social network services (SNS) or blockchains, can be fatal in an environment where the identity and owner of terminal nodes are not identifiable. In particular, networks that use a reputation for specific nodes are more vulnerable to Sybil attacks. An attacker can cause the node to be blocked from the network by disseminating the target node as dangerous. If the attacker's capabilities are sufficient during the CTI data sharing process, the node can be completely excluded from the network, which could compromise the availability of the entire network. Thus, the CTI system requires a fundamental defense mechanism against Sybil attacks.

2.3.3. Privacy

The biggest problem of operating a CTI system is the privacy issue [42]. The CTI system aims to increase the understanding of cyber threats by sharing information about the threat and IoC with other nodes. When the identification of a specific object is exposed during this information sharing process, the actions involved in information sharing may cause a deterioration of the source's reputation. Therefore, there is a need to apply deidentification techniques in the process of sharing data. However, deidentification means the partial loss of data, and this could reduce the usefulness of the information.

If the quality of information is decreased in the course of preserving privacy, the performance of the CTI system deceases. Thus, CTI systems must be able to meet the trade-off between privacy and the usefulness of the information in the process of data sharing.

3. System Model

In this section, we describe the system model of blockchain-based CTI data-collection and sharing framework. The proposed framework consists of nodes and entities with multiple roles. Table 1 describes the notations used in the proposed framework.

Table 1. Descriptions for notations used in the proposed framework.

Notation	Description
$F = \{F_1, \ldots, F_i, \ldots, F_n\}$	Set of CTI Feeds, $i = 1, \ldots, n$
$c = \{c_1, \ldots, c_j, \ldots, c_m\}$	Set of contributors, $j = 1, \ldots, m$
$s = \{s_1, \ldots, s_k, \ldots, s_l\}$	Set of consumers, $k = 1, \ldots, l$
β_{c_j}	Reliability value of contributor c_j
α_{c_j}	Observed threat-related data from contributor c_j
ϵ_{c_j}	Deposit cost c_j's data contribution
$\pi_{F_i}(\alpha_{c_j}, \beta_{c_j}, \epsilon_{c_j})$	Evaluation function of F_i on reported data from c_j
q	Network resource such as IP, domain, hash value of malware
$\theta(q, p)$	CTI data query using specific resource q and cost p
$\lambda(q)$	Result on CTI data query for resource q
ψ_{F_i}	Consumer subscription list of CTI feed F_i
$E_k(d)$, $D_k(d)$	encrypted and decrypted data d using key k
$S_k(d)$, $V_k(d)$	digital signing and verification using key k and data d
Pr_{c_i}, Pu_{c_i}	Private key and Public key of c_i

3.1. Blockchain-Based Cyber Threat Intelligence System Model

As shown in Figure 3, the blockchain-based CTI system is composed of five main entities and their interactions: feeds, contributor, consumer, miner, and blockchain network.

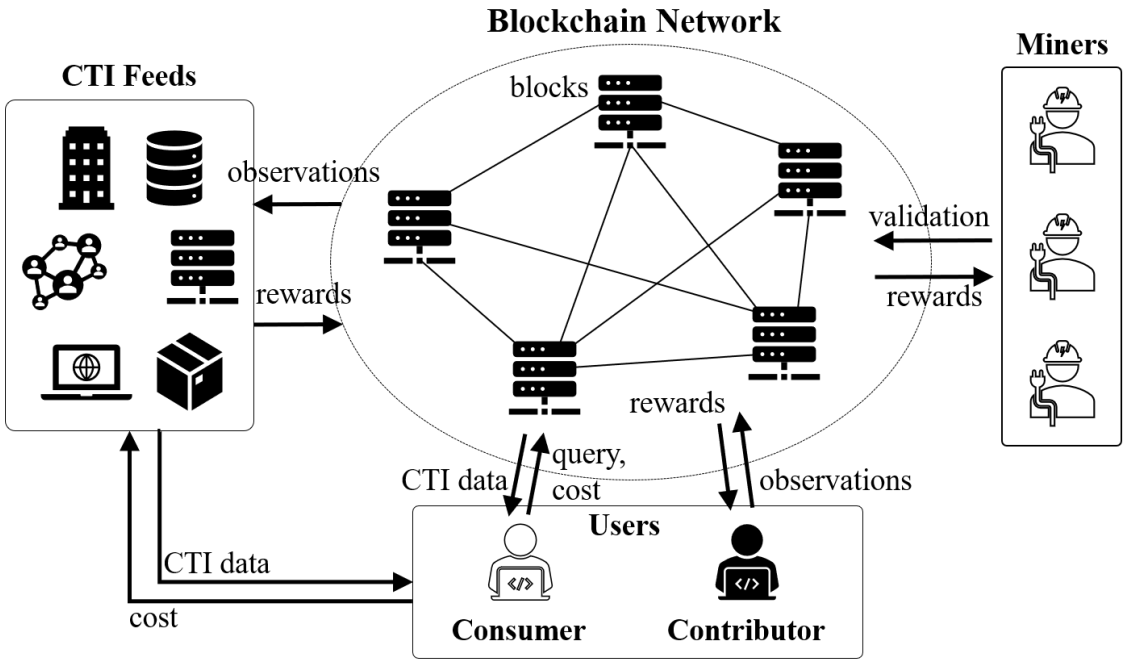

Figure 3. System model of blockchain-based cyber threat intelligence system.

Feeds: defined as $F = \{F_1, \ldots, F_i, \ldots, F_n\}$. Feeds collect security-related data from users and their data-collection channels, reconstruct it into an actionable CTI information. Each CTI feed defines an evaluation and reward method π_{F_i} for the observed data to evaluate and encourage users' contributions. In addition, CTI feeds have a user list ψ_{F_i} to propagate the analyzed CTI data when a critical threat is reported.

Contributor: defined as $c = \{c_1, \ldots, c_j, \ldots, c_m\}$. A contributor is a user who participates in the CTI system and shares the threat-related data observed from their internal systems, such as firewall or IDS, with the CTI system. Contributors transmit the observed threat-related log data α_{c_j} through the smart contract to the blockchain network of the CTI system and pay a deposit ϵ_{c_j}. The data reported by the contributor is evaluated by the function π_{F_i} of each CTI feed F_i, and if the data is determined to be useful to the feed, the feed rewards the contributor. Each contributor has an individual reliability value β_{c_j}, which is adjusted by the reward provided by the CTI feeds.

Consumer: defined as $s = \{s_1, \ldots, s_k, \ldots, s_l\}$. A consumer is a user who participates in the CTI system and is the primary entity that consumes CTI data. Consumers can query specific CTI data q to blockchain networks by consuming cost p_{s_k}, and each feed provides CTI data $\lambda(q)$ corresponding to consumer's requests and obtains p_{s_k}. Also, by registering themselves in the CTI feed's user list ψ_{F_i}, consumers are periodically provided with information related to cyber threats when the CTI Feed detects a critical threat.

Blockchain Network: A blockchain network performs the core functions of delivering and storing data in the CTI system. Smart contracts implemented on the blockchain allow users to communicate and share data through reliable procedures. CTI feeds can also operate CTI data through a set of procedures implemented as smart contracts. Blockchain-based CTI frameworks can provide a high level of integrity and traceability for data, thus facilitating the assessment of the validity of the data. CTI systems must encrypt the data for privacy while storing them in a block.

Miner: Using a consensus mechanism of the blockchain network, miners store the data request, contribution, and reward transactions in the block. In blockchain-based CTI framework, cryptocurrencies mined by miners are used as a means to use the CTI system, and all users can play both roles as a contributor and as a miner simultaneously. Users spend the cryptocurrency as a cost for requesting CTI data, or they get the cryptocurrency as a reward for data contribution. Each cost on data request and reward on data contribution has a different amount based on the importance of data.

3.2. Threat Model

In this section, we describe the threat models of existing Sybil attacks on the CTI system and the threat models that can arise in the operation of blockchain-based CTI systems.

Malicious Contributor: Attackers can attack CTI systems through malicious contributors by reporting false data. An attacker can pay a high deposit so that the malicious data reported by the attacker are stored first in the block. In addition, by reporting a large number of redundant data, the attacker can increase the probability of malicious data stored in the block. By creating a relation between the target node and malware or malicious behavior, the attacker can reduce the reputation of the target node. When this malicious information is input to the CTI system, a security policy may be generated to block access to the target node. In this case, the attacker aims to inject as much malicious data α as possible into the system at the least cost e. At this point, the attacker acts to maintain a high level of reliability β to increase the probability of malicious data being injected into the system.

Malicious Miners: An attacker can act as both a malicious contributor and a malicious attacker to efficiently inject malicious data into the CTI system. The attacker aims to inject as much malicious data α as possible into the system. At the same time, since continuous malicious data injection requires the deposit ϵ, the attacker aims to recover as much deposit e as possible in the mining process.

The proposed framework should be able to block such malicious user behavior systematically.

4. BLOCIS: Blockchain-Based Open Cyber Threat Intelligence System

The traceability of the data and the validity of the data and its contributors must be covered to build a reliable open CTI system. We used blockchain to meet these two security considerations. The basic structure and function of the blockchain ensure the traceability of CTI data. Additionally, validity assessments of data and data contributors can be carried out through the smart contract on the blockchain, and the results can also be shared transparently through the smart contract. Furthermore, we covered the privacy issues that may arise when sharing data on the blockchain. In this section, we illustrate the proposed BLOCIS (blockchain-based open cyber threat intelligence system) framework.

4.1. Architecture of BLOCIS

The proposed BLOCIS is a blockchain-based open cyber threat intelligence sharing framework that is resistant to Sybil attacks. BLOCIS classifies the layers according to the environment in which the actual data are collected and operated for data-interchangeability. This section describes each layer of the BLOCIS framework. Figure 4 shows the architecture of proposed framework. On the basis of the system model mentioned in Chapter 3, BLOCIS consists of three layers: the user layer, blockchain network layer, and feed layer.

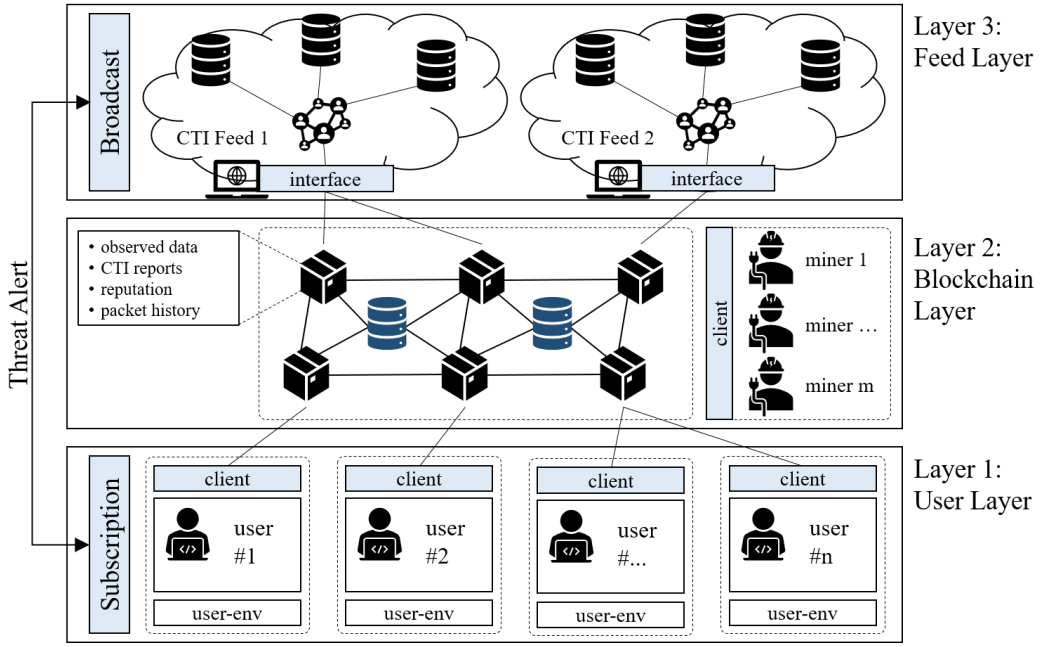

Figure 4. Architecture of the blockchain-based open cyber threat intelligence system (BLOCIS).

User Layer: At the user level, contributors and consumers act as actual users. BLOCIS is an open CTI system that collects data from multiple data sources, including the user's environment. Users can have internal security systems such as firewalls, IDS, IPS, honeypots, and get benefits from sharing the threat-related data observed with the BLOCIS framework. Contributors also need to convert threat-related data that they collected and observe into standard specifications such as STIX. To end this, the client needs a data parser that can collect and preprocess the data appropriate for the user's environment. The users can build this parser as an extension of the security system of their environment. Consumers request specific threat-related data and periodically receive CTI reports from feeds through the blockchain network.

Blockchain Network Layer: The BLOCIS framework uses blockchain technology for efficient management and sharing of CTI data. The blockchain network layer is composed of blockchain storage nodes and miner nodes. When observation data are reported from the user layer or when

a user requests specific CTI information, the information is transmitted to the blockchain network. Miner nodes obtain a reward by checking the user's requests and get the deposit by processing them. All processes for querying user requests, and reporting and receiving data from feeds are conducted through smart contracts. The blockchain network ensures the integrity and traceability of the data running in the CTI system by recording data reported from users, information about each contributor, data-collection procedures, and the history between users and feeds.

Feed Layer: In the feed layer, various feeds provide CTI services. These are individual web or application services that provide CTI information to consumers included in their user list. Each feed serves a different purpose and does not need to utilize all of the CTI data reported from the various channels. Each feed has its data evaluation function, which selectively collects data reported to the blockchain network. The feed also determines the validity of the collected data to adjust reputation of the contributors. The feed generates an alert about the data contributor if the data obtained from the blockchain network are determined as being malicious. This alert information lowers the reputation of the contributor. By doing this, the user's expected result on an evaluation function of feeds decreases, and this makes a malicious user's data contribution more difficult.

4.2. CTI Data Contribution and Sharing Process

In this section, we describe the CTI data sharing process of the proposed BLOCIS framework. Inspired by [40], we use a cryptocurrency as a token to represent the reliability and solvency of users. The data sharing process of BLOCIS consists of the five steps for CTI data sharing and propagation.

Step 1: In the first step, users such as contributors or consumers register their account and address to the blockchain network. The blockchain network gives them tokens for the solvency of CTI data requests and reporting. The blockchain network adjusts the initial reliability of each contributor and transmits an encryption key to be used in the data contribution and sharing processes to the users.

Step 2: The user (contributor) converts observed data into STIX-based CTI data and transmits it to the blockchain network. The contributor sets target feeds to provide their data and executes a smart contract to contribute data to that feed. This smart contract receives information about data contributors, target feeds, CTI data, and deposits as input.

Step 3: The smart contract first performs data validation using reported data. Validation is a prefiltering operation to detect unstructured data or noise data and perform verification of the data format and integrity. If the reported data are valid, the smart contract performs the following steps. If the entered data are not valid, the smart contract adjusts the reliability of the contributor who contributed the abnormal data using the penalty term p.

Step 4: The smart contract executes the data evaluation function of the target CTI feeds on valid input. This function evaluates the validity and importance of the reported CTI data according to the feed's internal policies and determines whether to accept the data based on the results of the evaluation. If the results of the evaluation function are higher than the feed's criteria, the feed stores the reported CTI data in an internal database, and gives rewards to the contributor by offering cryptocurrency and increasing the contributor's reliability. If the result of the evaluation function is lower than the evaluation criteria, the smart contract ignores the data and adjusts the contributor's reliability according to a predetermined policy.

Step 5: After the evaluation of feeds, each feed analyzes new data with their internal strategies, policies, and database to find a substantial threat that the data represents. If a critical threat is expected through the accumulation of CTI data, the feed broadcasts the report on the expected threat to the users which are in the feed's list. The broadcast process is delivered directly to the user through a separate and reliable communication channel, and log information about the broadcasted data is encrypted and stored on the blockchain.

5. Implementation of BLOCIS

In this section, we illustrate the detailed scheme and procedures of BLOCIS using the pseudocode of the algorithm and smart contracts.

5.1. Environments

For the implementation of the proposed BLOSIS framework, we used the Ethereum framework [43]. The functions of BLOCIS are implemented using Solidity language [44]. We used the Ganache framework for the blockchain environment and the Truffle framework as an integrated development environment to write and compile the smart contracts. In addition, we used Metamask as a wallet interface for the user.

5.2. Smart Contract for CTI Data Sharing

In this subsection, we illustrate the detailed content of the smart contract that composes the proposed framework. The smart contracts are designed based on the interactions between each node in the framework. To implement the interactions, we devised three smart contracts: the user management contract (UMC), the data report contract (DRC), and the alert contract (AC).

5.2.1. User Management Contract (UMC)

As the open network, the BLOCIS includes various types of client users, and each user is classified into two roles: consumer or contributor. To manage and adjust the action and behavior of users, each user should enroll their identity to the blockchain network, where the address of the user is the only way to identify them. This address helps the user to keep their privacy. Each user enrolls their address into the blockchain network, and into the broadcast list of CTI feeds. The broadcast list is used to disseminate the alerts when the critical threat is observed. Furthermore, when a contributor gets into the blockchain network, the proposed framework gives them the initial reliability. This reliability is used as a reward for data contribution. Algorithm 1 shows the procedure of UMC. Algorithm 1 conducts initialization of user reliability and key exchange between users and feeds using a key-exchange scheme based on a public-key cryptographic scheme such as RSAES-OAEP [45]. Each user and feed have their public and private key pair, and these are used to verify each other and to share a secret key for encrypted data sharing.

Algorithm 1: Pseudocode of User Management Contract (UMC)

Data: $addr_{c_i}$: address of user(consumer or customer) c_i
Result: Initializing user information
$\beta c_i \leftarrow i$
for F_i *that user choose* **do**
  set k ; // generating secret key
  $KEM \leftarrow E_{Pu_{F_i}}(k)||S_{Pr_{c_i}}(k)$; // generating key-exchange message
  c_i send KEM to F_i
  calculate $V_{Pu_{c_i}}(S_{Pr_{c_i}}(k))$; // verifying signature
  if *valid signing flag* **then**
    $k \leftarrow D_{Pr_{F_i}}(E_{Pu_{F_i}}(k))$; // decrypting secret key
    $\psi_{F_i} \leftarrow (addr_{c_i}, k)$
  else
    return false
  end
end

5.2.2. Data Report Contract (*DRC*)

CTI data collected from the end node (user) is encrypted using a preshared secret key k, and then reported through a smart contract. However, in the open data-collection channel, the validity of data is one of the significant issues. To evaluate the validity of reported threat-related data, each feed F_i assesses the validity of reported data using their evaluation function π_{F_i} based on their internal policy.

If the result of the evaluation function exceeds the threshold defined by each feed, the feed sends the transaction that includes reward on the contribution of the user. The reward on the contribution adjusts the reliability of the contributor and gives the token as an incentive. If the result of the evaluation function is lower than the threshold, the feed ignores the contributions and adjusts the reliability of the contributor with a penalty term p. This penalty term mitigates the impact and damage from invalid or malicious contributions and can be adjusted considering the network circumstances, such as number of reported data, users, and contributors. Algorithm 2 shows the data reporting process and rewards for the user.

In the process of contributing data, the size of the deposit is usually set smaller than the amount of compensation for the contribution. As a result, users can accumulate their assets (currency) through continuous data contributions. This accumulated asset represents the user's activity. An enormous asset means that the user has contributed a lot of high-quality data, which can be another indicator of the reliability of the user.

Futhermore, the user's deposit is entered as a parameter in the data evaluation function π_F of each CTI feed. This parameter can make the user adjust the amount of deposit to adjust the π_F output. By spending high deposits in data *DRC*, users can increase the probability of successful reporting and the priority of the contribution. In other words, users can use their assets as their ability to contribute data.

CTI feeds generally have dedicated purposes, some of which may require critical, sensitive, and urgent data. CTI feeds that only need reliable data can filter out users who contribute the CTI data by setting a high minimum deposit amount. In addition, by intentionally setting high deposits and low compensation (even less than deposits), only voluntary data contributions can be allowed. This operation strategy of the CTI feed can improve the reliability of the reported data, and users can use the reliable CTI feed by consuming assets.

Algorithm 2: Pseudocode of Data Report Contract (*DRC*)

Data: k: preshared secret key between user and CTI feed
$E_k(\alpha_{c_i})$: encrypted CTI data of STIX format collected by c_i
β_{c_i}: reliability of contributor c_i
ϵ_{c_i}: deposit cost of c_i to report data α_{c_i}
$\bar{\epsilon}$: mean of total deposit costs
τ_j: threshold of evaluation function π_{F_j}
f: weight value for change ratio of reliability
p: weight value for penalty of invalid data

Result: β_{c_i}: updated reliability of user c_i
ρ: reward of feed F_j on the data contribution α_{c_i}

for *each feed* F_j *in* F **do**
  $tmp \leftarrow D_k(E_k(\alpha_{c_i}))$; // decrypting user data
  if α_{c_i} *is valid* ; // data format validation
  then
    $\sigma \leftarrow \pi_{F_j}(tmp, \beta_{c_i}, \epsilon_{c_i})$
    if $\sigma > \tau_{\pi_{F_j}}$ **then**
      $\varphi \leftarrow \frac{1}{(1-\tau_j)^2} \times (\sigma - \tau_j)^2$
    else
      $\varphi \leftarrow \frac{-1}{\tau_j^2} \times (\sigma - \tau_j)^2$
    end
    $n(\epsilon) \leftarrow \epsilon_{c_i} / \bar{\epsilon}$; // normalizing deposit
    if $n(\epsilon) > 0$ **then**
      $\beta_{c_i} \leftarrow max(min(\beta_{c_i} + n(\epsilon) \times f, 1), 0)$
      $\rho \leftarrow max(0, \epsilon \times (n(\epsilon) + \bar{\epsilon}))$
    else
      $\beta_{c_i} \leftarrow max(min(\beta_{c_i} + n(\epsilon) \times f \times p, 1), 0)$
      $\rho \leftarrow 0$
    end
  else
    $\beta_{c_i} \leftarrow p \times \beta_{c_i}$
    $\rho \leftarrow 0$
  end
end
return (β_{c_i}, ρ)

5.2.3. Alert Contract (*AC*)

The alert contract disseminates the threat-related alerts to the consumers. CTI feeds in the BLOCIS framework continuously analyze reported data to make profiles on the cyber threat. Each feed has its analysis mechanism for a specific purpose. If a feed F_i deduces a cyber threat from the result of data analysis, it alerts the information to the users in their broadcast list ψ_{F_i}. Metadata about these alerts are encrypted with the secret key and stored in the blockchain networks, and the full contents of encrypted CTI information are transmitted to the users of the user layer through the blockchain or other data links (Algorithm 3). Figure 4 shows the detailed procedures of alert contracts.

Algorithm 3: Pseudocode of Alert Contract (AC)

Data: α: Reported threat-related data
ψ_{F_i}: User list for alert broadcast
$E_k(\alpha_{c_i})$: encrypted CTI data of STIX format collected by c_i
Result: l: meta-information of alerts
λ: Analyzed CTI information related with cyber threats
q: list of IoC resources

deduce (λ, q) from α ; `// CTI and IoC deduction from data`
set l ; `// generating meta-information of alert`
send l to BLOCIS network
for c_j *in* ψ_{F_i} **do**
  send $E_k(l, \lambda, q)$ to c_j
end
return $E_k(l, \lambda, q)$

6. Experiment and Result

This section explains the experiment and simulation results that we performed to prove the efficiency of the proposed framework. The primary purpose of the proposed framework is to have the ability to resist the threat model of Sybil attacks mentioned in Section 3.2 Therefore, in this experiment, we evaluated how the attacker's attack ability in the CTI system could be compromised.

To evaluate the attacker's ability to attack, we first defined the attacker's ability to attack. The attacker's ability to attack Sybil is related to the amount of cryptocurrency the attacker has and the attacker's reliability value. Since the amount of cryptocurrency possessed by an attacker can be used to increase the probability of malicious contribution, it represents the risk of a Sybil attack from a short-term perspective. The attacker's reliability is related not only to the probability of malicious contribution but also to a long-term attack. If an attacker can maintain a high degree of reliability while simultaneously conducting a malicious contribution within the CTI system, the attacker can continue to contribute malicious data, thereby compromising the reliability of the entire CTI system. Therefore, in this experiment, we considered the attacker's ability to attack Sybil as the amount of cryptocurrency possessed by the attacker and the reliability of the attacker's node.

This experiment shows how the proposed framework can reduce the attacker's attack ability through smart contracts implemented on the blockchain network.

6.1. Normal and Malicious Contributor

In the experiment, we simulated two types of contributors: a normal contributor and malicious contributor. A CTI system that uses open sources as the data-collection channel has the risk of noise data. Noise data are defined as useless or unusable data or data intentionally modified by an attacker [34]. This definition of noise data means that the malicious contribution through noise data could be conducted not only by the attacker, but also occasionally by the normal user. Furthermore, determining whether a threat-related data are malicious (i.e., fabricated by an attacker) is different for different CTI systems' operational algorithms and policies. Evaluating the accuracy of data requires additional postanalysis of data, such as cross-validation. Thus, we simulated the behaviors of normal and malicious contributors using the possibility of the noise-data contribution. To simulate the possibility of the noise data, we used the normal distribution. Each contributor in the experiments has a mean and standard deviation for the possibility of noise-data contribution. In addition, in our experiment, we supposed that the malicious contributor is not naive. Thus, the malicious contributor imitates the behavior of the normal contributors. We set up a malicious contribution cycle t for each attacker, and the attacker attempts the malicious contribution after t normal contribution cycles. These periodic Sybil attacks retain the reputation of malicious contributors during their attacks. In the experiment, we set various malicious contributors with different Sybil attack cycles.

6.2. Reliability of Malicious Contributor

Each contributor has a capacity of $\delta = (m, s.d.)$ on noise data. m is the mean, and $s.d.$ is the standard deviation of possibility for noise-data contribution. To show the trends of reliability for each contributor, we simulated three contributors with different possibilities of noise-data contribution: normal contributor $n : \delta_n = (m = 0.8, s.d. = 0.1)$, malicious contributor $m_1 : \delta_{m_1} = (m = 0.8, s.d. = 0.1)$ with $t_{m_1} = 10$, and recklessly malicious contributor $m_2 : \delta_{m_2} = (m = 0.4, s.d. = 0.1)$ with $t_{m_2} = 2$. Figure 5 shows the trends in the reliability of each contributor. The malicious contributor m_1 has the same capacity as the normal contributor and performs the Sybil attack with noise data every $t_{m_1} = 10$ normal contributions. The recklessly malicious contributor m_2 is set with much lower capacity and short Sybil cycle $t_{m_2} = 2$ to emphasize the availability of the attacker by comparing m_1.

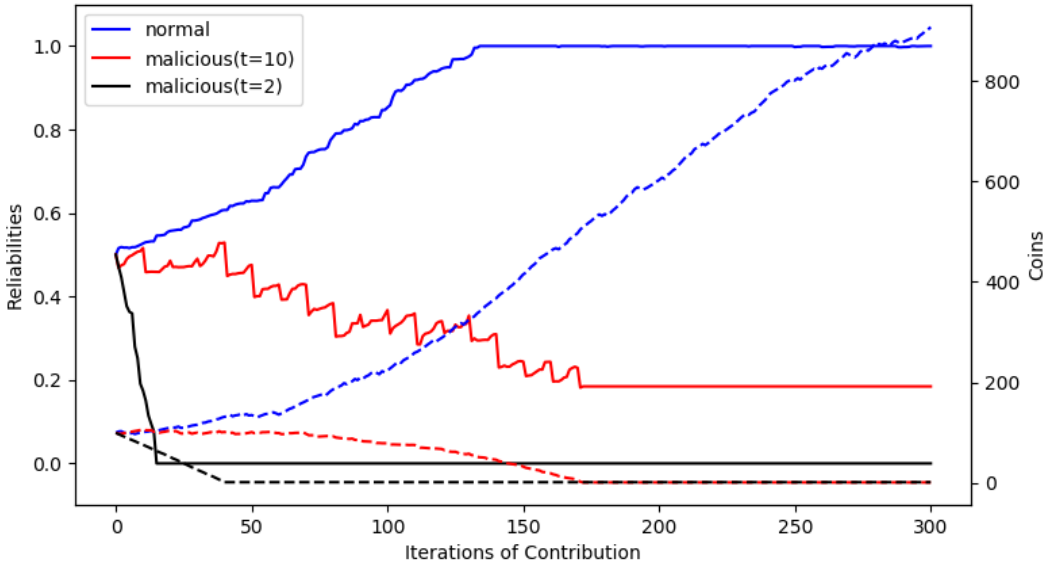

Figure 5. Reliability and property (cryptocurrencies) trends on each contributor and attacker (solid lines show the reliability and dashed lines shows the property.)

As illustrated in Sections 4 and 5, each contribution is validated by the CTI feed F. If the result of the validation function is lower than the threshold, that means a contribution is determined as noise data, the contributor of that data does not get the rewards and loses the deposit. In addition, the reliability of each contributor affects the result of the validation function. Thus, the reliability of contributors is related to the availability of contributors. In the experiment shown in the Figure 5, the threshold of validation function was set to $\tau_F = 0.7$. Each solid line shows the reliability value of each contributor, and each dashed line shows the cryptocurrencies each contributor has.

The contributions from normal contributors are determined as valid data by the CTI feed F. This probability increases the reliability of normal contributors by the iterations. Thus, with each iteration, the reliability of the normal contributors (solid blue line) increases continuously and converges to $\beta_n = 1$. However, the reliability of the malicious contributor (solid red line) and the recklessly malicious contributor (solid black line) converged to the $\beta_{m_{1,2}} = 0$ after iterations. Even though the attacker imitates the behaviors of normal contributors through valid contributions, the attacker cannot retain their reputation since the loss from the penalty of malicious contributions lowers their reputation more significantly compared to the gains through the rewards. Thus, the results show that the proposed framework can effectively and promptly screen the malicious contributions of the Sybil attack.

6.3. Cost of Malicious Contribution

Increasing the cost of the Sybil attack gives the Sybil-resistance to the system. In the proposed framework, each contribution consumes default cost (deposit). To maintain the ability for contribution, each contributor needs to get the rewards from the CTI feeds. However, the reliability of contributors affects the result of the validation function, and this mechanism makes the attacker lose their ability for contribution by preventing them from gaining the rewards for their malicious contributions. In Figure 5, the amount of cryptocurrencies of the normal contributor (dashed blue line) has continuously increased since the normal contributor gets the rewards through their valid behavior. However, the cryptocurrencies of the recklessly malicious contributor (dashed black line) have dramatically decreased since the low reliability of that node means them continuously losing the deposit . Even though the malicious contributor imitates the behavior of a normal contributor, they could not earn a meaningful amount of cryptocurrencies or even lose it. In the experiment shown in the Figure 5, the malicious contributor, which has the Sybil cycle $t_{m_1} = 10$, has lost all their cryptocurrencies after 170 iterations.

In our experiment, we supposed that the attacker imitates the behavior of a normal contributor. To disguise themselves as a normal user, the attackers need to contribute valid data. These valid contributions are used as a cost to perform the Sybil attack, maintaining the reliability of the attacker for continuous attack. Thus, we performed simulations to show the impact of the attacker's cost at the point of the Sybil cycle t. Figures 6–9 show the property of each contributor. In these experiments, we simulated one normal contributor and four malicious contributors by their Sybil cycle. Every contributor has the same capacity for normal contribution $\delta = (m = 0.8, s.d. = 0.1)$, and each malicious contributor reports invalid data $\delta = (m = 0.2, s.d. = 0.2)$ after every t normal contributions.

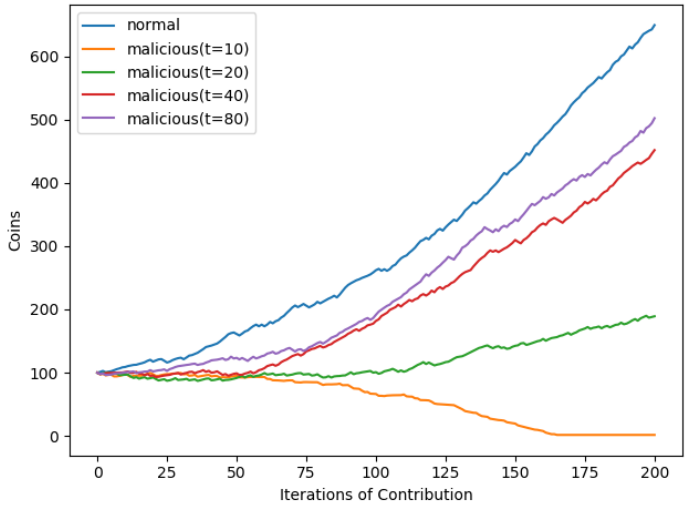

Figure 6. Changes of cryptocurrencies held by each contributor and attacker ($p = 4$).

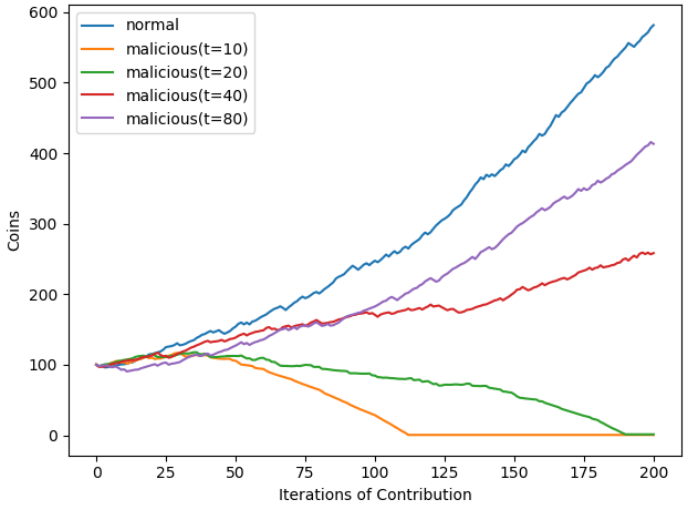

Figure 7. Changes of cryptocurrencies held by each contributor and attacker ($p = 8$).

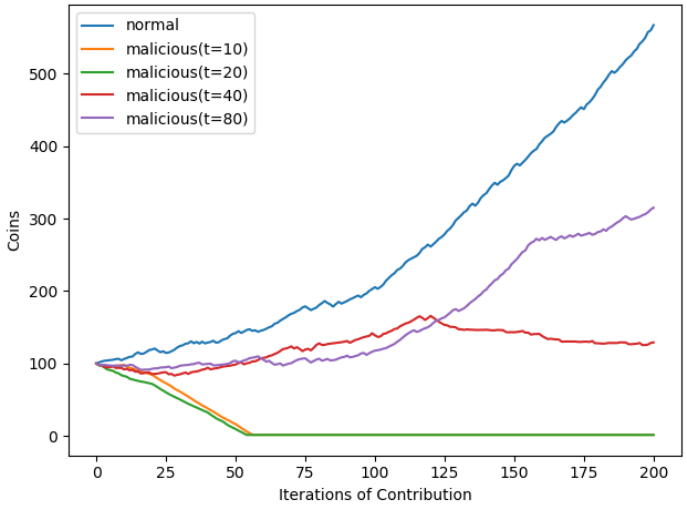

Figure 8. Changes of cryptocurrencies held by each contributor and attacker ($p = 16$).

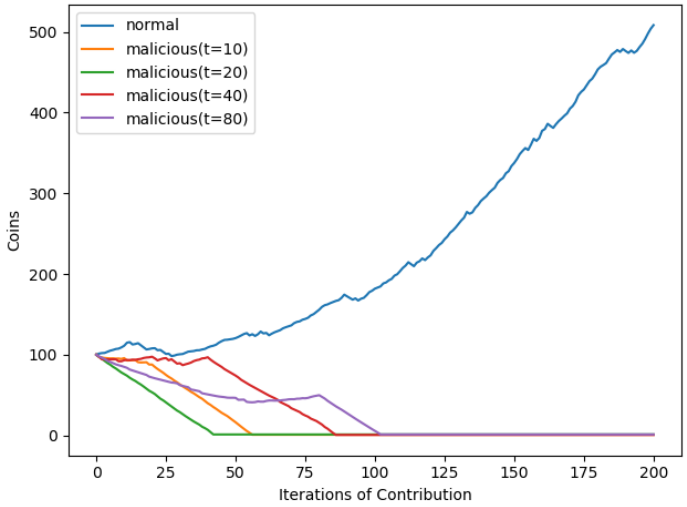

Figure 9. Changes of cryptocurrencies held by each contributor and attacker ($p = 32$).

The result of the CTI feed's validation function gives the penalty p to the updates of the contributor's reliability. The high weight of the penalty term accelerates the decrease in reliability caused by invalid contributions. In the figures, we show the impacts of the attacker's cost (Sybil cycle t) and the penalty term p. In Figure 6, when the penalty weights are much lower than the attacker's Sybil cycles (i.e., $p = 4$), the system cannot screen the tricky contributors with high Sybil cycles such as $t = 40$ or $t = 80$. Their reliabilities became similar to the reliability of the normal contributors. However, when the penalty weights are adjusted corresponding with the attacker's cost (i.e., $p = 8, 16$), in Figures 7 and 8, malicious contributors and their Sybil cycles are revealed. When the penalty weight is enough ($p = 32$), in Figure 9, the availability of an attacker converges quickly to 0. Each CTI feed could adjust the penalty term concerning the circumstances of the whole network. These results show that by evaluating the data reported from users through the blockchain network and sharing the evaluation results, it is possible to effectively block attackers who attempt to perform Sybil attacks. In the proposed CTI framework, attackers quickly lose cryptocurrency assets in the process of conducting Sybil attacks. This loss leads to an attacker losing the ability to contribute the malicious data. Additionally, even if an attacker is disguised as a normal user to perform a long-term Sybil attack, the attacker shows that he must perform a considerable number of normal contributions to achieve the average reliability level of the normal users. This dramatically reduces the effectiveness of long-term Sybil attacks. Thus, the results of this experiment show that it can effectively mitigate two Sybil threats to the CTI system.

7. Conclusions

CTI technology gives an effective and proactive method to mitigate intelligent and advanced cyber threats. Through data analysis and profiling of cyber threats, the CTI system enhances the comprehension of them and provides actionable countermeasures. For this purpose, the CTI system requires various forms and types of data and a massive dataset. Many CTI feeds, which provide CTI services, use open-source intelligence as a data-collection channel to cover the dataset. However, the major problem of this approach is the reliability of the data. Because this data-collection approach permits unconstrained reporting, an attacker can inject maliciously generated or modified data into the system through a Sybil attack to compromise the reputation of specific nodes. Security policies generated by malicious data can misjudge the reputation of network nodes, and this can seriously deteriorate the availability of the entire network.

The BLOCIS framework introduced in this paper is a way to give Sybil-resistance to the CTI system through blockchain-based smart contracts. In our framework, we defined a three-layered architecture for the blockchain-based CTI system. The proposed framework collects the CTI data from various sources and evaluates the validity of data and contributors. This approach can effectively distinguish malicious contributors in numerous data-collection methods . Evaluating all the data and assessing the contributor's reliability can isolate the nodes that continuously report invalid or malicious data. In this paper, we suggest a detailed way to operate and implement the proposed framework in the form of smart contracts and explain the evaluation model for the reliability of contributors. Furthermore, to prove the effectiveness and performance of the proposed framework, we performed simulations in terms of the attacker's reliability and their cost to operate the Sybil attack. In the simulation results, we show that our proposed framework can effectively distinguish the malicious contributor without harmful effects on other normal contributors.

In this research, we evaluated the validity of the data with probability. Evaluating and analyzing the meaning and impact of CTI data is a very domain-specific problem. In future research, we will discuss a way to analyze CTI data considering the risk of Sybil attacks in order to expand the domain of the proposed framework.

Author Contributions: Conceptualization, S.G. and C.L.; methodology, S.G. and C.L.; software, S.G.; formal analysis, S.G.; investigation, S.G.; resources, S.G.; writing–original draft preparation, S.G.; visualization, S.G.; supervision, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00158, Development of Cyber Threat Intelligence (CTI) analysis and information sharing technology for national cyber incident response).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SCADA	Supervisory control and data acquisition
SIEM	Security information and event management
IoC	Indicator of compromise
OSINT	Open source intelligence
STIX	Structured Threat Information eXpression
TAXII	Trusted Automated eXchange of Indicator Information
RSAES-OAEP	RSA Encryption System with Optimal Asymmetric Encryption Padding

References

1. Li, S.; Da Xu, L.; Zhao, S. 5G Internet of Things: A survey. *J. Ind. Inf. Integr.* **2018**, *10*, 1–9. [CrossRef]
2. Javaid, N.; Sher, A.; Nasir, H.; Guizani, N. Intelligence in IoT-based 5G networks: Opportunities and challenges. *IEEE Commun. Mag.* **2018**, *56*, 94–100. [CrossRef]
3. Ahmad, I.; Kumar, T.; Liyanage, M.; Okwuibe, J.; Ylianttila, M.; Gurtov, A. Overview of 5G security challenges and solutions. *IEEE Commun. Stand. Mag.* **2018**, *2*, 36–43. [CrossRef]
4. Khan, M.A.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411. [CrossRef]
5. Lemay, A.; Calvet, J.; Menet, F.; Fernandez, J.M. Survey of publicly available reports on advanced persistent threat actors. *Comput. Secur.* **2018**, *72*, 26–59. [CrossRef]
6. Ghafir, I.; Hammoudeh, M.; Prenosil, V.; Han, L.; Hegarty, R.; Rabie, K.; Aparicio-Navarro, F.J. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Gener. Comput. Syst.* **2018**, *89*, 349–359. [CrossRef]
7. Lobato, A.G.P.; Lopez, M.A.; Sanz, I.J.; Cardenas, A.A.; Duarte, O.C.M.; Pujolle, G. An adaptive real-time architecture for zero-day threat detection. In Proceedings of the 2018 IEEE international conference on communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
8. Duessel, P.; Gehl, C.; Flegel, U.; Dietrich, S.; Meier, M. Detecting zero-day attacks using context-aware anomaly detection at the application-layer. *Int. J. Inf. Secur.* **2017**, *16*, 475–490. [CrossRef]
9. Burger, E.W.; Goodman, M.D.; Kampanakis, P.; Zhu, K.A. Taxonomy model for cyber threat intelligence information exchange technologies. In Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security, Scottsdale, AZ, USA, 3 November 2014; pp. 51–60.
10. Barnum, S. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corp.* **2012**, *11*, 1–22.
11. Tounsi, W.; Rais, H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Comput. Secur.* **2018**, *72*, 212–233. [CrossRef]
12. AlienVault ThreatCrowd. Available online: https://threatcrowd.org/ (accessed on 10 February 2020).
13. ThreatMiner: Data Mining for Threat Intelligence. Available online: https://www.threatminer.org/ (accessed on 10 February 2020).
14. ThreatCrowd Maltego Transforms. Available online: https://github.com/AlienVault-OTX/ThreatCrowd-Maltego (accessed on 10 February 2020).
15. Wagner, T.D.; Palomar, E.; Mahbub, K.; Abdallah, A.E. A Novel Trust Taxonomy for Shared Cyber Threat Intelligence. *Secur. Commun. Netw.* **2018**, *2018*. . [CrossRef]
16. Mihai, I.C.; Pruna, S.; Barbu, I.D. Cyber Kill Chain Analysis. *Int. J. Inf. Sec. Cyber.* **2014**, *3*, 37. [CrossRef]
17. Kiwia, D.; Dehghantanha, A.; Choo, K.K.R.; Slaughter, J. A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence. *J. Comput. Sci.* **2018**, *27*, 394–409. [CrossRef]

18. Cho, S.; Han, I.; Jeong, H.; Kim, J.; Koo, S.; Oh, H.; Park, M. Cyber Kill Chain based Threat Taxonomy and its Application on Cyber Common Operational Picture. In Proceedings of the 2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA), Glasgow, Scotland, UK, 11–12 June 2018; pp. 1–8.
19. Bahrami, P.N.; Dehghantanha, A.; Dargahi, T.; Parizi, R.M.; Choo, K.K.R.; Javadi, H.H. Cyber Kill Chain-Based Taxonomy of Advanced Persistent Threat Actors: Analogy of Tactics, Techniques, and Procedures. *J. Inf. Process. Syst.* **2019**, *15*, 865–889.
20. Qamar, S.; Anwar, Z.; Rahman, M.A.; Al-Shaer, E.; Chu, B.T. Data-driven analytics for cyber-threat intelligence and information sharing. *Comput. Secur.* **2017**, *67*, 35–58. [CrossRef]
21. Ussath, M.; Jaeger, D.; Cheng, F.; Meinel, C. Pushing the limits of cyber threat intelligence: Extending STIX to support complex patterns. In *Information Technology: New Generations*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 213–225.
22. Asgarli, E.; Burger, E. Semantic ontologies for cyber threat sharing standards. In Proceedings of the 2016 IEEE Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 10–11 May 2016; pp. 1–6.
23. Nunes, E.; Diab, A.; Gunn, A.; Marin, E.; Mishra, V.; Paliath, V.; Robertson, J.; Shakarian, J.; Thart, A.; Shakarian, P. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA, 28–30 September 2016, pp. 7–12.
24. Samtani, S.; Chinn, K.; Larson, C.; Chen, H. AZSecure Hacker Assets Portal: Cyber threat intelligence and malware analysis. In Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA, 28–30 September 2016; pp. 19–24.
25. Fachkha, C.; Debbabi, M. Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1197–1227. [CrossRef]
26. Bou-Harb, E. A probabilistic model to preprocess darknet data for cyber threat intelligence generation. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.
27. Moore, D.; Rid, T. Cryptopolitik and the Darknet. *Survival* **2016**, *58*, 7–38. [CrossRef]
28. Lee, S.; Shon, T. Open source intelligence base cyber threat inspection framework for critical infrastructures. In Proceedings of the 2016 Future Technologies Conference (FTC), San Francisco, CA, USA, 6–7 December 2016; pp. 1030–1033.
29. Liao, X.; Yuan, K.; Wang, X.; Li, Z.; Xing, L.; Beyah, R. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; ACM: New York, NY, USA, 2016; pp. 755–766.
30. Symonenko, S.; Liddy, E.D.; Yilmazel, O.; Del Zoppo, R.; Brown, E.; Downey, M. Semantic analysis for monitoring insider threats. In *International Conference on Intelligence and Security Informatics*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 492–500.
31. Kim, N.; Kim, M.; Lee, S.; Cho, H.; Kim, B.I.; Park, J.H.; Jun, M. Study of Natural Language Processing for Collecting Cyber Threat Intelligence Using SyntaxNet. In *International Symposium of Information and Internet Technology*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 10–18.
32. Roturier, J.; Schlatter, B.; Schlatter, D.S. *Bootstrapping a Natural Language Interface to a Cyber Security Event Collection System using a Hybrid Translation Approach*; Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks; European Association for Machine Translation: Lisbon, Portugal, 2019; pp. 134–141.
33. Niakanlahiji, A.; Wei, J.; Chu, B.T. A Natural Language Processing Based Trend Analysis of Advanced Persistent Threat Techniques. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 2995–3000.
34. Gong, S.; Cho, J.; Lee, C. A Reliability Comparison Method for OSINT Validity Analysis. *IEEE Trans. Ind. Informat.* **2018**, *14*, 5428–5435. [CrossRef]

35. Mahlangu, T.; January, S.; Mashiane, T.; Dlamini, M.; Ngobeni, S.; Ruxwana, N. Data Poisoning: Achilles Heel of Cyber Threat Intelligence Systems. In Proceedings of the ICCWS 2019 14th International Conference on Cyber Warfare and Security: ICCWS 2019, Academic Conferences and Publishing Limited, Stellenbosch, South Africa, 28 February–1 March 2019; p. 221.
36. Jasper, S.E. US cyber threat intelligence sharing frameworks. *Int. J. Intell. CounterIntell.* **2017**, *30*, 53–65. [CrossRef]
37. Kim, E.; Kim, K.; Shin, D.; Jin, B.; Kim, H. CyTIME: Cyber Threat Intelligence ManagEment framework for automatically generating security rules. In Proceedings of the Proceedings of the 13th International Conference on Future Internet Technologies, Seoul, Korea, 20–22 June 2018; ACM: New York, NY, USA, 2018; p. 7.
38. Masombuka, M.; Grobler, M.; Watson, B. Towards an Artificial Intelligence Framework to Actively Defend Cyberspace. In Proceedings of the European Conference on Cyber Warfare and Security, Academic Conferences International Limited, Oslo, Norway, 28–29 June 2018, p. 589–XIII.
39. Abe, S.; Uchida, Y.; Hori, M.; Hiraoka, Y.; Horata, S. Cyber Threat Information Sharing System for Industrial Control System (ICS). In Proceedings of the 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Nara, Japan, 11–14 September 2018; pp. 374–379.
40. Riesco, R.; Larriva-Novo, X.; Villagra, V. Cybersecurity threat intelligence knowledge exchange based on blockchain. *Telecommun. Syst.* **2019**, 1–30. [CrossRef]
41. Homan, D.; Shiel, I.; Thorpe, C. A New Network Model for Cyber Threat Intelligence Sharing using Blockchain Technology. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–6.
42. Kshetri, N. Blockchain's roles in strengthening cybersecurity and protecting privacy. *Telecommun. Policy* **2017**, *41*, 1027–1038. [CrossRef]
43. Wood, G.; Ethereum, F. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
44. Dannen, C. *Introducing Ethereum and Solidity*; Springer: Berlin/Heidelberg, Germany, 2017.
45. Housley, R. RFC3560: Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS). Available online: https://tools.ietf.org/html/rfc3560 (accessed on 11 February 2020).