Cybersecurity threat intelligence knowledge exchange based on blockchain

Proposal of a new incentive model based on blockchain and Smart contracts to foster the cyber threat and risk intelligence exchange of information

R. Riesco · X. Larriva-Novo · V. A. Villagra

Abstract
Although cyber threat intelligence (CTI) exchange is a theoretically useful technique for improving security of a society, the potential participants are often reluctant to share their CTI and prefer to consume only, at least in voluntary based approaches. Such behavior destroys the idea of information exchange. On the other hand, governments are forcing specific entities and operators to report them specific incidents depending on their impact, otherwise there could be sanctions to those operators which are not reporting them on time. Obligations and sanctions are usually discouraging participants to share information voluntarily which will just share and report what is strictly required. We propose a paradigm shift of cybersecurity information exchange by introducing a new way to encourage all participants involved, at all levels, to share relevant information dynamically. It will also contribute to the support and deployment of Dynamic Risk Management frameworks to keep risks under an acceptance level along the time. Participants will have new and specific incentives to share, invest and consume threat intelligence and risk intelligence information depending on their different roles (producers, consumers, investors, donors and owner). Our proposal leverages from standards like Structured Threat Information Exchange, as well as W3C semantic web standards to enable a workspace of knowledge related to behavioral threat intelligence patterning to characterize tactics, techniques and procedures. At the same time, we propose an Ethereum Blockchain Smart contract Marketplace to better incentivize the sharing of that knowledge between all parties involved as well as creating a standard CTI token as a digital asset with a promising value in the market. Simulations and an experimentation were performed to demonstrate its benefits and incentives, but also its potential limits with regard to storage and cost of transactions.

Keywords STIX™ · SWRL · OWL · Dynamic Risk Management (DRM) · Cyber threat intelligence (CTI) · Ethereum Blockchain Smart contract

1 Introduction

1.1 Motivation

In cyber security, we have traditionally been focusing on identifying what we want to protect, and then building defenses around them along the time. When adversaries breach defenses, organizations adapt themselves to prevent those breaches to occur, and, of course, as we adapt, our adversaries adapt as well. While it is critical to ensure good enough defenses, it is known that it is not enough. Cyber Threat Intelligence (CTI) is used to better understand, predict and adapt to the behaviors of malicious actors whether

they are criminal groups, activist or even nation states. Cyber threat intelligence can take several forms including detailed information about the malware, indicators of compromise (IoC) or specific techniques that malicious actors use to steal information. By having this knowledge, we can update our defenses against the threats. There are many possible sources of cyber threat intelligence, such as historic incidents, open source intelligence (OSINT) [1], any threat feed, ISACs (information sharing and analysis centers) or even government threat's sharing programs.

The collective ability to detect and defend against malicious activity, by sharing information about adversaries and their behaviors, is paramount. It is a joint initiative where public and private organizations (including security vendors) should work together to find better ways to create, share and use cyber threat intelligence.

When a certain threat shares the same motivation among different organizations, all of them are in danger. Once a piece of knowledge about such threat is available (the threat is characterized somehow), all potential affected organizations could benefit from having access to that knowledge. Until today, Indicators of Compromise (IoC) are used as the de facto type of information to be shared about threats, especially if we want automatic and actionable intelligence.

Information Sharing and Analysis Centers (ISAC) are non-profit organizations that provide a central resource for gathering information on cyber threats (in many cases related to critical infrastructures). They allow two-way sharing of information between the private and the public sector about root causes, incidents and threats, as well as experience, knowledge and analysis.

European legislations like the NIS Directive[1] and the Cybersecurity Act nourish the creation of sectoral ISACs and PPPs within the EU. The NIS Directive, among others, separates the operators of essential services in sectors and tasks the operators to implement requirements on incident reporting which is far from a voluntary and enriched approach. It is mainly an obligation to report incidents under certain conditions to their national CSIRTs, that could have associated sanctions by competent authorities, if reporting and incident handling was not properly done by the entity. Obligations and sanctions, came (usually) after a period of unsuccessful voluntary sharing. This approach is very different from OSINT initiatives but also different from any expected win-win approach, plenty of incentives, like the ones presented in our work.

The creation of sectorial ISACs at national level could further assist with the implementation of these provisions. During the transposition of such European legislation to each national law, these communities could be further informed and advised by policy makers.

Information sharing between national stakeholders but even in cross country cases is one important aspect for cyber security. Knowledge on tackling cyber attacked, incident response, mitigation measures and preparatory controls are recommended to be shared between the relevant stakeholders.

On the other hand, unsuccessful voluntary sharing has several and different root causes. Several studies have been analyzing why people is often reluctant to share [2–7].

We present, in Table 1, an inventory of the open challenges and limits of existing solutions in information sharing nowadays. The table includes references from the bibliography to support each concept. All the open challenges can be grouped into the following groups:

- The lack of trust (infrastructure, admin and peers)
- The lack of incentives (business cases) provided to all roles simultaneously.
- The asymmetry between consumers and producers.
- The reliability and accuracy of CTI data.
- The lack of semantics (unambiguous data) to exchange knowledge (beyond single pieces of data).
- The effectiveness and efficiency of platforms (automation).

We propose a solution, combining the use of semantic web ontologies, Structured Threat Information Exchange (STIX™) and Ethereum Blockchain, in order to cover all open challenges at the same time. As an example, we will use built-in features provided by the blockchain, like the accounting, identity management, availability and the possibility to create, and to run decentralized applications (smart contracts). In our case, the whole system will be running as a smart contract to handle all data and interactions.

We provide also incentives based on other relevant feature: the tokens. It is a great built-in feature of Ethereum, created to provide a standard way for developers to create any digital representation of any asset. By definition, tokens are interchangeable between them. A token can represent anything from a physical object, like gold, to a native currency. It can also represent any financial instrument like stocks and bonds.

We created the first digital asset (the CTI Token) to represent Cyber Threat Intelligence data, as a digital asset. The token provides great benefits comparing to its physical representation, it can be operated automatically between smart contracts and, if it is a standard token (e.g. ERC20 standard), it is interoperable by definition with any other standard token. As an example, you would be able to exchange stock options and CTI tokens, or even gold by their digital representation within Ethereum. This incentive will attract investors, which are often reluctant to invest in cybersecurity products due to the difficulties to estimate their return on investments (ROI) and potential exits. We will use Montecarlo simulations to

Table 1 Threat intelligence sharing: open challenges and limitations of existing solutions

ID	Concept	References
1.	Users reluctance to participate in cyberincident information sharing	[6–8]
2.	Users reluctance to share sensitive CTI data (e.g. due to privacy concerns)	[2,6,8,9]
3.	Trust issues between users and platform providers are mostly neglected	[6,7,10]
4.	The lack of trust in the sharing infrastructure	[6,11]
5.	Information sharing asymmetry: more consumers than producers	[3,7]
6.	Lack of incentives for information sharing	[3,4]
7.	Lack of business models associated within cybersecurity initiatives	[3,4]
8.	Misapprehended costs	[5,7]
9.	CTI data mostly limited to IoC instead of advanced intelligence	[7,9,10]
10.	Low reliability, accuracy and quality of threat intelligence information data	[1,12,13]
11.	IoC as ephemeral data	[9,14]
12.	Timing: Fast sharing is important but not enough	[7,8]
13.	Unmanageable volume of (big) threat data	[8]
14.	CTI data taxonomies still lack of enough expressivity (e.g. TTP definition)	[9,15]
15.	CTI data taxonomies lack of semantics (e.g. unambiguous and universal understanding by reasoners)	[9]
16.	Static approach (e.g.signatures) does not match the dynamic nature of new generation of threats	[8,9]
17.	Grained situational awareness need to be linked to information sharing	[9], [11]
18.	CTI not synchronized within information security management systems dynamically at all levels	[9,11]
19.	There is no a common definition of threat intelligence sharing platform	[10]
20.	The majority of platforms are closed source	[10]
21.	Most platforms focus on data collection instead of analysis	[10]

define key parameters for the proposed system dynamics. We also evaluated the benefits and limitations of our model by an experimentation.

1.2 Approach and results

The objective of the work is to provide a coherent solution for all of the open challenges in Cyber Threat Intelligence sharing, seen in Table 1, at the same time. The detailed characteristics of our all-in-one solution, are presented in the Table 2.

We propose an innovative Cyber Threat Intelligence (CTI) Exchange model, based on the combination of semantic web, STIX™ and the Ethereum Blockchain. It provides new type of incentives to all roles involved. Current approaches do not provide enough incentives for producers, as a result, most of the entities involved are just consumers.

Our proposal also provides incentives to engage new type of roles like the investors. We propose different type of economic incentives:

- A new CTI token (under the ERC20 [16] standard), which represents the threat and risk intelligence data, as a digital asset. The creation of a new token will facilitate the interoperability with any other standard token within the Ethereum blockchain
- Crypto currencies (cash in Ether currency) to support the use of taxes in the exchange of knowledge (pay per use).

In our work, we also propose to share enhanced knowledge in the format of semantic algorithms or rules (beyond IoC). It uses semantic variables in an OWL[2] version of STIX™v2 format, to support complex representations with more expressivity. Rules are provided in the format of semantic rule language rules (SWRL[3]), as proposed by Riesco et al. [9]. The use of ontologies, by definition, enables the interoperability and unambiguity of concepts. At the same time, it allows the usage of semantic reasoners to infer new knowledge, which might help to bring the needed automation keeping the data consistency (reliability). It will be also more effective with regard to the number of data to be shared, because the value of the algorithm is greater than just a specific IoC. It is also less ephemeral [14].

We run different experiments and simulations to demonstrate the benefits and limitations of our model. As a result, the proposed model brings several benefits to the state of play, among them:

- **Accountability** and **trusted sources** who (private key) really shared/consumed/invested what and when.

[2] https://www.w3.org/OWL/.

[3] https://www.w3.org/Submission/SWRL/.

Table 2 Solutions provided by our proposal to all open challenges and limitations identified in Table 1

ID	Solution provided	IDs of Table 1 solved
1.	CTI and risk data expressiveness provided by W3C semantic web ontologies and SWRL	10, 14, 15
2.	Sharing of CTI and Risk behavioral context aware rules or algorithms beyond IoC	1, 2, 9, 11, 12, 13, 16
3.	Interoperability due to the use of semantic web ontologies	10, 14, 15
4.	Interoperability and easy adoption due to the use of an OWL version of STIX™	14, 15, 18
5.	Automation by using semantic web reasoners (analysis beyond data collection)	10, 12, 13, 17, 21
6.	Quality of the data as inconsistencies are automatically detected by reasoners	10, 15, 21
7.	Inference (new knowledge) provided by the use of semantic web reasoners	1, 2, 6, 19
8.	Reduced volume of data needed (e.g. algorithms vs. IoC)	8, 9, 11, 12, 13
9.	Situational awareness linked to information exchange provided by ontologies	15, 16, 17
10.	Information sharing at all levels (operational, tactical and strategic)	15, 16, 17, 18
11.	Evolutionary economic incentives ("Ether" and a new "CTI token") as business model	1, 5, 6, 7, 8
12.	Blockchain inherent trusted network (validations and verifications of all transactions)	1, 3, 4, 12
13.	Smart contract decentralized application (dApp) instanced in the blockchain	3, 4, 8, 19
14.	Transparent running (open source) code (what, when, why, who)	3, 4, 10, 20, 21
15.	Time-stamping of each transaction within blockchain blocks	3, 4, 12
16.	Very low cost (just gas) (no infrastructure needed)	6, 7, 8
17.	Interoperability between smart contracts (calls between dApps)	3, 4, 19, 20, 21
18.	Interoperability between tokens	1, 6, 7, 19
19.	dApp efficiency (EVM runs optimized code only) and decentralized availability	3, 8, 13, 19

- **Availability** and **low cost** creating a decentralized marketplace application in Ethereum Blockchain with all the benefits of blockchain decentralization supported by several nodes, without the need of specific investments in an owned infrastructure.
- **Secure** by using specific design-patterns and secure coding together with specific security enhancements provided by the Ethereum Blockchain. It is by definition a trusted network. It provides trust even between untrusted parties (e.g. crypto currency transaction).
- **Invest ready** and **attractive** by a **win-win** approach. It provides **economic incentives to several roles simultaneously**: creating a token named "CTI token" (crypto-token), which is interesting for investors and users. It will provides incentives. We made Montecarlo simulations to better select key parameters of our system.
- **Semantic expressiveness, effectiveness and efficiency by using a combination of standards**: the use of a combination of standards enables the reasoners to better contribute to the automation and decision making processes. The exchange of algorithms beyond IoC will provide better detection and prevention capabilities to all stakeholders involved. At the same time, it will contribute for the exchange of real knowledge, its expressivity, its consistency and its reliability. We propose the exchange of the analysis of the data (knowledge or advanced intelligence) instead of the exchange of the data itself.

1.3 Contributions

(i) A new incentive model for Cyber Threat Intelligence (CTI) sharing, based on Ethereum Blockchain. New evolutionary economic incentives are provided by a combination of Ether and the creation of a new digital asset for threat intelligence, the CTI token (ERC20 compliant). (ii) A new enhanced version of peers. A semantic approach of Cyber Threat Intelligence sharing systems within Dynamic Risk Management (DRM) processes. For that, we support the exchange of CTI semantic web algorithms (beyond the exchange of IoC data) in the format of SWRL and an OWL enhanced version of STIX™v2, at all levels (operational, tactic and strategic level). (iii) Simulations for model optimization, and experimentation, to demonstrate its benefits and its limits, especially in terms of costs.

1.4 Paper organization

The rest of this paper is organized as follows. The next Sect. 2 reviews related work. In Sect. 3, the problem and our proposal is described. Then, the selection of a specific implementation design is justified among the different alternatives in Sect. 4. We then provide in Sect. 5, major implementation details to facilitate reproducibility and a better understanding on how the incentives are provided along the exchange of CTI data. After evaluating the benefits and limitations in Sect. 6, the paper concludes in Sect. 7.

2 Related work

Information exchange was mainly invented to help others, by sharing our knowledge, at the same time we can improve our level of protection. It is clear, that no one is ready to fight alone against cybersecurity threats.

Wagner et al. [12] presented MISP, the Malware Information Sharing Platform, which has been one of the most used open sourced platforms since 2016. Authors observed, from 20 to 40 new instances deployed each day, since its first release. MISP allows the exchange of Indicators of Compromise (IOC: hashes, hostnames, IP addresses, URL, etc.). The size of the data shared since the beginning (about 100 K md5 hashes a month), has demonstrated the real demand of the end users. On the other hand, authors propose new research directions with regard to the quality of the data, as it is usually not reliable. Authors also suggest that new ways to handle such volume of (big) data in the future, might be needed. They suggest also, to work in the development of a formal model or taxonomy to overcome the issues of their triple tag approach, as it still lacks enough expressivity.

Sauerwein et al. [10], analyzed 22 threat intelligence sharing platforms. Their key findings are really interesting, among them:

- There is no a common definition of threat intelligence sharing platform.
- STIX™is the facto standard for describing threat intelligence.
- Platforms primarily focus on sharing of IoC.
- The majority of platforms are closed source.
- Most platforms focus on data collection instead of analysis.
- Trust issues between users and platform providers are mostly neglected.
- Academic and commercial interest in threat intelligence sharing increases.
- Many manual tasks make the user the bottleneck.

As a result, the authors indicate that the threat intelligence sharing lacks a consistent definition. They also indicate that threat intelligence sharing is comparable to data warehousing, and it does not provide real intelligence. We agree with authors in all these conclusions.

De Fuentes et al. [6], indicate that privacy is paramount to foster cooperation, particularly when insecure infrastructures are used to support sharing. They propose PRACIS, a protocol that provides privacy-preserving and aggregatable cybersecurity information sharing. PRACIS provides these properties by leveraging existing format-preserving and homomorphic encryption techniques and adapting them to the particularities of standard message formats such as STIX™. Authors anyway provide a new layer of security, when the network is untrusted.

We propose a network based on blockchain precisely to overcome the lack of trust in the network, in the administrator of the platform and in the peers. Moreover, the system is running as a smart contract (or a dApp), which is instanced in the blockchain. It will be able to receive external calls either by peers or from other smart contracts (e.g. other CTI Marketplaces). There will be no human intervention (administrator), after it is deployed in the blockchain. As a good practice, the source code is usually appearing along the smart contract instance, which give more trust to the users (the ABI[4]).

Tounsi et al. [8], suggest that the static approach of traditional security, based on heuristic and signatures, does not match the dynamic nature of new generation of threats, that are known to be evasive, resilient and complex. Organizations need to gather and share real-time cyber threat information, as well as to transform it into threat intelligence. Authors explain why there is reluctance among organizations, to share threat intelligence data. They provide sharing strategies based on anonymity, in order to reduce the risks in case of a data leak. They also show in their work, why having a standardized representation of threat information, can improve its quality. We support such statement as well.

Leszczyna et al. [11], indicate that there is a need to link a fine grained situational awareness to information sharing data. They propose different topologies, ranging from a decentralized peer to peer, to a confederated topology as seen in Figs. 1 and 2. Authors propose a framework, to enhance current threat intelligence services by providing data at all levels.They propose to integrate all the data at the same time, specially in the context of critical infrastructure protection. They still work in the IoC domain which under our opinion, can be improved by our work.

In addition to this, we consider that their decentralized topology approach is really not a decentralized platform. It is a peer-to-peer network of different instances. We propose a pure decentralized infrastructure by using the blockchain. Our decentralized application will be running into all network nodes at the same time as seen in Fig. 3.

Visik et al. [3], characterize the main reasons about why people is reluctant to share information, if there are not enough incentives for them. Authors suggest that users do not have the same knowledge about the value of artifacts offered, or simply they do not want to share their own data or incidents, as they could be giving a competitive advantage to their own competitors. Authors provide a clear understanding that introducing new economic incentives and business models in cybersecurity initiatives, as part of an economic modeling, would derive in greater societal benefits. As suggested by authors, we propose new economic incentives in our work.

[4] https://solidity.readthedocs.io/en/develop/abi-spec.html.

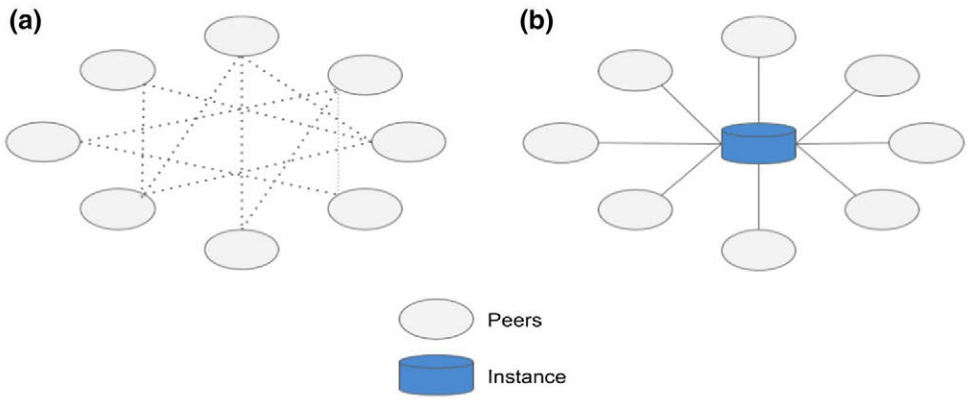


Fig. 1 Information sharing topologies: **a** decentralized peer-to-peer sharing **b** centralized sharing through an instance

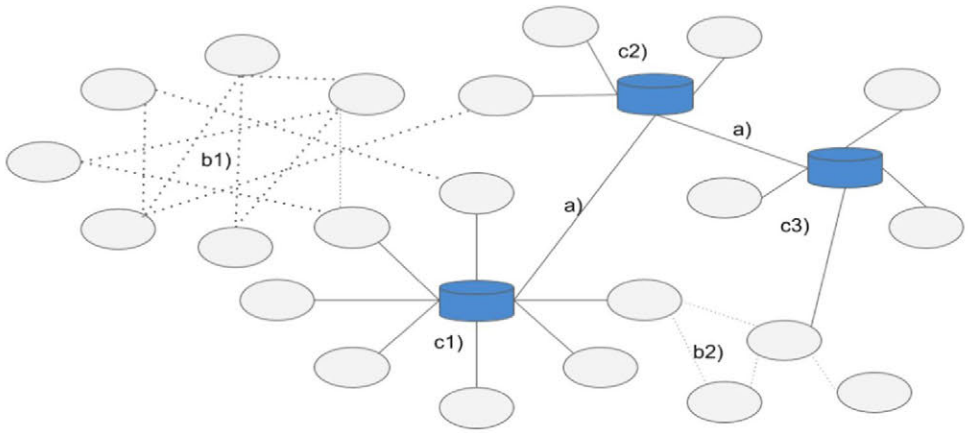


Fig. 2 Confederated information sharing topology (a) mixed with either decentralized P-to-P (b1), (b2) or centralized topologies (c1), (c2) and (c3)

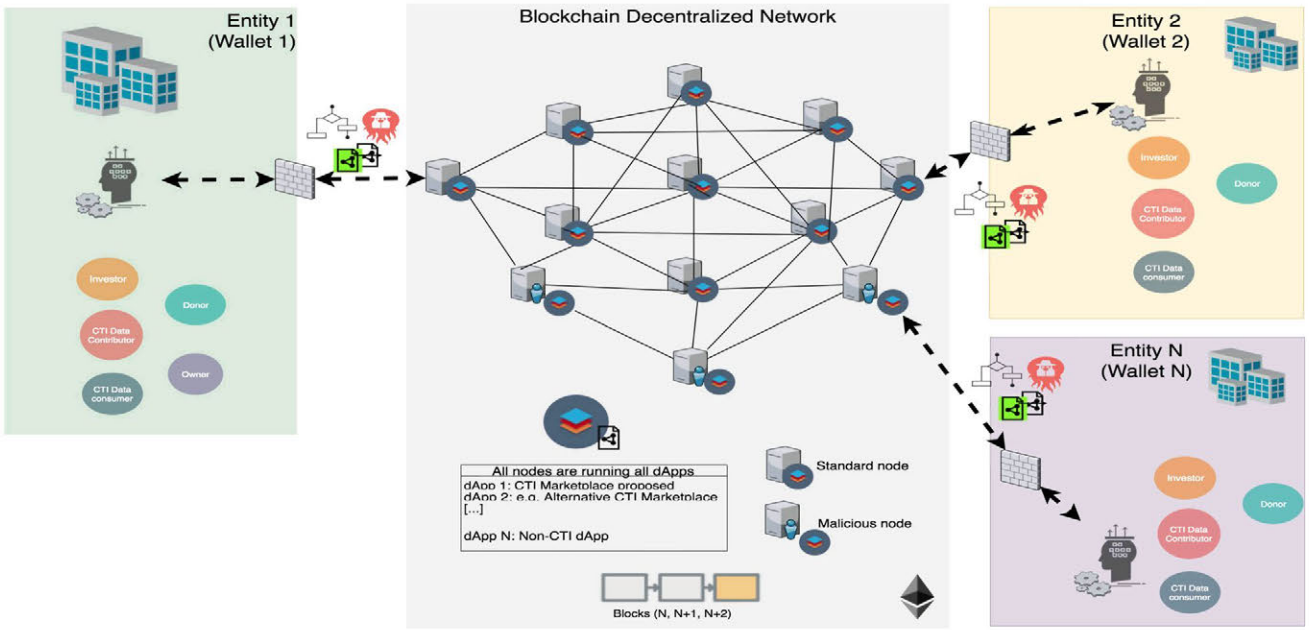


Fig. 3 Proposal of an all-in-one pure decentralized topology based on blockchain

Tosh et al. [4], propose a cybersecurity information exchange framework (CYBEX), based on a gaming exercise. Authors used an evolutionary analysis of the "participation cost", a driving parameter for attracting firms to join and transact cyber threat intelligence (CTI) with other firms. There are some interesting conclusions in the study like, for example, a simple incentive can be the reduction of the participation cost (negative cost vs positive cost). On the other

hand, when the cost is positive, the strategy of each organization depends on how many members are still contributing. If the participation is less than the 65% , then the organizations stop to share. As a conclusion, any external motivation associated to economic incentives will foster the participation of peers. They used an evolutionary cost model using a gaming theory.

Skopik et al. in [5], suggest that due to networks have grown to a scale and complexity, and have reached a degree of interconnectedness, their protection can often only be guaranteed and financed as shared efforts. Consequently, authors suggest that new paradigms are required, for detecting contemporary attacks and mitigating their effects. Many attack detection tasks are still performed only by each individual organization. Authors indicate that information sharing is a crucial step, to acquiring a thorough understanding of large-scale cyber-attack situations, and is therefore seen as one of the key concepts to protect future networks. Authors also propose improvements in standardization and legal aspects, as open challenges. Our proposal will address most of these topics.

Riesco et al. [9], propose to work in risk domain dynamically to keep risks under an acceptance level along the time. In order to do that, authors propose to leverage cyber threat intelligence for a Dynamic Risk Management (DRM) framework. One of the building blocks, is having an enhanced or advance intelligence knowledge, which can be understood by semantic reasoners [17] and humans, at the same time. The intelligence goes beyond indicators of compromise (IoC) in the format of intelligence algorithms using a combination of standards like STIX™v2 [18] formatted as OWL [19] ontologies [20] and semantic web rule language (SWRL) [21]. By using this combination of standards, authors are able to represent context behavioral patterns like the representation of techniques, tactics and procedures (TTP) [15]. Authors propose as future research direction, to exchange this type of algorithms beyond simple IoC in order to provide and share the real knowledge (how to detect). By using a mix of standards, authors can overcome the expressivity barriers to define TTP in the format of algorithms. It also enables the usage of semantic reasoners [17] to infer new knowledge. They propose that new incentive models still have to be developed, as a future research direction. We implemented this specific approach in our work, to demonstrate their benefits.

In [22], the authors propose the use of Blockchain for fighting insurance fraud. The authors propose the creation of a marketplace to share intelligence data about fraud of users, however it lacks of empirical deployment, an incentive model, as well as calculations related to cost. On the other hand, it is handling fraud intelligence which is different from cyber threat intelligence.

There is a very interesting approach, in [23], that is implementing a bug bounty program using a decentralized Marketplace. It is used to connect different experts with potential customers which are willing to pay for professional cybersecurity services. They are also creating a token, named Nectar, in order to use it for exchanging value. Customers can raise a kind of auction for any interested expert or send a direct contract to check if a specific file is malicious or not. In our case, we propose to go beyond IoC or bug bounties into threat intelligence and risk intelligence exchange. Their model is still based on IoC (hashed) and is specific oriented to antivirus.

Another interesting example [24], where Blockchain is being proposed as a cost effective storage of intelligence data, does provide storage and sharing features for the life-cycle state of incidents. Authors suggest a Blockchain-based solution for life-cycle management and automatic classification of cyber security documents according to their expected threat level. It is clear that blockchain provides several possibilities like we demonstrate in our work.

3 The problem and our proposal

3.1 The problem: open challenges and limitations of existing solutions

There are multiple benefits of information exchange in cybersecurity, like threat prevention and detection, between others. On the other hand, there are also several unsuccessful voluntary sharing initiatives, which have been studied by multiple authors [2–7]. A summary of all the following problems can be seen in the Table 1.

Nowadays, there are multiple information sharing platforms running, with millions of IoC being shared everyday, and thousands of users [10,12]. One of the most promising models for information exchange was Open Source Intelligence (OSINT). Several entities and individuals share cyber threat intelligence data openly, along several platforms, since a lot of time ago. This promising model, based on open source intelligence, has demonstrated that is not reliable [1,13] but necessary. Organizations can also use deception techniques, where they can intentionally or unintentionally add, delete, modify, or otherwise filter the information made to the general public. It is important to evaluate the reliability of open sources, in order to distinguish their objectives, factual information; bias; or deception. Non authoritative sources lack reliability and trustworthiness and seldom stand apart from authoritative sources.

When evaluating sources of information to determine reliability and credibility, we should consider:

- *Identity* Who produced the information (for example a student, teacher, political organization, or reporter)?

- *Authority* How much does the source know about the information?
- *Motive* Why was the information published?
- *Access* Did the source have direct access to the event or information?
- *Timeliness* What is the date of the information?
- *Internal and external consistency* Does the information contradict any policies?

Cyber threat intelligence information should be reliable by definition, if we want actionable intelligence, that is, the possibility to run automatic security processes. Organizations will only automate those processes using reliable information, if not, their processes will not be reliable at all. More automation is needed due to the trend in the number of systems to protect, but also due to the increasing complexity of our networks. If our systems are automatically loading IoC data to build filter rules, we need to trust the entity or individual behind, that is, the one who shared that IoC. If not, we can be loading incorrect IoC or even worse, we can help a malicious actor to success. It is then clear that we need to trust OSINT sources [1,13], as a necessary condition, before loading those IoC in our systems. More than that, maybe it is not enough to trust the entity itself, as some IoC shared by them could be part of a third party detection, we then need to trust each piece of data. Specific validation processes, as well as sources and evidences' auditing, accounting or checks, are recommended. As an example, time-stamping related data to when that malicious behavior was seen, who shared the data, when it was updated, modified or even removed, is very interesting for a healthy and up to date system.

These platforms are based on sharing indicators of compromise (IoC) [10]. Once an IoC is shared by a certain entity or individual, it can be used by anyone who has access to that open platform. The usage can also be automated, for example, to protect the perimeter. If a firewall receives a new malicious IP address, it can block either the incoming or outgoing network traffic related to that IP. Several tools and plugins are available to facilitate such integrations.

Despite these integrations, all the static approaches, based on signatures, have demonstrated that they do not match the dynamic nature of new generation of threats [8,9,14]. Using IoC to fight against unknown threats, is then ineffective. New emerging techniques, tactics and procedures (TTP) remain undetected to any vendor solution, whose detection capabilities are based on known threats. Contextual or behavioral aware rules, are the only plausible solution to fight unknown threats. Patterning, algorithms and rules, are the only way to detect suspicious behaviors beyond a detection strategy, based on signatures (e.g. hashes).

Vendors are usually focused on data collection instead of analysis [10]. They are providing unmanageable volume of (big) threat data [8], however, having threat intel data is not having real intelligence. Real intelligence is associated to analyzed and processed threat intel data. This new approach is known as advanced intelligence. Advanced intelligence might also be represented, as complex structures like a TTP (Techniques, Tactics and Procedures) [15]. If we were able to work in such dimension, we will not need to share and store dozens or hundreds of IP addresses, domains and or hashes but the TTP algorithm itself. The idea is equivalent to the usage of DGA (domain generation algorithm) versus the use of single domains, however we go beyond that in our proposal. A TTP description can combine a DGA together with many other concepts in the same rule or algorithm as a more effective behavioral pattern. Once such an algorithm is shared, it can be more effective than sharing hundreds of related IoC, if each of them is ephemeral. As a result, new proposals suggest the idea to work in the dimension of TTP as a combination of intelligence data, malicious behaviors and patterns to detect them.

Defining TTP rules is not easy nowadays as there is still room for improvement with regard to the lack of algorithm expressivity [9,15,25]. CTI data taxonomies nowadays, lack of semantic, unambiguous and universal understanding. However, initiatives like STIX™are being considered as a de facto standard and it is widespread adopted by the industry. It is probably one of the most promising standard for it. What is clear is that sharing a TTP, will be more effective than sharing an isolated IoC. Once a taxonomy or standard is agreed among a huge number of stakeholders, an efficient and effective knowledge transfer will be possible between them. Without an agreed taxonomy, it would be nearly impossible to automate detection without the need of customized parsers. We propose to use a semantic enhanced version of STIX™to solve that expressivity issue.

In order to better prevent or mitigate any attack, the timing to receive new knowledge is always critical. Fast sharing is important then however it is not enough [7,8]. The grained situational awareness, as a way to understand and make decisions at strategic and tactical levels, needs to be linked to the information sharing initiatives [9,11]. New knowledge must be applied to reduce our risk exposure along the time. Despite the great number of information sharing initiatives, all of them are at technical level. As a result, CTI is usually not synchronizing threat intel data with the information security management system (ISMS) dynamically as suggested by authors in [9].

Users are reluctance to participate in cyber incident information sharing beyond cyber threat intelligence sharing [7,8]. These are bad news because several organizations have the same vulnerabilities, or have in common, the same threat actor's motivation. If certain incident is shared, several entities or users could be activated to prevent a potential attack.

It is well known, that in practice, there are usually more consumers than producers of CTI Data. In some cases it is

because most of the users, are usually reluctance to share sensitive CTI data [2,8,9]. In other cases it is because the lack of enough incentives [3,7]. There are few original contributors of CTI data in proportion to the consumers of that data. The lack of trust, reliability, the sensibility of the information to share as well as the type of data to be shared (IoC), are some of the factors that are discouraging.

In some cases, the users are worried about their reputation risks in case of a sensitive information leakage. The anonymization of certain data is proposed [2] as a preventive solution, but it also has trade-offs. The problem is how to provide actionable intelligence, to protect another entity from a similar attack, if it is based on an anonymized version of the real information. In other cases, users are also worried to give any kind of indirect competitive advantage to their own competitors, as they cannot control the redistribution of such data. There are some initiatives to address this issue like the Traffic Light Protocol (TLP),[5] between others.

Trust issues between users and platform providers are mostly neglected [7,10]. Trust is the key. No one will share relevant information to anyone, without it. Trust must exist between peers, between users and the sharing infrastructure [6,11], and also between the user and the platform administrator. The majority of platforms are closed source [10], which is reducing trust.

In order to have more CTI data producers joining the system, it must be trustworthy and attractive. In general, cybersecurity initiatives lack of associated business models [3,4]. Economic incentives can be a good solution, if those are designed in a win-win approach (e.g. honest approach). We propose an innovative model to introduce such economic incentives. They are some kind of evolutionary or dynamic incentives like the use of a new CTI token.

There are usually high associated costs, but there are also misapprehended costs [5,7]. There are different ways to build an information sharing topology: centralized, decentralized (as seen in Fig. 1), confederated or even there can be a mixed information sharing approach (as seen in Fig. 2). At the end, all of them are instances managed by an administrator, which is in charge of its operation and management. Cost are proportional to the number of instances but all of them have associated ongoing costs, which is far from our proposal.

The blockchain will provide us with several benefits, however it still has some limitations associated to the novelty of technology. It uses an EVM (Ethereum Virtual Machine) which still far from a Turing complete machine. The compilers limitations and the limitations of its coding language (e.g. solidity), are still not good enough. There are important limitations, among them, the support of non-integer values, as well as the confidentiality and privacy of specific data. As an example, a privacy declared variable (private storage), could be read by using certain hacking techniques nowadays. Other limitations are related to the storage capacity of their blocks, however there are certain possible workarounds that will be discussed later.

Despite the blockchain limitations, we will use some of its great benefits, among them, the accounting, identity management, availability and especially, the tokens, as a built-in feature to represent any digital asset that will enable new incentives for CTI sharing.

3.2 Our proposal: a new model for information sharing

3.2.1 A pure decentralized infrastructure based on Blockchain

As seen in Figs. 1 and 2, there are different types of topologies used nowadays, however, none of them are based on a pure decentralized infrastructure as we understand. All instances or platforms are managed by a centralized or unique entity at the end. We understand that authors in [11], are really talking about a peer-to-peer interaction between different user's instances, instead of a pure decentralized platform. Suggested platforms by authors are not decentralized, at least in a pure way. Their users are using either a specific centralized platform (provided by a third party) or their own platforms. Trust is needed between peers as it is not provided by the network. Confederated instances' administrators belong to specific entities. Administrators have then the possibility to change the behavior of the system without the user consent anytime. Trust, as seen in Table 1, is one of the main pain points that we are addressing in our work.

We propose a new model, based on a pure decentralized architecture, based on the Blockchain Decentralized Network. As seen in Fig. 3, the model also includes an enhanced version of the participants or peers ($Entity1 \ldots N$), as they are using semantic web ontologies combined with STIX™. The administrator concept does not longer exist as it will be a decentralized application (dApp).

The main components of each entity can be seen in Fig. 4. Each entity will have a reasoner to handle its own data, reasoning about when to share, what can be shared, which investment can be done. At the same time, it will work the other way around, that is, what information is useless to spend cash (or tokens) on it. In the next subsection we will give more details about the different building blocks of the enhanced entities.

The new architecture of our model is addressing most of the problems seen in Table 1. It has some specificities, among them:

- *Trust* is provided by the network itself:
  Integrity Once the application is deployed, it runs

[5] https://www.us-cert.gov/tlp.

autonomously without any admin or human intervention in all the nodes (nodes run EVM: Ethereum Virtual Machines).
Validations all transactions are validated within the network using cryptographic algorithms.
Immutable register There is a transparent and immutable register. The network use Merkle trees and other advanced cryptographic features to protect it.
Code transparency The application does what it has been coded. The running code (ABI) can be checked within the source code. High level of transparency is then provided.
Availability is provided by the nodes. Nodes are running by network incentives. All transactions within the blockchain have economic incentives for nodes. There is no need to run an owned infrastructure so there are not ongoing costs.
PKI users are using wallets which represent the public keys in a PKI infrastructure. Private keys will then be needed to execute any transactions and only those private keys will be allowed to make transactions with those wallets.
Time-stamping of each transaction. In CTI, one of the issues is related to the reliability of the information [1]. All transactions will be time-stamped, within the immutable register of Blockchain, to keep track of who shared what and when.

- We will create a token. It is a crypto-feature which inherits all above characteristics to provide trust.
- There can be multiple decentralized applications interacting between themselves, all of them will be running in the network (in all of the nodes).
- The interoperability between decentralized applications (which is inherent to the system), can be understood as a confederated topology where different CTI Marketplaces can coexist and interact between themselves.
- There can be malicious nodes as noted in the figure. For simplicity, we do not consider those which try to create their own blockchain. We will consider, as malicious, those nodes that will try to get as much profitability from the system but also from the rest of users as possible.
- It has very low cost (just gas), as there is no infrastructure needed to deploy our information sharing application. Costs will be associated to transactions fees.
- Investments and payments are proposed as economic incentives. As described by authors in [4], just a simple reduction of the fee, was enough to provide incentives to users. In our case, they will have incomes, the more they share (with quality), the more it will be consumed. Each time the CTI Data shared is consumed, the producer of that data will receive incentives. There is also a new opportunity for non cybersecurity experts, to invest in the CTI token, as it becomes a trading opportunity within the cybersecurity arena.

In addition to this, the use of Ethereum Blockchain, will allow us to provide economic incentives to better incentivize the users. It might help us to improve the dynamics of the information sharing as well.

We will use the built-in feature to create "tokens", by creating the first CTI token as a digital asset to represent de CTI Data, data which is considered very valuable nowadays. We will use the ERC20 [16] standard to enable the interoperability between tokens. We could then be exchanging our CTI tokens with any other digital asset someday (e.g. gold or any other stock options).

3.2.2 An enhanced (semantic) version of the participating entities

In this section we will cover the rest of the open challenges seen in Table 1. We propose a solution to exchange rules, based on semantic web ontologies, semantic web rule language (SWRL), semantic reasoners and a OWL version of STIX™. It might contribute to solve most of the issues related to expressiveness, effectiveness, efficiency, anonymity, interoperability, automation and inference.

For that, we propose that all entities need to be enhanced peers, working with the same ontologies, as seen in Fig. 4. An enhanced entity has the following building blocks:

- A *semantic reasoner*, as the key element of all entities. Each reasoner of each entity will handle the inbound and outbound traffic as well as all internal transactions (within the same entity).
  Inbound traffic: the reasoner will decide what CTI Data is interesting to retrieve. Then it will use it to infer new knowledge.
  Outbound traffic: the reasoner will decide what CTI Data is interesting to share, after checking that the data is eligible (green or white TLP). It will also decide what data will be interesting to be shared (because its interest but also because its potential value in the system).
  Internal transactions: the reasoner will handle all CTI and Risk Data together allowing new relationships to infer knowledge dynamically.
- A *semantic data set*, which represents all the data known by each entity. Data coming from external sources like the CTI Marketplace will be used to infer new knowledge.
  The CTI Data is always under semantic web format (OWL, SWRL). The (semantic) info shared must be stored within the Blockchain. Each entity will have their own CTI knowledge base (because of their knowledge and interest).
  Risk data imports CTI Data. It enables the combination of both domains within the same SWRL rule, as an example to provide enriched knowledge.

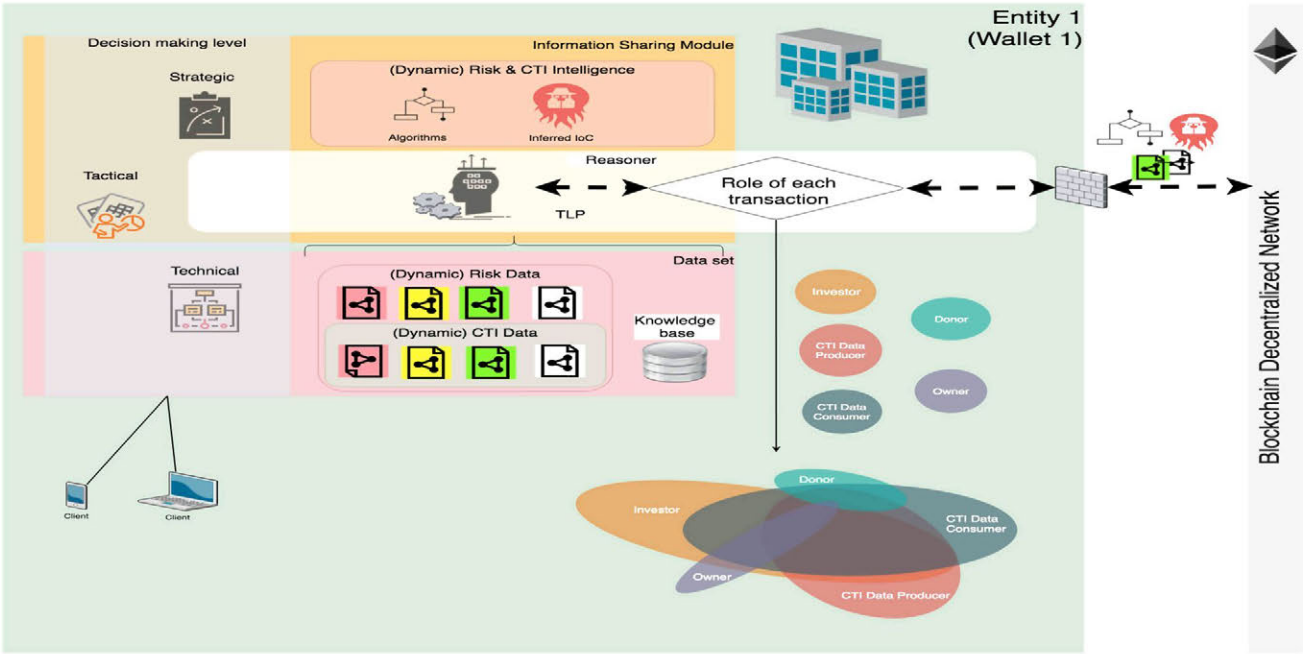


Fig. 4 Proposal of a new model of "Entity"

The CTI Data can be IoC (as usual) but we propose to support algorithms in the format of SWRL (beyond IoC) as a much better solution (anonymization, less ephemeral knowledge transfer, etc.).

The Traffic Light Protocol (TLP), understood as the restrictions on data redistribution, can also be handled and used by the reasoner.

- A *Dynamic Risk and CTI Intelligence module*: it is the advanced intelligence, the knowledge of analyzed and processed data beyond the data itself.

  Risk intelligence: risk advanced knowledge, in the format of algorithms, rules or even inferred data. As an example, a rule to define new investments needed or awareness training sessions depending on the dynamics of threats. That rule can be interesting for another entity to improve the automation of their information security management systems.

  CTI intelligence: CTI advanced knowledge, again, in the format of algorithms, rules or even inferred data (e.g. possible attribution of a cyber attack to a threat actor). These types of reasoning axioms can be interesting to other peer.
- *Pseudo-automated decision making level*: thanks to the contribution of the reasoner, it can be useful to infer, classify and automate the knowledge for each of the decision making levels.

  Strategic: e.g. recommended investments or partnerships, to reduce risks. This can be modeled in algorithms by using this enhanced version of the language.

  Tactical: e.g. dynamic situational awareness and tactics (when to share, what to share, what must be under TLP, at what level, etc.). In this case, the reasoner can handle optimal investments decisions in the CTI token as another example.

  Technical: e.g. detection and behavioral patterns, updating security policies (e.g. password length).
- *Role of each transaction:* all entities can (and probably will) work under different roles (CTI Data producer, CTI Data consumer, Investor, Donor); however only one can be the owner of each smart contract. The reasoner will also contribute to any decision to be taken related to each transaction (e.g. how much to invest, in case of an investment is recommended).
- *Wallet and connection to the blockchain:* all entities 1..*N*, are connected to the same blockchain through their own Wallets using either public or private Ethereum nodes.

As we propose to exchange algorithms, or rules, beyond IoC, some examples of SWRL algorithms can be seen in the pseudocode of Algorithms 1 and 2.

In order to demonstrate the potential of this model, we provide a piece of a more complex CTI algorithm, which is using an OWL/XML syntax. "Stix2" prefix, seen in "abbreviatedIRI", is the OWL version of the current STIX™v2 object. "Drm" prefix belongs to a different ontology related to dynamic risk management, which is interoperable within the stix2 ontology.

```
<DLSafeRule>
        <Annotation> <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <Literal>ThreatIntelligence#1 Security Event Dropper Detection</Literal>
        </Annotation>
        <Body>
            <ClassAtom>
                <Class abbreviatedIRI="stix2:NetworkTraffic"/>
                <Variable IRI="urn:swrl:var#nt"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty abbreviatedIRI="stix2:dstPayloadRef"/>
                <Variable IRI="urn:swrl:var#nt"/>
                <Variable IRI="urn:swrl:var#pl"/>
            </ObjectPropertyAtom>
            <ClassAtom>
                <Class abbreviatedIRI="stix2:Artifact"/>
                <Variable IRI="urn:swrl:var#pl"/>
            </ClassAtom>
            [<DataPropertyAtom>
                <DataProperty abbreviatedIRI="stix2:mimeType"/>
                <Variable IRI="urn:swrl:var#pl"/>
                <Literal>javascript</Literal>
            </DataPropertyAtom>
            [...]
            <ClassAtom>
                <Class abbreviatedIRI="stix2:URL"/>
                <Variable IRI="urn:swrl:var#red"/>
            </ClassAtom>
            [...]
            <DataPropertyAtom>
                <DataProperty abbreviatedIRI="stix2:extensions"/>
                <Variable IRI="urn:swrl:var#pl2"/>
                <Literal>windows-pebinary-ext</Literal>
            </DataPropertyAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class abbreviatedIRI="drm:SecurityEvents"/>
                <Variable IRI="urn:swrl:var#x"/>
            </ClassAtom>
            [...]
                <Literal>Dropper behavior of Malicious Windows Executable</Literal>
            </DataPropertyAtom>
        </Head>
    </DLSafeRule>
```

Example of a CTI rule or algorithm in the format of SWRL and STIX™v2 to define a TTP detection

---

4 Design

4.1 Roles

The main roles of the CTI Marketplace will be the following (all of them have associated functions for their type of interactions):

- *CTI Data producer:* an individual or a group which might be using a single Ethereum address to contribute or share (write) CTI data rules in the CTI Marketplace Smart contract.
- *CTI Data consumer:* an individual or a group which might be using a single Ethereum address to consume (read) CTI data rules from the CTI Marketplace Smart contract.
- *Investor:* an individual or a group which might be using a single Ethereum address to invest into the CTI token.
- *Donor:* an individual or a group which might be using a single Ethereum address to donate ether to support the CTI Marketplace.

– *Owner:* an individual or a group which might be using a single Ethereum address to launch each CTI Marketplace instance. In this case, the owner are the authors of this work.

Algorithm 1: Example of the pseudocode of a simple SWRL rule handling an OWL ontology

Data: W3C semantic OWL ontology: Classes, properties, variables and individuals.
Result: The SWRL (algorithm) is loaded as an axiom into the semantic reasoner. The reasoner will infer new knowledge over the OWL data
SWRL syntax **begin**
  $Antecedent(Body) \longrightarrow Consequent(Head)$;
  /* When all conditions in the "Antecedent" are met, all conditions at the "Consequent" are also met */;
Example **begin**
  $hasparent(?x, ?y)$ AND $hasbrother(?y, ?z) \longrightarrow hasuncle(?x, ?z)$;
  // If your parents, have brothers; all of them will be your uncles;
  // Variables are represented by "?";
Example in pseudocode **begin**
  while *(ont1:hasparent(?x,?y)* **and** *ont1:hasbrother(?y,?z))* **do**
    $ont1 : hasuncle(?x, ?z)$;
    /* The reasoner will infer all "uncles" of all "people" in the data set */;
    /* Note: "ont1" is the prefix of the ontology used. Different ontologies can be used within the same SWRL rule */;
    // Note: We use "while" as pseudocode, as the axiom is always active;

Algorithm 2: Example of the pseudocode of a simple CTI rule / algorithm in the format of SWRL and an OWL version of STIX™v2

Data: CTI Data under an OWL version of STIX™
Result: The reasoner will infer new knowledge over CTI Data
OWL STIX™ example **begin**
  $stix2 : IPv4Addr = ?ip$ AND
  $stix2 : belongToRefs(?ip, ?ref)$ AND
  $stix2 : DomainName(?ref) \longrightarrow$
  $stix2 : DomainName(?ref)$ AND
  $stix2 : resolvesToRefs(?ref, ?ip)$;
OWL STIX™ example in pseudocode **begin**
  while *(stix2:IPv4Addr = ?ip* **and** *stix2:belongToRefs(?ip, ?ref)* **and** *stix2:DomainName(?ref))* **do**
    $stix2 : DomainName(?ref)$ **and**
    $stix2 : resolvesToRefs(?ref, ?ip)$;
    /* The reasoner will infer all "DNS matches" from "RDNS (Reverse DNS)" in the data set */;
    // Note: We use "while" as pseudocode as the axiom is always active;

Above roles can be combined under the same single Ethereum address. As an example, an organization could contribute with its own CTI data rules at the same time it consumes data from other organizations. It can also becomes an investor. As seen in Fig. 5, we propose a paradigm shift, where we expect a significant increment in the number of active CTI Data producers. Most of them are today just consumers. Becoming CTI Producers, they will be able to also become investors. They will have special investment conditions thanks to the introduction of the Tax per transaction and the CTI Token.

4.2 Market growth: Montecarlo simulation of new incentives

In order to better design the system, we have made theoretical calculations as seen in Figs. 6 and 7. We used multiple Montecarlo simulations to forecast the potential market growth, in case of introducing our new incentive model. It helped us to better design and select which are the key variables of the model. In addition to this, it enabled us to evaluate it versus current legacy information sharing platforms (OSINT, Feeders and ISAC), taking into consideration the specificities of each role.

As a preliminary condition, we used the CTI Data volume of the top 20 types of data which are described in MISP [12], as a baseline for all systems. In our case, as it is oriented to go beyond the exchange of IoC, it will have additional CTI Data, which are in the format of algorithms. At the same time, our system will be used to handle data from 3 different levels (technical, tactical or strategic), as seen in Fig. 4. We then estimated the additional CTI data by using the inverse of the normal cumulative distribution. We adjusted its parameters with a 15% probability, a mean of 33% of the baseline volume of data and a standard deviation of 10,000.

Due to the fact that the investment by CTI producers is optional, we will consider that the volume of optional investments are an inverse of a lognormal distribution applied to the 50% of total data in the system. We are also considering that all the data is read at least a hundred times. We will consider the worst case scenario to read all the data (by CTI Data consumers), that is to say, each data is read in a different transaction (each transaction has an associated fee). We then applied the same condition for writing: each single piece of data uploaded to the system, will be uploaded in a single transaction.

We introduced a tax for CTI Data consumers each time they execute a read transaction, as they get important value back. For simplicity, we fixed the 50% of such tax will go for the CTI producers wallets and the 10% to the owner's wallet of the system. The number of CTI tokens provided

Fig. 5 Roles overlapping

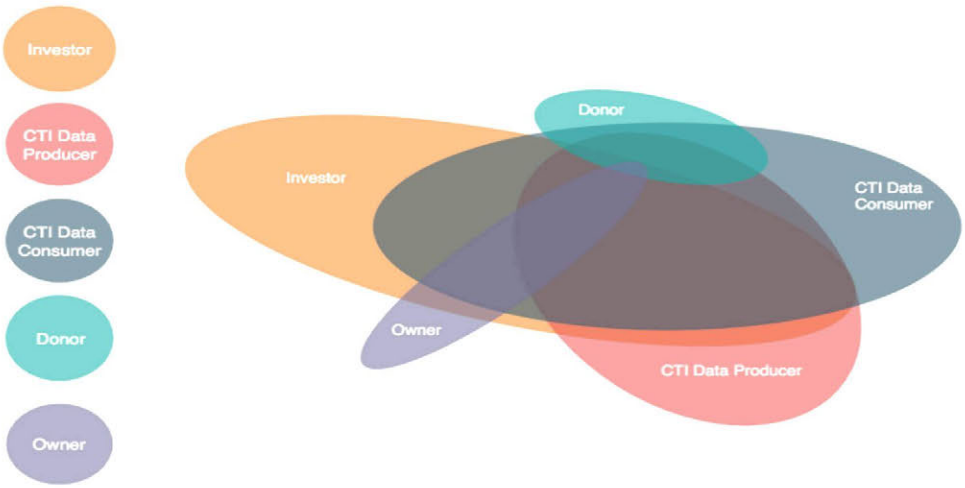


Fig. 6 Montecarlo simulation of benefits with an evolutionary tax and CTI token value (p_c)

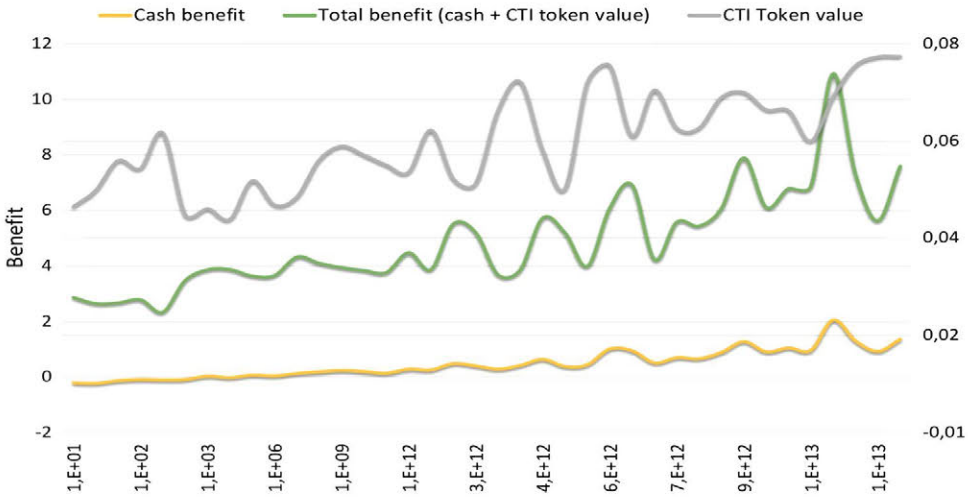
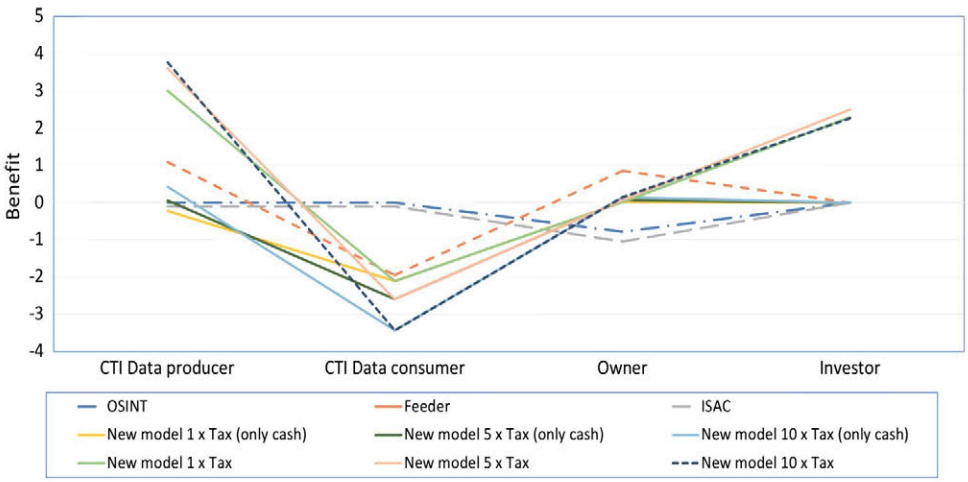


Fig. 7 Montecarlo simulation to compare our new model versus current models. We include some Montecarlo simulations of our model in the comparison



to investors will be different depending on the role. To simplify our calculations, we used the worst case scenario in our Montecarlo simulations, that is, we used the ongoing incentive factor of 20 for CTI data producers and 10 for a standard investor. The smart contract will keep the remaining cash (the remaining 40%).

We will define the dynamic or evolutionary benefit B at time t of a entity x as seen in Eq. 1

$$B_t(x, p_c) = I_t(x, p_c) - C_t(x, p_c) \quad (1)$$

where p_c is the token prize at time t, I_t is the income (gross benefit without cost) at time t, item C_t is the cost at time t.

The gross benefit or total income I_t of an entity x, can be defined as seen in Eq. 2:

$$I_t(x, p_c) = IC_t(x, p_c) + ITK_t(x, p_c) \quad (2)$$

where IC_t is the income made by cash at time t, ITK_t is the income made by the CTI token value at time t.

In our proposal, CTI Data producers and the owner, will receive cash (IC_t), due to the introduction of taxes into each CTI read operation. At the same time, these payments will also represent variable costs VC_t for CTI Data consumers. CTI Data producers will also receive tokens ITK_t in case they decide to invest when uploading CTI Data to the system. We have an evolutionary token value p_c, depending on the balance within the smart contract. The smart contract will sum the cash of all the investments (made either by investors or CTI Data producers) to the 40% received from the applied taxes.

On the other hand, the total cost C_t of an entity x, can be defined as seen in Eq. 3:

$$C_t(x, p_c) = FC_t(x, p_c) + VC_t(x, p_c) \quad (3)$$

where FC_t is the fixed cost at time t, VC_t is the variable cost at time t.

To simplify our calculations, we propose zero fixed costs ($FC_t = 0$), due to the use of a decentralized infrastructure. Variable costs VC_t will be associated to any transaction. We have two types of variable costs: blockchain network fees (gas) and taxes. The gas used to write (store) CTI rules is much higher than the gas used to read (query) those rules. In order to calculate these type of variable costs, we implemented, deployed and evaluated a draft smart contract (see Sect. 6 for more information).

Equation 1 can then be divided into different equations as seen in 4:

$$B_t(x, p_c) = BC_t(x, p_c) + BTK_t(x, p_c) \quad (4)$$
$$BC_t(x, p_c) = IC_t(x, p_c) - CC_t(x, p_c) \quad (5)$$
$$BTK_t(x, p_c) = ITK_t(x, p_c) - CTK_t(x, p_c) \quad (6)$$
$$I_t = IC_t(x, p_c) + ITK_t(x, p_c) \quad (7)$$
$$C_t = CC_t(x, p_c) + CTK_t(x, p_c) \quad (8)$$

where BC_t is the cash benefit at time t, BTK_t is the token benefit at time t, IC_t is the cash income (cash gross benefit) at time t, ITK_t is the cash income (token gross benefit) at time t, CC_t is the cost paid in cash at time t, CTK_t is the cost paid in tokens at time t.

We made a Montecarlo simulation (seen in Fig. 6) to define the minimum tax value of our system equivalent to a breakeven calculation. In this case we look for a positive **cash benefit** $BC_t(x, p_c) > 0$ of the CTI Data producers.

In order to study how our evolutionary incentives (especially tokens through investments), are also contributing to the dynamics of the system, we also used a Montecarlo simulation, as seen in Fig. 7. In the figure, we see a comparison of our model versus other current approaches (OSINT, Feeders, ISAC), including benefits per type of role. We marked with discontinuous lines the OSINT, the Feeder and the ISAC values, as well as one selected from our model (the "New model 10 × Tax"). Our evolutionary model is anyway represented by different curves which are using different tax values as parameters. There are two types:

- "New model [1, 5, 10]× Tax": to represent the evolutionary benefit B at time t of a entity x (equivalent to $B_t(x, p_c)$),
- "New model [1, 5, 10]× Tax (only cash)": to represent the evolutionary cash benefit BC at time t of a entity x (equivalent to $BC_t(x, p_c)$)).

As a result, the CTI token is especially contributing to the benefits of the CTI Data producers ($B_t(x, p_c)$ vs. $BC_t(x, p_c)$). Furthermore, it is also contributing to the engagement of new key roles of any business: the investors. They will be able to invest in Cybersecurity CTI Data sharing, easily. Our proposal, then, provides new ways of getting value back from the CTI Data to several roles simultaneously.

4.3 CTI token

Despite the great number of ICO[6] (initial coin offering), there are still several open discussions about the benefits and the economic model of tokens [26]. Our proposal slightly differs from a typical ICO. In our proposal, we propose a CTI token to represent the value of the CTI Data, as its digital asset representation. We also propose a customized version of an ongoing ICO, because the very first time a new CTI producer joins the network with new CTI Data, it has a special offer to invest in CTI tokens. After that, they are still getting more tokens than standard investors, as a way to keep attractive incentives associated to the sharing of new CTI data.

As described by the author in [26], the pricing service in a fixed token rate will correctly capture the proportionate relationship between demand and price, the greater the demand for the service, the more people will buy tokens, the price of tokens will increase, and so will the cash equivalent of the service; and vice versa. The network has honest but also malicious miners (as seen in Fig. 3), each having

[6] https://www.coingecko.com/en/ico.

different motivations. Malicious miners, want token price to depreciate, while honest miners and investors want token price to increase. The incentives should be modeled in such a way that no player will want to deviate from the honest conditional equilibrium. The author proposes a version of the Cobb-Douglas utility function to model the utility of a service, based on the quality but also the popularity of the service. In our case, we propose new parameters for the utility function to model our service as seen in Eq. 9.

$$U(tp_t, rd_t) = (1 + E[tp_t]^{\alpha}) \cdot E[rd_t]^{\beta} \tag{9}$$

where $E[tp]$ is the expected volume of trusted experts contributing (trusted CTI producers) in the system and $E[rd]$ is the expected reliability and accuracy of the data (CTI Data).

Note that even when there is zero number of trusted experts ($E[tp] = 0; E[rd] > 0$) the customer still receives a non-zero utility from the service. Conversely, when the service is very popular among trusted producers and has high demand but zero reliable data ($E[tp] > 0; E[rd] = 0$) the utility from the service is zero despite its popularity.

Our proposal is similar like the one proposed by the author in [26] about Investors. They are people who buy our tokens for the CTI market value and hold them in expectation of value appreciation. They are not really interested in the Threat Intelligence Data but its market growth (as seen in Figs. 6 and 7). We will also assume that there are N identical investors with equal wealth level w who are using a version of Markowitz's mean-variance formula (see Eq. 10) to choose between investing in stocks and our CTI token. We will assume that all investors are risk neutral. Because the more investors a system has, the more cash we will have in our system.

We also used the inverse of the lognormal cumulative distribution for our Montecarlo simulations to simulate ongoing investments.

$$\alpha_t = \frac{\mu_c - \mu_s}{\gamma \cdot \sigma_c^2} \tag{10}$$

where μ_c and μ_s are coin and stock returns respectively, and γ and σ_c are risk-aversion coefficient and coin volatility respectively.

Malicious miners are nodes who buy tokens and vote at quorum to force their version of blockchain. To simplify calculations, we do not consider worst types of malicious miners, which are those that fight to create their own blockchain. We are considering only those who want to maximize their profits within the system. That is to say, we consider those nodes that will try to act as major investors when the prize of the token p_c is as lowest as possible.

We decided to simplify the token prize equation suggested by Ciaian et al. [27]. In our simulations we will use the equation seen in 11. Also for simplification, we will not limit the number of tokens NT as it might happen with proof of stake setups although it is a feasible alternative.

$$p_c = \eta \frac{Bal}{\mu_s \cdot NT} \tag{11}$$

where Bal is the balance of the smart contract (cash), $\mu_s = 1$ which is the opportunity interest rate (stock returns), NT is the volume of the token supply (number of minted tokens), $\eta = 1$ which is an exogenous stabilization parameter related to the token attractiveness.

Despite the simplification for our calculations, we expect that our model will improve the dynamics of the threat intelligence sharing as seen in Fig. 7.

4.4 Architecture

Ethereum is a peer-to-peer network of nodes running Ethereum Virtual Machines (EVM) that stores a copy of all the data and code on the blockchain. Ethereum network is then a blockchain with dual use, it supports the integrity, reliability and availability of the transactions of Ether (ETH) crypto currency as well as it supports the integrity, reliability and availability of the execution of Smart contracts.

Decentralized applications (DApps) can be deployed as running code in all the Ethereum nodes at the same time under the name of Smart contracts. Ethereum nodes use, as Bitcoin (BTC), consensus algorithms to validate transactions however in Ethereum, transactions can go beyond money transfers. They can use transaction payloads to enable the possibility to execute applications. Transactions are now resolving read and write access to a running application which is running in all nodes at the same time, decentralized. Validation is then confirming that executions in certain DApps are validated by a consensus algorithm likewise that the same validation algorithms are validating money transfer.

All the network is optimized in such a way that each instruction or assembly call has associated a certain amount of gas, which is the equivalent to the use of resources needed (e.g. CPU) for each instruction. The idea is to have the most efficient running code, if not, a DApp could not finish each execution if it runs out of gas. Our proposal to solve above problems and take suggested opportunities is based on the use of Ethereum Blockchain as a decentralized infrastructure to run our decentralized application (DApp) created for the purpose of this work. The DApp consists on a Smart contract that will handle the storage and exchange of CTI data in the form of a Marketplace of rulesets (e.g. TTP detection algorithms) as described by authors in [9].

An online marketplace is a type of e-commerce site where product or service information is provided by multiple third parties, whereas transactions are processed by the marketplace operator. Online marketplaces are the primary type of multichannel e-commerce and can be a way to streamline the

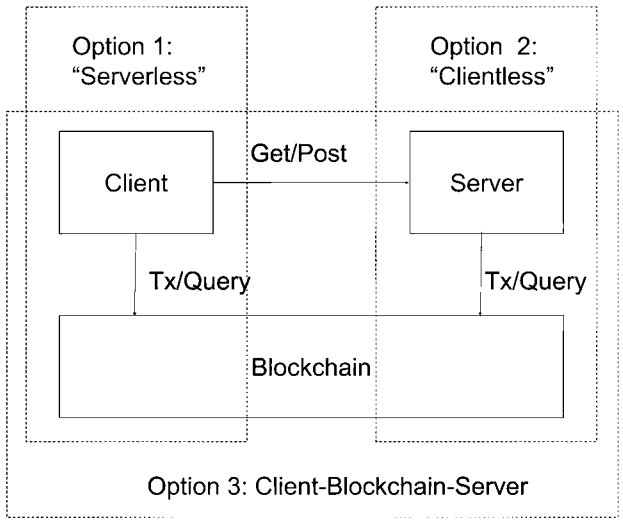


Fig. 8 Architecture design options: 1. Client-Blockchain (Serverless), 2. Server-Blockchain (Clientless), 3. Client-Blockchain-Server

production process. In our case, the product itself is the CTI Data ruleset (threat and risk intelligence algorithms).

The Smart contract provides new incentives to all type of users. Between them, a new ERC20 standard token is created, named CTI token. It represents a digital asset with special value and is related to the value of the CTI data ruleset. In addition to this, the Smart contract is coded in such a way that economic incentives are amplified to CTI Data producers. New roles are created as well, these are investors and donors.

In order to create the Marketplace within the Ethereum Blockchain there are different options and topologies as seen in Fig. 8:

- *Client-Blockchain (serverless)*
- *Server-Blockchain (clientless)*
- *Client-Blockchain-Server*

With regard to the first option **Client-Blockchain (serverless)**, the client interacts directly with the blockchain. In order to do that, the client needs to connect to an Ethereum node via a web3 provider that will handle the connection. There are lightweight clients like Metamask[7] or Mist with their nodes.

If there are no clients available, a public new "Geth or Parity" node would have to be deployed as a gateway disabling personal API to manage accounts for security reasons. As an alternative, Infura[8] nodes are available for free as well. Transactions via Metamask will ask for approval immediately. If not using Metamask our application should be monitoring all smart contract's events to update the end user interface.

With regard to the second option **Server-Blockchain (clientless)**, the solution is to install a local Ethereum node in order to use its RPC JSON interface from the server application to execute the operations in the Blockchain. Its RPC JSON interface should not be accessible outside our application, otherwise anyone could access and steal our funds. If we use public nodes, we could make transaction signatures offline. If nodes are not trusted to always deliver transactions to the network, we could mitigate this risk by delivering to more than one node at the same time.

Last option is the **Client-Blockchain-Server**, as seen in Fig. 9, it requires more coordination between the different interactions from the client or the server within the Blockchain network. Observations are based on event listeners where an efficient management might be implemented. As the example presented in Fig. 9, clients will be interested to listen only to their related (filtered) events however, the server will monitor all smart contracts related to the application. Messages from client to server are only recommended as informative, as it is better to wait until the blockchain confirmations. By using servers, our application will have the possibility to use several off chain backend connections to enriched applications (e.g. email, external storage, etc.).

We also take the opportunity to inherit an ERC20 token standard implementation to create a new token named CTI token. By that, we will contribute with more incentives around this digital asset. Investors, like they do in a ICO (Initial Coin Offering) are then able to invest in CTI tokens.

For the purpose of this work, we selected the **Option1: Client-Blockchain (serverless)** topology due to simplicity and efficiency. We use Metamask to connect to Ethereum Network as an injected web3 provider. At the same time it allows us to deploy a complete implementation of the CTI Marketplace to test our work and all related incentives per each role defined.

5 Implementation

This section details specific building blocks that are considered interesting for reproducibility.

5.1 Environments

For the implementation we have developed an Ethereum Smart contract written in Solidity language which has been deployed into the public blockchain as a Decentralized application (DApp). We use Remix IDE[9] as a powerful online integrated development environment (IDE) to code and debug the decentralized application. We also use Metamask as a Wallet and as an injected web3 provider that is needed to connect to and interact with Ethereum network nodes.

[7] https://metamask.io/.

[8] https://infura.io/.

[9] https://remix.ethereum.org/.

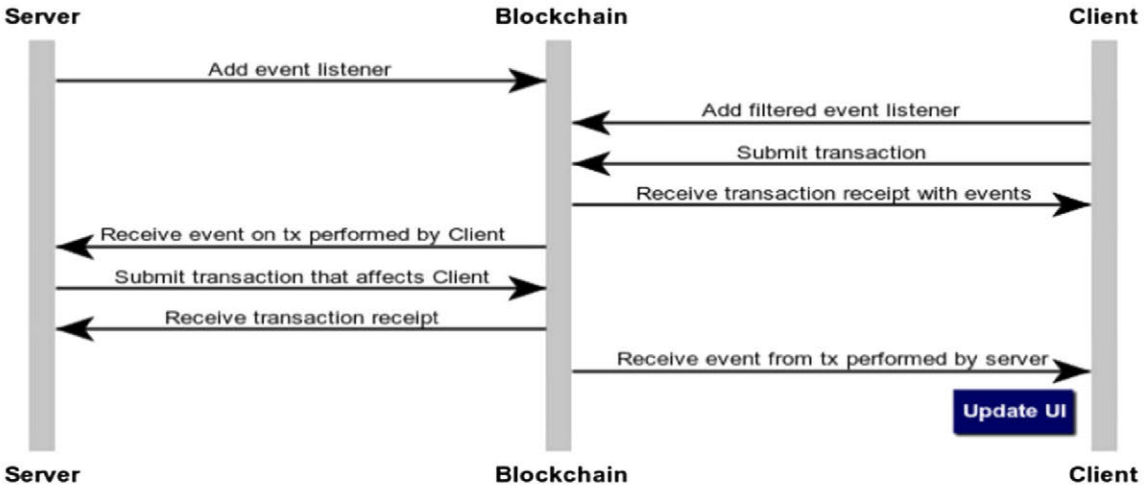


Fig. 9 Client-Blockchain-Server design option which might need enhanced synchronization

On the other hand, we have deployed and tested our implementation in two different blockchain environments:

- Development environment: A local and private Ethereum blockchain by using Truffle-Ganache framework.[10]
- Integration environment: A remote and public Ethereum blockchain by using one of the most used test blockchain networks named Ropsten.[11]

The main-net (Production environment) is equivalent to Ropsten with regard to the consensus and behavior, on the other hand it has more nodes but it also needs real ether. We then decided that using Ropsten was enough to demonstrate the benefits of our work, by using the main-net, the benefits will be higher for end users as there are more nodes to provide better and faster confirmations. Ropsten anyway can support all the features and CTI data from our work.

5.2 Smart contract (dApp) and the "CTI" token

With regard to our Smart contract, we named it "CTI Marketplace". In order to follow coding best practices, we extended an IERC20 (ERC20 token interface [16]). One of the benefits of using this ERC20 standard is the interoperability between tokens within Ethereum blockchain. Tokens are a way to represent any digital asset, once the interoperability is guaranteed within the Ethereum blockchain, there could be exchange between different tokens if they follow the same standard implementation like the one we provide in our work.

We created a specific token for the purpose of this work named CTI, the acronym of "Cyber Threat Intelligence" token. The token is created the very first time the contract is deployed. For that, we included its definition inside the **constructor** function as seen in the pseudocode of Algorithm 3.

Algorithm 3: Constructor and token initialization

Data: Constructor function executed just once: while our smart contract is being deployed
Result: Initializing variables
properties;
public ⟵ *visibility*
initialization;

```
owner = msg.sender /* Wallet address of the
source of the transaction who deployed the
smart contract (address)          */;
symbol = "CTI" // New token symbol (string);
tokenDescription = "Cyber Threat Intelligence Token"
/* Description of the new token (string) */;
```

A token enables the possibility to invest in it, it is equivalent to a digital asset, in our case a digital "Cyber Threat Intelligence" token asset, which value might be getting more and more interest in the market. More interest means more value, like any other asset in the market.

In the Fig. 10 a sequence diagram is shown to describe the process, messages and transactions between the browser, the Metamask Wallet, Ethereum nodes, the Smart contract itself and the CTI token, which is initialized as a ERC20 standard token.

Another coding best practice implemented, is the use of safe math libraries[12] written in solidity to implement math operations with safety checks that revert on error.

Our CTI Marketplace smart contract, will store CTI data which will be exchanged under certain conditions. It will also provide different incentives to each role. This time, we

[10] https://truffleframework.com/ganache.

[11] https://ropsten.etherscan.io/.

[12] https://github.com/OpenZeppelin/openzeppelin-solidity/blob/master/contracts/math/SafeMath.sol.

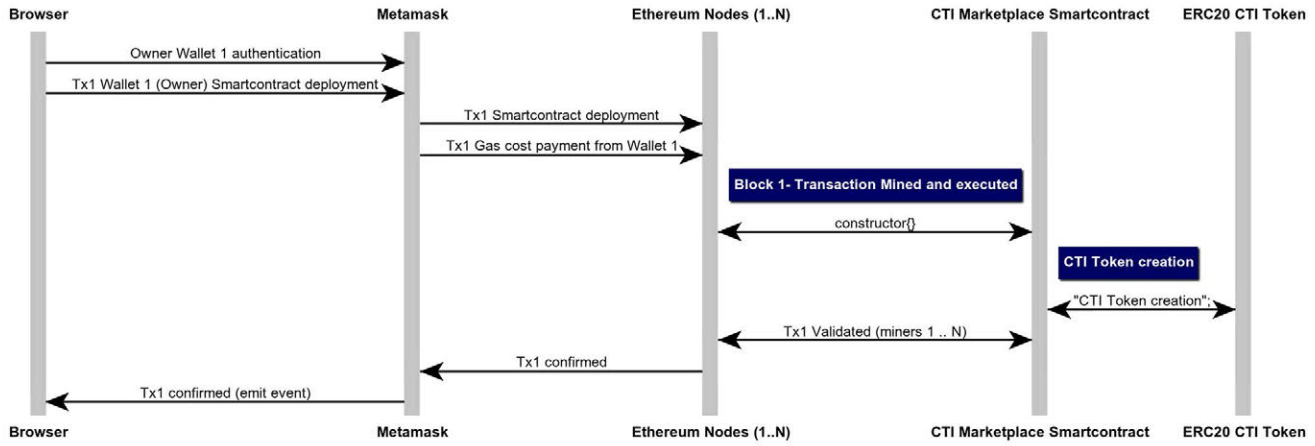


Fig. 10 Sequence diagram of the CTI marketplace smart contract deployment by its owner (wallet 1)

will implement the use case of sharing algorithms (e.g. TTP detection rules) beyond individual indicators of compromise (IoC) as proposed by authors in [9]. To store that information, we decided to use **mappings** which have some similarities to a key-value data store. At the same time, we decided to use a **struct**. Part of its data model is seen in the pseudocode of Algorithm 4.

Algorithm 4: Pseudocode of the data set structure to store CTI Data Rules in the Blockchain

```
Data: CTI Data from CTI Data producers
Result: Mapping between wallet addresses of each entity and
        their CTI Data (CTIRule)
 properties;
addressId ⟵ address // address of each entity;
entity ⟵ string // name of each entity;
rating ⟵ uint // rating of the entity;
ruleList ⟵ string[] // CTI Data rules;
ruleName ⟵ string[] // List of rules
description;
ruleConfidence ⟵ uint[] // List of rules
confidence;
initialization;
CTIStructs ⟶ address ⟶ CTIRule // mapping
addresses and rules;
balance ⟵ uint256 // economic balance;
```

In this case, each Ethereum blockchain address will belong to an individual or an organization. At the end, the address is the public key of a standard asymmetric encryption system, where each private key associated to each public key is only known by each of the owners. Each organization has a struct with its name, its rating by users or consumers about the quality of the entity as a CTI source, and at the same time, it will have an array of rules represented by **string[] ruleList** where each rule is representing an algorithm (e.g. TTP detection algorithm).

We used a string format to store each rule as it will be able to represent not only a SWRL rule but also any other less expressive rule language. We suggest the usage of SWRL rules formed by an antecedent and consequent together with CTI and DRM ontologies presented by authors in [9] to have actionable intelligence once the rule is downloaded (e.g. downloading a rule means a rule which is read or listed by calling the CTI Marketplace smart contract functions). If two organizations use the same semantic ontology [20], CTI concepts will be the same, and rules (using those concepts) will be working as plug and play, as they will be understood by their semantic reasoners [17]. Downloading a CTI Data rule means learning new knowledge by our semantic reasoner. CTI ontology provided by authors in [9] was based on STIX™v2.0 [18] as a de facto standard to work in cyber threat intelligence domain.

Each rule will have associated a **ruleName** which is the name or description of the rule (e.g. describing the objective of the rule) as well as a **ruleConfidence** representing the confidence of the rule which is related to the declared quality of each of the rules provided by the organization. Being the rules an array of strings, names and confidence of rules will also be arrays of string and unsigned integers respectively. Depending on the real confidence experienced by consumers using those rules, the **rating** of the organization will be higher or lower as consumers will have the opportunity to vote.

5.3 Business model and economic incentives

Wei is the minimum value of Ether. Gas is measured in Gwei so the ratio between the three of them is shown in equation 12.

$$1Wei = 10^{-18} Ether = 10^{-9} Gwei \tag{12}$$

As seen in the pseudocode of Algorithm 5 when a CTI Data producer is inserting a new rule in its mapped struct, (equivalent to share a new CTI data rule), it has the option to attach to the transaction any Ether (cash) to get special investor's con-

Algorithm 5: Pseudocode of the solidity function to share CTI Data by CTI Data producers

```
Data: Arrays of new CTI Data shared by CTI Data producers
Result: New CTI Data added to the storage structure of each
        entity, ready for reading
 properties;
payable ⟵ public // the function can receive
Ether by anyone;
addressId ⟵ address // address of each entity;
entity ⟵ string // name of each entity;
rating ⟵ uint // rating of the entity;
CTI Data addition (writing into storage);
for i ← 0 to NewCTIData.length do
    ruleList ⟵ newString[] // Adding new CTI
    Data rules;
    ruleName ⟵ newString[] // Adding new CTI
    Data rules description;
    ruleConfidence ⟵ newUint[] // Adding new CTI
    Data rules confidence;
Optional investment;
if is a new entity then // only for new entities
// Mint of tokens with special investment
conditions;
mint(msg.sender, msg.value∗1000∗NewCTIData.length);
emitUpdateStatus('A new user has shared first bulk of CTI
rules');
else // known entities
    // Mint of tokens with standard
    investment conditions;
    mint(msg.sender, msg.value∗20∗NewCTIData.length);
    emitUpdateStatus('An existing user has shared a new CTI
    rule');
```

ditions. A real transaction mined and executed in Ropsten blockchain, which is inserting 2 new rules simultaneously, is shown in Fig. 11. In this case, the CTI Data producer is attaching 10 Wei of cash in the same transaction. We used 10 Wei in our experimentation, as a reduced value that will not interfere in the calculation of the network cost. The final tax value, to be used in the production environment, will need to be guessed and fine tuned by using the Montecarlo simulations (as seen in Fig. 6). As paying is optional, special investment conditions are applied (20 tokens/each new rule). It is then receiving back 400 CTI ERC-20 standard tokens = 10 Wei * 20 tokens/each new rule * 2 rules added.

In this case, depending if it is the first time it shares or not, (there is a check for that condition in Algorithm 5), they will get different incentives. If it is the first time sharing, it will get 1000 tokens per each Wei sent with the transaction. From that moment, it will get 20 tokens (not 1000 but double of a standard investor which is getting 10) per each Wei sent associated to any new sharing.

As sending cash is not mandatory, being an investor (although it is optional), is a new incentive to CTI producers as well. This is one of the new contributions, as new incentives, of our work. CTI tokens will be created (using **mint** function from IERC20) in case the CTI Data producer takes this opportunity.

The function named insertBulkEntityRules (which pseudocode is seen in Algorithm 5), is defined as public as well as a payable function. The former characteristic of the function allows the reception of external calls from anywhere. The latter characteristic enables the reception of payments within the function call (for investments).

The mint of tokens uses an inherited function from the ERC20 standard named *mint* as seen in [16]. Its usage can be seen in our Algorithms 5 and 7.

With regard to **CTI Data consumer** role, this role has several functions available to consume CTI Data rules. All of them are related to the possibility to read from the key-value data store of our Smart contract.

In an example, knowing the (wallet) address of a CTI Data producer we will use the function seen in the pseudocode of Algorithm 6, to list its CTI Data rules. In order to do that, the CTI Data consumer is forced to send a minimum of 10 Wei of cash in order some taxes and benefits (as incentives for owner and CTI Data producer respectively), are applied.

Algorithm 6: Pseudocode of the solidity function used by CTI Data consumers to get access to CTI Data of a specific producer

```
Data: Input parameters: the address of the CTI Data producer
      AND payment of cash (msg.value > 10 Wei)
Result: A CTI Data consumer read CTI Data from a specific
        producer storage
 properties;
payable ⟵ public // the function can receive
Ether by anyone;
ifValue ⟵ modifier // filter to require a
minimum of 10 Wei;
CTI Data read (reading from storage);
if msg.value > 10 Wei then // requisite
    emitUpdateStatus('These are the rules of the selected
    entity");
    ruleList[address] ⟶ OutputString[] // Reading
    CTI Data rules;
    ruleName[address] ⟶ OutputString[] // Reading
    CTI Data rules description;
    ruleConfidence[address] ⟶ OutputUint[]
    // Reading CTI Data rules confidence;
else // error is thrown
 | revert
Economic Incentives;
owner.transfer(msg.value/10); // 10 percent of
taxes are sent to owner;
a.transfer(msg.value/2); // 50 percent of taxes
are sent to CTI Data producer;
```

The sequence diagram shown in Fig. 12 gives an overview of the process followed to list all CTI Data rules of a specific CTI Data producer. There are taxes that will be automatically transferred (in Ether) to the Owner and the CTI Data

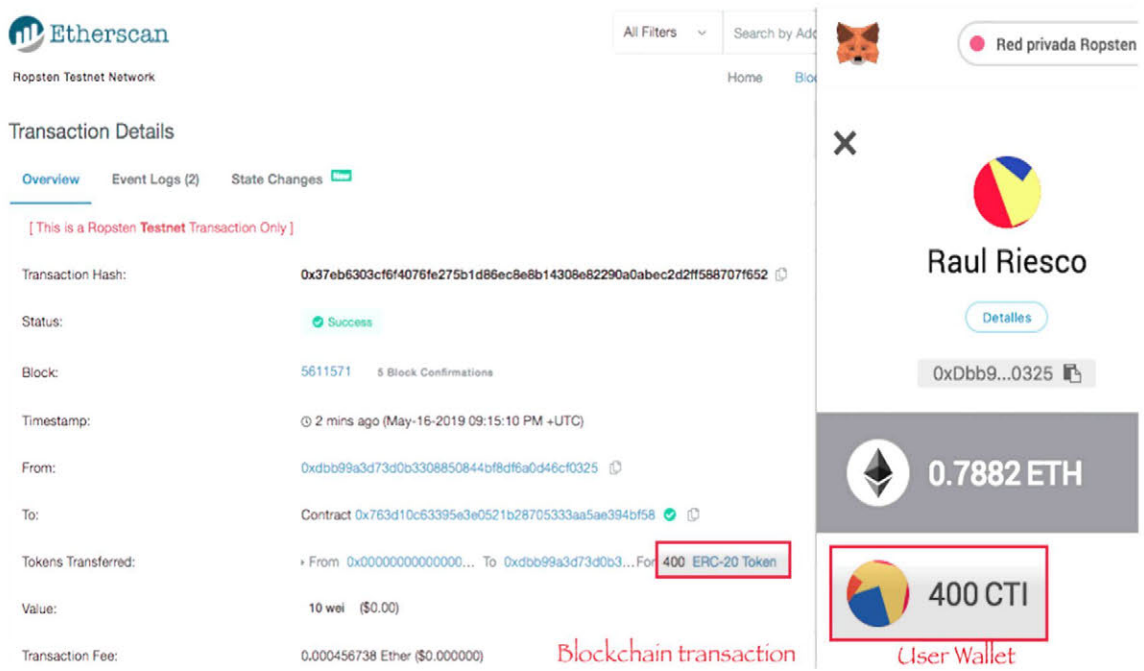


Fig. 11 Capture of a running-example where a CTI Data producer inserts two new rules simultaneously. On the right, the metamask end user Wallet is shown with the same received Tokens

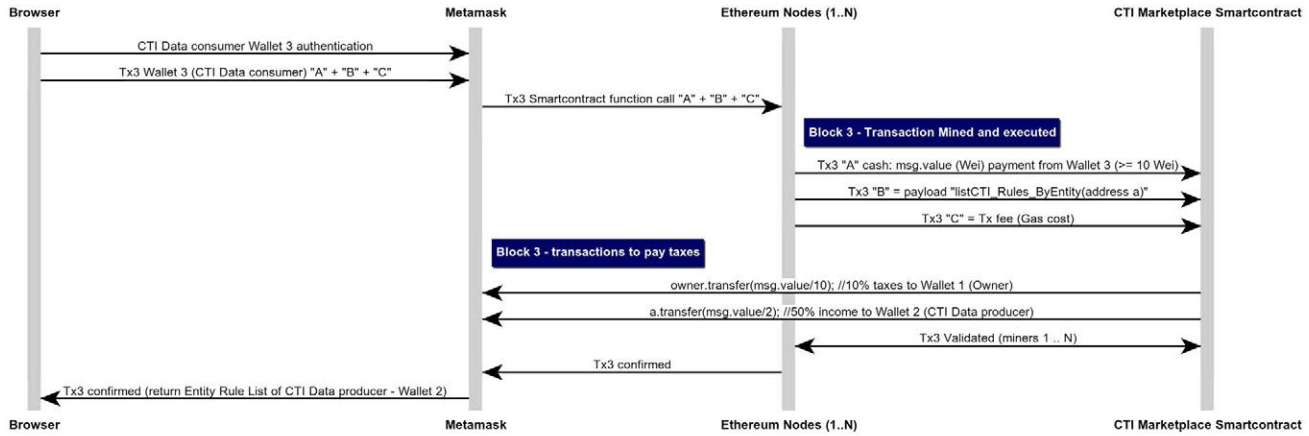


Fig. 12 Sequence diagram of a CTI Data consumer listing CTI data rules of a specific producer

producer, taxes paid by the CTI Data consumer. Owner and CTI Data producer will receive 10 and 50% of the value (msg.value) of the transaction respectively. The remaining 40% of cash sent will be accumulated within the Smart contract in order to boost the value of the CTI Token.

The role of **Investor** is available to anyone who sends cash to our Smart contract. If the cash is sent associated to a CTI Data producer transaction, it can get specific investment conditions as seen in the pseudocode of the Algorithm 5. If it is just a simple payment to our Smart contract by a standard investor, we implemented the fallback function to handle that payment converting it into an investment.

The fallback function (**function()**) without parameters is usually at the end of the solidity code of an smart contract. It will create an automatic investment in our CTI token on behalf of the sender address which has sent cash directly to our Smart contract. The pseudocode can be shown in the Algorithm 7.

In Fig. 13, a sequence diagram is shown to describe two different investment transactions:

– *Single Investor (Wallet 4)* is sending cash to the CTI Marketplace Smart contract which fallback function will handle the payment as an investment. Cash will be con-

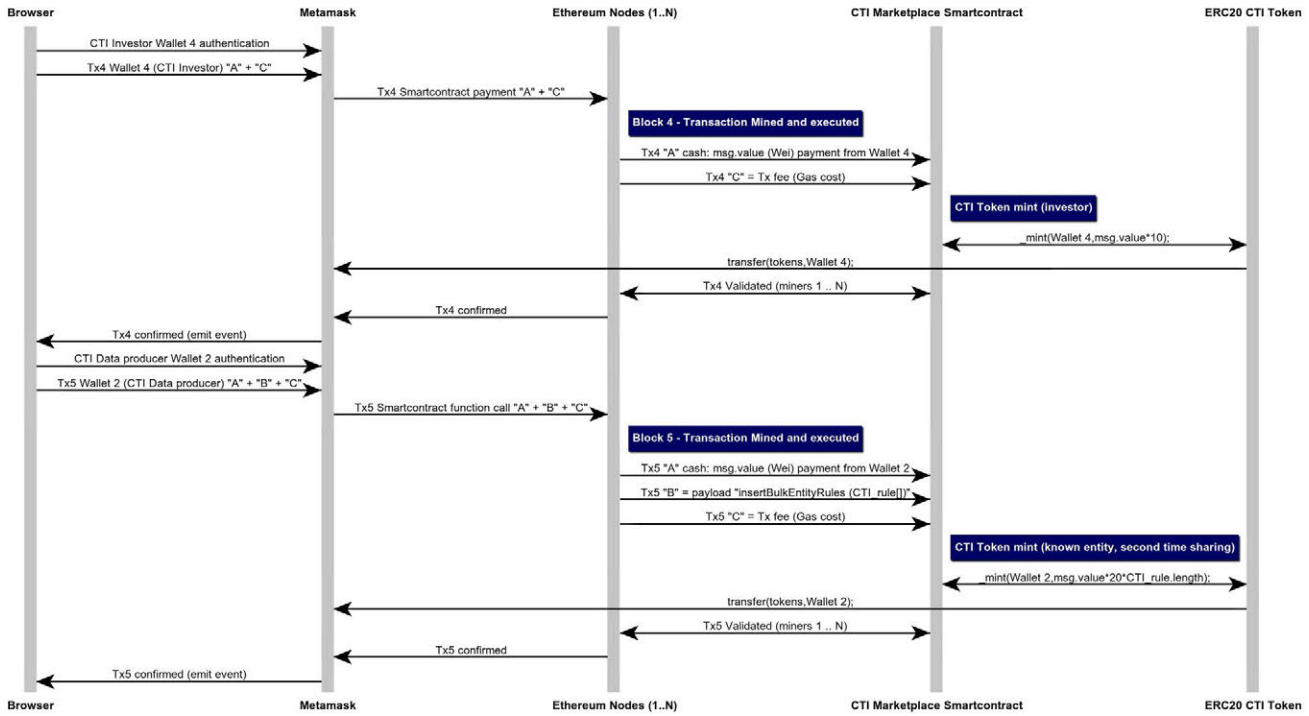


Fig. 13 Sequence diagram of two different type of investments: single investor (wallet 4) receiving 10 times the cash invested and a CTI Data producer which at the same time of sharing takes the opportunity of investing to get 20 times the cash invested in CTI tokens

Algorithm 7: Pseudocode of the solidity "fallback" function for investments in the "CTI" token

Data: Input parameters: payment of cash (msg.value > 10 Wei)
Result: An Investor receive back a number of CTI tokens proportional to the investment made
properties;
payable ⟵ *public* // the function can receive Ether by anyone;
ifValue ⟵ *modifier* // filter to require a minimum of 10 Wei;
Investment (getting CTI tokens from the smart contract);
if *msg.value* > *10 Wei* **then** // requisite
  tokens = *msg.value* ∗ 10 // 10 tokens per 1 Wei paid;
  mint(*msg.sender*, *tokens*) // tokens are created and automatically sent back to Investor;
else // error is thrown
  revert
Economic Incentives;
balance ⟵ *msg.value*;
// the smart contract itself is receiving and keeping the cash of the transaction;
// CTI tokens are interoperable under ERC20 standard. ;

verted in tokens by 10 times its value as seen in the pseudocode of the Algorithm 7.

– *CTI Data producer (Wallet 2)* is sending cash associated to a new exchange of CTI Data rules to the CTI Marketplace Smart contract. This time the function insertBulkCTIrules will handle the payment as an investment. Cash will be converted in tokens by 20 times its value (as it is a known CTI Data producer already).

Donor role will need to use a specific function to avoid an investment. The payable function to just make a donation is named **depositFunds()**, with no input parameter. In this case, the **Owner** of the Smart contract (the one who created and deployed it into the blockchain), will receive a donation for the CTI Marketplace.

In order to test our implementation, we deployed the CTI Marketplace in two different environments. As indicated earlier, the development environment is a Truffle-Ganache instance running locally and the integration environment is the Ropsten, which is one of the main (and more used) Ethereum Decentralized Test Networks, due to its similarity to the main net.

We implemented specific conditions along the code to better incentivize the CTI Marketplace's dynamics, by improving the appeal to contribute and to exchange new knowledge in the format of CTI Data rules. As described in the introduction, nowadays there is an asymmetric balance between consumers and producers, consuming data is far more frequent than producing or sharing it. Our work is focused on providing new ways to model this kind of CTI sharing by providing new incentives to have more producers.

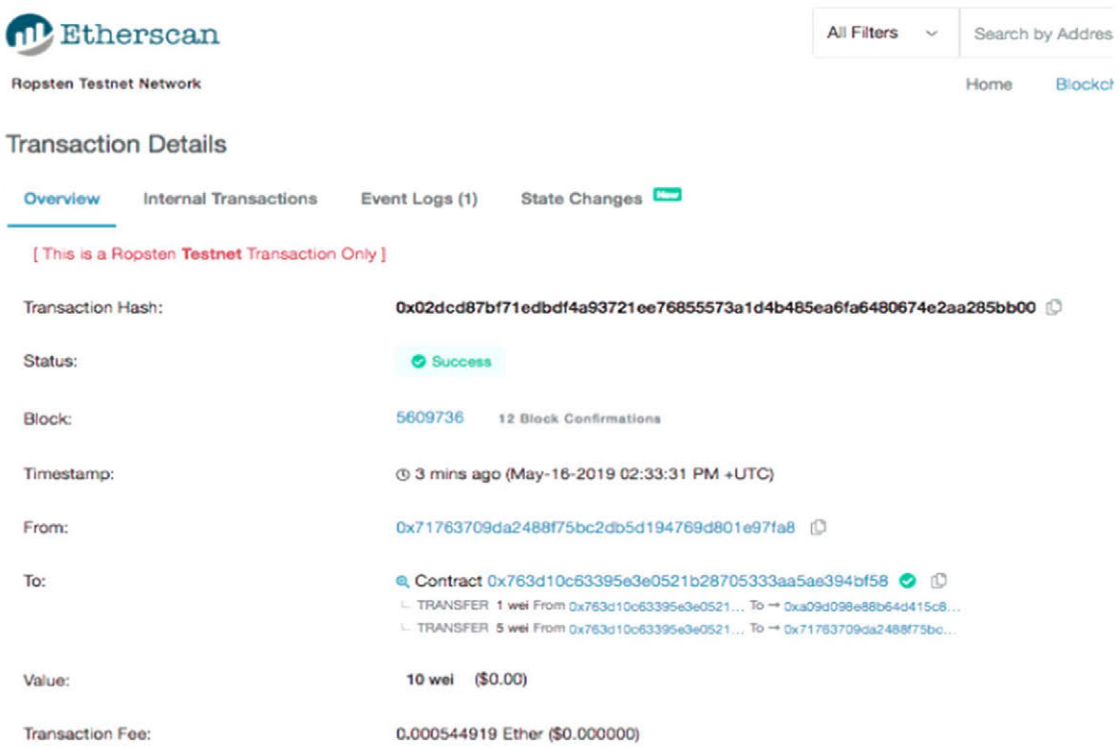


Fig. 14 Capture of a read/list (query) blockchain transaction by a CTI data consumer where taxes are applied. The owner and the CTI data producer are automatically receiving 10% and 50% of the cash sent by the consumer respectively

As seen for example in the pseudocode of the Algorithm 6, there are specific cash transfers stated as **owner.transfer** and **a.transfer** (where "owner" is the address who deployed the CTI Marketplace in the blockchain, and "a" is the address of the CTI Data producer). Each of these transfers are implementing automatic tax payments, when the function is called or invoked from a CTI Data Consumer. A real transaction receiving those tax payments is shown in Fig. 14.

Etherscan is an online service that provides access to all the immutable registers of the blockchain. In Fig. 14 we see the hash of the transaction, if it was successful, how many confirmations validated the transaction, in what block the transaction was handled, the time-stamp of the transaction, the source and the destination together with the amount transferred. The most interesting part is that the destination is a identified as a contract, not a wallet owned by a person. Then the contract, which is running in the blockchain, will handle all the business logic automatically when someone interacts with it. In our case, we see that the contract automatically ordered 2 payments or transfers, 1 Wei to the owner address and 5 Wei to the CTI Data producer which data has been read.

In this case, we decided to implement taxes to the function call, for that, we implemented a **modifier** which is a solidity specific function to force additional conditions to be met before the function is executed.

The condition is to send a minimum of 10 Wei when calling this function by a CTI Data consumer. The function will not work without it.

Then, we use this modifier named **ifValue** as part of all functions that are related to CTI Data consumer role. Our work is then contributing with new incentives for CTI Data producers as any (read only) access to their data will provide them with 40% of the cash used in the query. At the same time, they get special investment conditions while sharing new knowledge. At least 4 Wei will be then automatically transferred to the Wallet of the CTI Data producer when someone is reading its CTI Data rules.

In addition to this, the owner of the CTI Marketplace will receive 10% of the cash used. Then the remaining 40% will be kept within the CTI Marketplace Smart contract itself. The idea is to manage the token value along the time in order its value can be maximized and linked to the balance of the Smart contract.

Again, investors using the fallback (as seen in the Algorithm 7) will need to send a transfer equal or above 10 wei (because ifValue modifier is used). Remember that investors will get an amount of CTI tokens 10 times the amount invested. On the other hand, a CTI Data producer could send a value below 10 Wei when contributing, as the investment is optional, when sharing new knowledge. Moreover, producers

are getting at least 20 times the amount invested, as a special condition for those who are really sharing CTI knowledge.

With regard to donations, there is no minimum value associated with them. Any donation will be received, however this time, no tokens will be generated related to donations. Another modifier named **ifOwner** is used in some functions, to protect the balance of our CTI Marketplace Smart contract. Only the owner will have rights to withdraw cash from it.

6 Results and discussion

6.1 Limitations of our proposal

We identified two types of limitations of our proposal:

- *Limitations of the proposed model, based on the exchange of algorithms:* limitations due to the novelty of the proposed model.
- *Technical limitations:* technical limitations encountered in our implementation, as well as other potential technical limitations.

With regard to the **limitations of the proposed model**, we realized that current versions of STIX™patterning language still lacks of enough expressivity to define any kind of TTP. The proposed solution (as seen in Algorithms 1, 2 and the code of "An enhanced (semantic) version of the participating entities"), could improve its expressivity. Then, in case that the STIX™draft evolves into a more expressive standard language someday, it will become the main CTI de facto standard. Once that expressivity milestone is achieved, all stakeholders within the Cyber Threat Intelligence Exchange must use the same ontology or taxonomy.

The technical limitations are also subdivided in two subtypes:

- *Technical limitations to support algorithms.*
- *Technical limitations within the blockchain CTI Exchange.*

The limitations due to the novelty of the proposed model based on the exchange of algorithms have associated **technical limitations to support algorithms**. Once a new algorithm is received by a CTI Data consumer, their security devices, which are working in the IoC domain today, must be upgraded or replaced by new technologies which will be compatible with evolved standards. Reasoners today can be a perfect mate to fight against cyber threats, but they will need to interoperate with any security device under a common language.

In order to test the **technical limitations within the blockchain CTI Exchange** of our proposal, we decided to write (store) several CTI rules in bulk operations of different sizes, to check its effectiveness, its functionality, its limitations as well as the cost to store this type of CTI Data in the blockchain. On the other hand, we decided to read (query) several CTI rules in the same way, comparing the differences between writing and reading within the blockchain.

6.2 Technical limitations within the blockchain CTI exchange

We deployed the same CTI Marketplace Smart contract in both environments (development and integration) to run tests. The main difference between them is the Gas Block Limit within Ropsten (8M). This limitation is agreed within the network by miners/nodes. On the other hand we had no such limitation in our development environment (Gas Block Limit it was above 672M by default) (Fig. 15).

With regard to the gas used to write (store) CTI rules in the CTI Marketplace, it follows a linear growing pattern along the number of CTI rules included in the bulk transactions as seen in Fig. 16. At the same time, the amount of gas used to store CTI rules is very high, especially when more than 30 or 40 rules are included in the same transaction. At this point we find the first limitation due to the Gas Block Limit of Ropsten (integration environment). We do not have this limitation at the development environment. Above 50 CTI Rules, the gas needed is above Ropsten Block limit of 8M so it will through an exception out of Development environment due to the transaction runs out of gas.

If we set up a limitation programmatically, we can avoid running out of gas. The CTI Data producer will have a limitation to share more than 40 CTI rule in the same transaction however it will be able to store hundreds of CTI rules if using different transactions (each of them limited to 40).

We used the gas station[13] to calculate the cost of storage of our Marketplace which results are presented in Fig. 17. As a result, each CTI rule will cost us approximately 4 euro cents. We have to differentiate the efficiency between an IoC (indicator of compromise) to a CTI rule as described by authors in [9]. In principle less rules could detect more (matches by detecting patterns) than having individual IoC (e.g. DGA algorithm versus a specific domain). It means that we will need much less space to store rules than single IoC.

On the other hand, how much a company is willing to pay to demonstrate its authorship while sharing a rule to detect a malicious TTP pattern?. Using the blockchain, there are inherent benefits like using PKI, immutable registers, etc.

Beyond that, we think that CTI Data producers would be willing to invest 4 euro cents to share a rule that will

[13] https://ethgasstation.info/index.php.

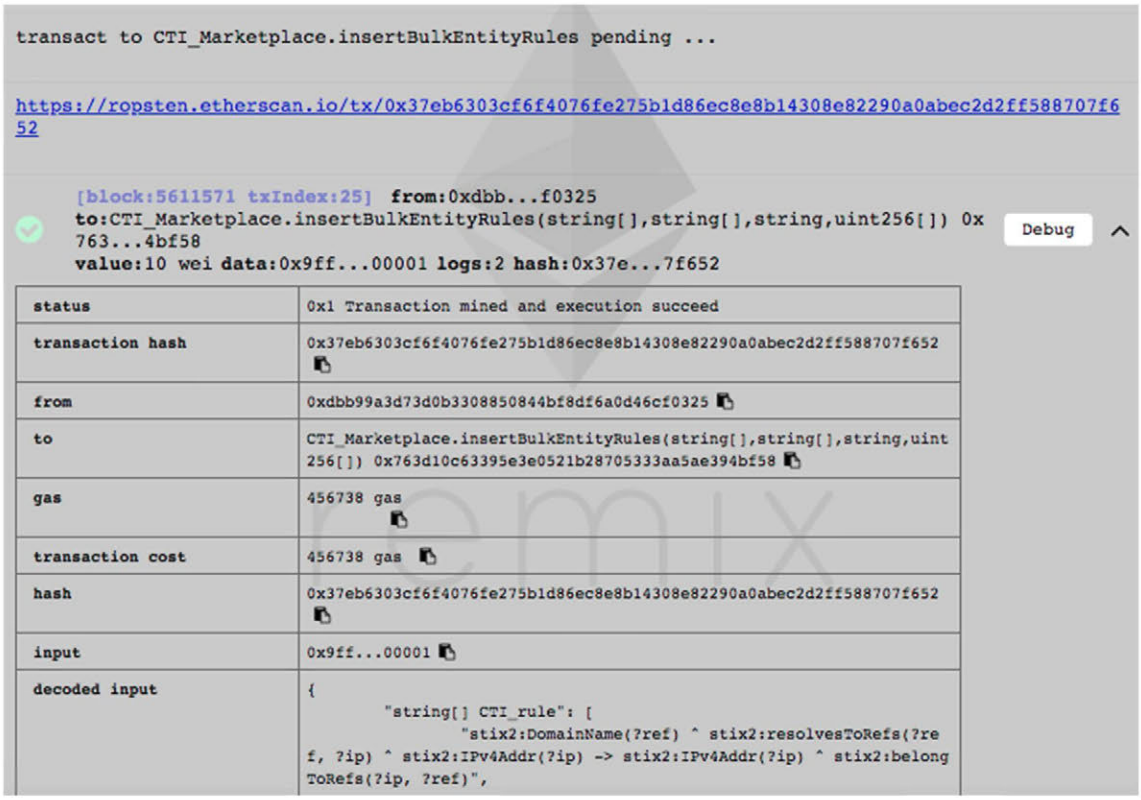


Fig. 15 Capture of Remix IDE of a transaction in Ropsten mined and executed successfully

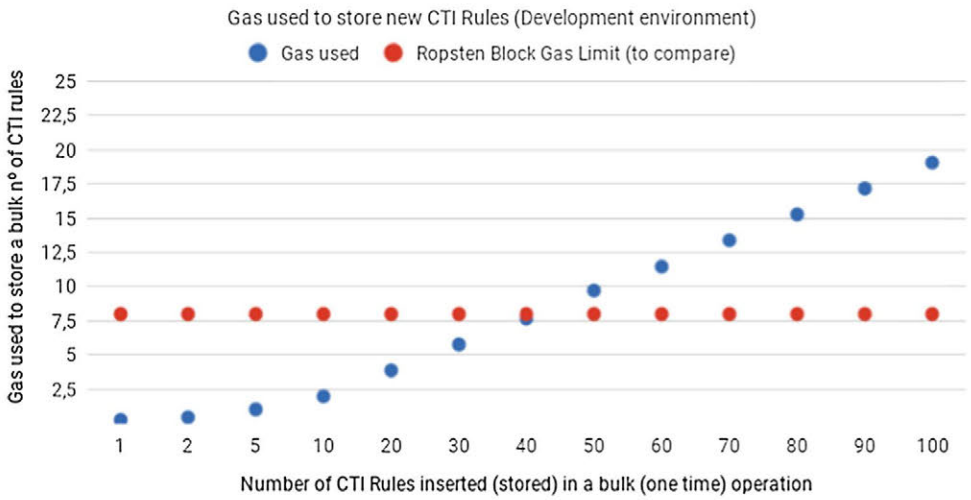
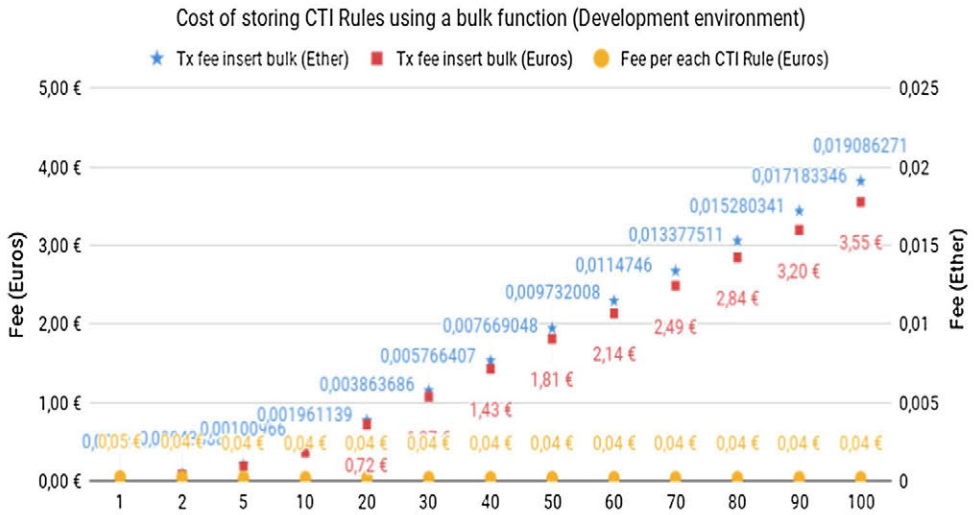


Fig. 16 Graph of gas used to write (store) CTI rules in the Marketplace in bulk operations

generate incomes above that investment. Each time that rule is read by any CTI Data consumer, it will generate to its source direct economic benefits (50% of cash on taxes in the same transaction). More than that, we think that CTI Data producers would also be willing to invest in CTI tokens (due to the opportunity to get special investment conditions while sharing new CTI Data) as its value is expected to be growing if the Marketplace usage is also growing along the time. The Gas Block Limit of Ropsten but also the main-net's suggest that the **maximum number of rules is 40 CTI rules per transaction**. That transaction will cost about 0.007669 Ether (equivalent to 1.43 euros today).

With regard to reading that data (role of CTI Data consumers), Fig. 18 shows that the situation is very different. Although it also follows a linear growing pattern, the cost to write is near 21 times the cost to read. The minimum cost to read a single CTI rule is about 0.0019 euros when the minimum cost to write a single CTI rule was about 4 euro cents (0.04 euros). Our implementation requires a tax payment to



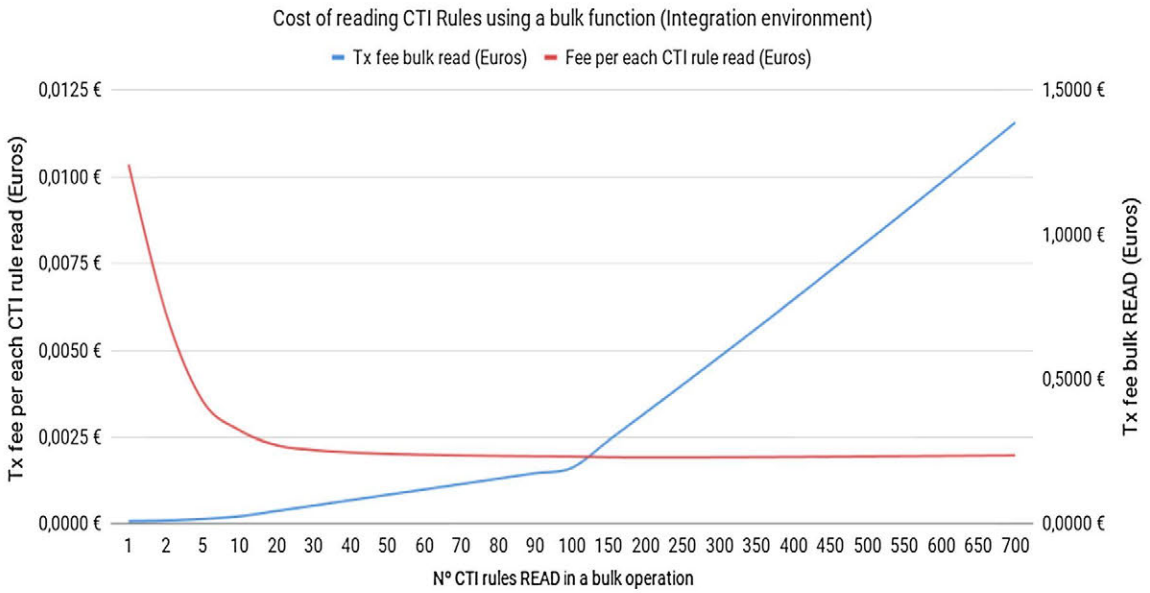


Fig. 17 Graph of gas cost to write (store) CTI rules in the Marketplace and the cost per rule in bulk operations

Fig. 18 Graph of gas cost to read (query) CTI rules in the Marketplace and the cost per rule in bulk operations

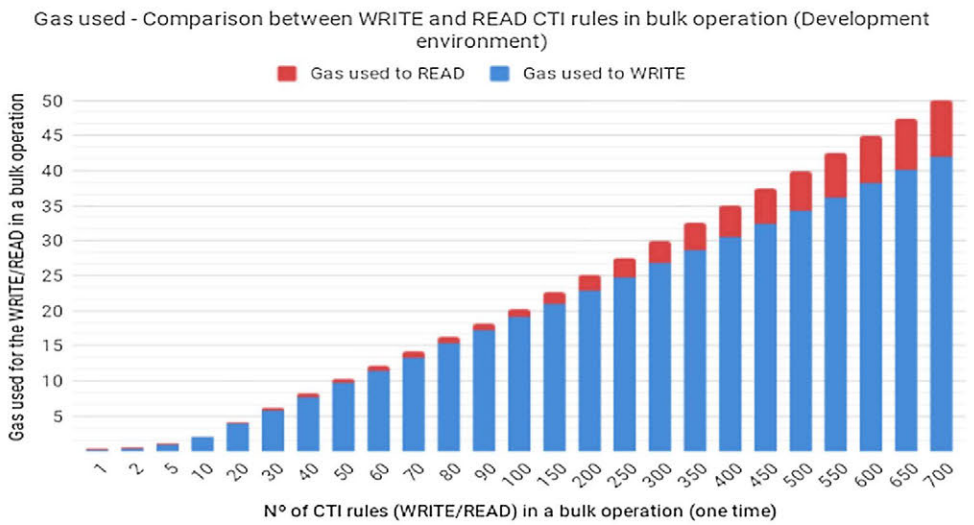


Fig. 19 Graph comparing the gas used between READ (querying) and WRITE (storing) CTI rules in bulk operations within the CTI Blockchain based Marketplace

read, so remember that it is not just a simple Ethereum "call" but a transaction due to the used modifiers.

As either the price of gas or the price of an Ether are variables, results can be higher (gas price or ETH price increase) or cheaper (gas price or ETH price decrease). At the time of this work the price of Ether was 186,08 euros and the gas cost was 1 Gwei (see Eq. 12).

In Fig. 19 we introduce a comparison between read and write operations from a single rule to a bulk insert of 700 rules in the same transaction.

6.3 Benefits of our proposal

CTI Rules (as seen in Fig. 15) are written in STIX™v2 [18], OWL [19] ontologies [20] and SWRL semantic rule language [21] as proposed by authors in [9].

Then, the exchange of this type of information (algorithms and rules), in the format described, will provide benefits like:

- *Reasoning and inference* data and algorithms will be understood by semantic reasoners (and humans) [17], to infer new knowledge.
- *Encouraging participation of more producers* it will avoid specific, ephemeral and static IoC to be shared, instead, the data will represent patterns in the form of rules (with variables).
- *Effective and efficient actionable intelligence* it will allow a more efficient and effective sharing of knowledge (e.g. just one algorithm representing a pattern, can represent multiple combinations of IoC values of different families). Better algorithms, will bring greater benefits.

At the same time, despite the limitations detected in the number of bulk CTI rules that can be stored in a single transaction (see Figs. 16 and 17) or the number of CTI rules that can be read (see Fig. 18) in the same transaction, there are great advantages and problems resolved by our approach, compared to legacy CTI exchange centralized models. Among them:

- *Identity and trusted sources* the usage of crypto-wallets addresses, where only their private key owners can update, use, send or approve transactions, is giving a level of identity and trust on the source that goes beyond nicknames, user/password or even double factor authentication approaches. In addition to this, the blockchain network functioning gives trust itself by default to untrusted sources, it combines the openness of the internet with the security of cryptography to give everyone a faster, safer way to verify key information and establish trust. As an example, transactions are verified by the network nodes to give trust of the whole transaction. Transactions are, in our example, calls or functions within a software program, a smart contract.
- *Authority* sharing rules or algorithms beyond just sharing detections (IoC). It allows a better understanding on the level of knowledge of the source. Rating provided by consumers (in the form of a blockchain smart contract transaction) will be used as a de facto way to provide feedback.
- *Time-stamping* feature can be provided by the blockchain.
- *Internal and external consistency* we use an ontology version of STIX™to foster standardization as it is widespread in the CTI industry. By using an ontology, it would be easy for a semantic reasoner to understand the data, as an example, it will know if forbidden data is used by the CTI rule (e.g. TLP - Traffic Light Protocol restrictions).
- *Accounting* what was shared, when it was shared and by whom. It is clear that blockchain technology will keep this immutable register which is very interesting for accounting and audit purposes.
- *Availability* the decentralization provided by the blockchain guarantees as much availability as possible. Due to the fact that processing and validations are giving incentives to network nodes, it is very difficult to have any network outage. In case of a centralized infrastructure or even cloud computing, there are ongoing cost and investments needed that will never be cheaper than a decentralized infrastructure like blockchain.
- *Low cost* a decentralized smart contract will have associated cost to real usage of the blockchain (gas used in mining calls, taxes applied to list CTI rules, etc.). No additional costs or investments are needed. On the other hand, the more usage of the CTI marketplace service, the more value and incentives are added to the system (CTI token value and balance). The value of the CTI token will be higher and more interesting for all stakeholders involved.
- *Attractive and Invest ready* special conditions and incentives apply to have more CTI Data producers in order to solve the asymmetric approach. As suggested by [3], some economic incentives, as part of a business model, are associated to this proposal. Our proposal enables any wallet to invest in the CTI token which is based on ERC20 standard. Being cyber threat intelligence a growing valuable asset within cybersecurity international community, its digital representation or digital asset will also follow the same path. As an on-going ICO (initial coin offering), our CTI token can be available for investment to anyone interested in it.
- *Secure* some very interesting security features are inherited from the blockchain (e.g. Merkle trees, PKI, hashes, Proof of Work, Proof of Stake, Verification, Immutable

registers, Transparent register, secure coding, modifiers, etc.).

In order to have a better understanding about how our specific contributions are addressing all of the open challenges today seen in Table 1, we created the Table 2. The last column indicates the ID reference of the Table 1.

In the Ethereum blockchain the confidentiality of data and transactions is not guaranteed. As an example, the use of variables declared as private within the smart contract are still visible by using certain techniques.

Most of the consumers, which share the same motivation for the attacker, are potential new victims. All of them might be interested to update their defenses with new knowledge to improve their detection, prevention and response capabilities.

6.4 Discussion

As a result, however there is a cost to store and read CTI Data from the blockchain, it could be cheaper than deploying and maintaining a dedicated infrastructure. It will also have better availability by definition of a decentralized application.

The cost are now related to the real usage of the CTI Data, which is very interesting, as there are not ongoing cost for maintenance because the use of a decentralized infrastructure. This setup, provides interesting and economic incentives to every role involved, either short (by cash) or long term (by CTI tokens). Consumers, which are willing to pay, will pay for less ephemeral intelligence and higher value added (knowledge in the form of TTP detection algorithms beyond oC). On the other hand, the grade of volatility and variability of either ETH or gas prices, could affect the behavior of users, fortunately an increment on the price does not mean a decrement of usage. Anyway, the CTI token could keep its value despite changes over the ETH prize.

The limitation of the storage per transaction and the needed gas for this type of (write) transactions can be improved if using a mixed approach (on and off chain), like for example using IPFS,[14] Filecoin[15] or Storj.[16]

In [28], authors propose a compression ratio of 0.0817 if using an IPFS-based blockchain data storage model to solve the problem of bitcoin ledger size. It had more than 200GB which was an issue for any node to join the network.

Intelligence of any Dynamic Risk Framework (as seen in [9]), should be updated along the time. Any detection rule but also any mitigation, tactic o strategy rule that can be modeled in the format of a SWRL rule, would be part and eligible for our CTI Data system.

[14] https://ipfs.io/.

[15] https://filecoin.io/filecoin.pdf.

[16] https://storj.io/whitepaper/.

On the other hand, our proposal provides new economic incentives as suggested by [3], that will help to reduce the asymmetry between producers and consumers today. CTI token value would be increased within the usage of the CTI Data. The value added to the Marketplace, if this type of knowledge is shared, will increase the value of the CTI token. More incentives to all will be possible only if the CTI data has very high quality. The different parameters (e.g. tax, p_c equation, etc.) were modeled to simulate the potential market growth. New simulations are recommended to better adjust the different key parameters to better address potential requirements by the investors or any other role.

With regard to security, there is still room for improvement in some specific but important topics like confidentiality or privacy. Nothing in the Ethereum blockchain is private. The keyword private, is merely an artificial construct of the Solidity language. Web3's getStorageAt(...) as seen in [29] can be used to read anything from (private) storage. It can be tricky to read what you want though, since several optimization rules and techniques, are used to compact the storage as much as possible.

7 Conclusions and future work

Any entity is exposed to several cybersecurity threats everyday. The Cybersecurity Threat Intelligence (CTI) data is considered, one of the most valuable assets of any organization, to better detect, prevent and response to cybersecurity threats on time. Its value is related to the quality, understood as the timing (when and how fast is available), the reliability and accuracy of the data. Because of that, there is a very high demand of CTI data, however, there is a limited size of providers, compared to the demand size. Entities are using different taxonomies without enough expressivity to define complex relationships, which are needed to create context-aware or behavioral rules. STIX™is a promising standard but it still lacks of semantics. Furthermore, users are reluctant to share.Trust is one of the main reasons behind, but there are much more reasons (see Table 1).

This paper presents a new model, to provide Cybersecurity Threat Intelligence Exchange, based on Blockchain. It provides new economic incentives to all roles involved, as well as an enhanced version of the peers. In order to operate, share and consume semantic advanced intelligence automatically, a semantic reasoner is considered a key and a powerful building block. It will contribute to all transactions by its reasoning capabilities, ranging from the role used in a specific transaction, to the inference of new knowledge at any level (e.g. attribution), or even a simple TLP eligibility check before a new data is shared.

A new CTI token is also provided, as a digital representation of the CTI Data asset. It will add more value (benefit

and market growth), especially for CTI Data producers, but also to Investors. We made some Montecarlo simulations as part of our proposal, to calculate and optimize certain key parameters. As a result, the system is more dynamic and it provides more value back, to more stakeholders, than current approaches (OSINT, Feeders, ISAC). We evaluated the dynamics of our model through an experimentation. It allowed us to validate the benefits but also it helped us to detect potential limitations, especially with regard to storage, querying and processing transactions costs.

As a result, our contribution is providing a coherent solution at the same time to all of the open challenges described in this work (see Tables 1 and 2).

At the same time, we identified some future research directions and limitations that were encountered during the evaluation of our implementation:

In our work, we were focused in the utility function's parameters (see Eq. 9) related to the incentives to have more CTI Data producers and the reliability of the data. We made experiments and simulations to analyze some key parameters, like the breakeven for this role (e.g. the limit to get favorable economic conditions). However, the model and all their variables, should be improved and fine tuned, running additional simulations, adding special profitability requirements of investors and its dependency with the profitability of producers and any other type of roles. The impact of malicious nodes has to be also introduced in our research.

With regard to the CTI token value (see Eq. 11), we depend on the value of Ether today, which is continuously fluctuating due to the trading of crypto currencies. We will work to give more stability alternatives to the CTI token, as well as considering the impact of the limitation in the number of tokens NT (as proof-of-stake) along the time.

There are also technical limitations in our model. The technology and standards should evolve together to work in this specific enhanced version of the peers and the network. Some research about the potential impact, dependencies and transition cost are recommended. We propose to research in potential transition and coexistence scenarios. In addition to this, most of the security limitations encountered are inherent to the blockchain technology itself, like all the confidentiality and privacy issues detected. Some workarounds can be implemented nowadays however they will provide less efficient coding as well as higher storing or processing costs. New research will be focused to keep the efficiency of the model, while providing more confidentiality and privacy by default.

Compliance with ethical standards

Conflict of interest All authors declare that they do not have any conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. OSINT, ATP 22-2.29. Open source intelligence headquarters departments of the army. Retrieved June 1, 2019 from https://fas.org/irp/doddir/army/atp2-22-9.pdf.
2. NIST Guide to CTI sharing. (2016). *Guide to cyber threat information sharing*, Special Publication 800-150. https://doi.org/10.6028/NIST.SP.800-150.
3. Vishik, C., Sheldon, F., & Ott, D. (2013). Economic incentives for cybersecurity: Using economics to design technologies ready for deployment. In H. Reimer, N. Pohlmann & W. Schneider (Eds), *ISSE 2013 securing electronic business processes* (pp. 133-147). Wiesbaden: Springer. https://doi.org/10.1007/978-3-658-03371-2_12.
4. Tosh, D., Sengupta, S., Kamhoua, C. A., & Kwiat, K. A. (2018). Establishing evolutionary game models for CYBer security information EXchange (CYBEX). *Journal of Computer and System Sciences*, *98*, 27–52. https://doi.org/10.1016/j.jcss.2016.08.005.
5. Skopik, F., Settanni, G., & Fiedler, R. (2016). A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing. *Computers and Security*, *60*, 154–176. https://doi.org/10.1016/j.cose.2016.04.003.
6. de Fuentes, J. M., González-Manzano, L., Tapiador, J., & Peris-Lopez, P. (2017). PRACIS: Privacy-preserving and aggregatable cybersecurity information sharing. *Computers and Security*, *69*, 127–141. https://doi.org/10.1016/j.cose.2016.12.011.
7. Ring, T. (2014). Threat intelligence: Why people don't share. *Computer Fraud and Security*, *2014*(3), 5–9. https://doi.org/10.1016/S1361-3723(14)70469-5.
8. Tounsi, W., & Rais, H. (2018). A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers and Security*, *72*, 212–233. https://doi.org/10.1016/j.cose.2017.09.001.
9. Riesco, R., & Villagra, V. A. (2019). Leveraging cyber threat intelligence for a dynamic risk framework. *International Journal of Information Security*. https://doi.org/10.1007/s10207-019-00433-2.
10. Sauerwein, C. et al. (2017). *Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives*. Wirtschaftsinformatik. Retrieved June 1, 2019 from https://www.wi2017.ch/images/wi2017-0188.pdf.
11. Leszczyna, R., & Wróbel, M.R. (2019). *Threat intelligence platform for the energy sector*. In Wiley online library. https://doi.org/10.1002/spe.2705.
12. Wagner, C., Dulaunoy, A., Wagener, G., & Iklody, A., MISP: The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 2016 ACM on workshop on information sharing and collaborative security (WISCS '16). ACM, New York, NY, USA* (pp. 49–56). https://doi.org/10.1145/2994539.2994542.
13. NATO OSINT Handbook. *NATO Open source intelligence handbook*. Retrieved June 1, 2019 from https://www.academia.edu/4037348/NATO_Open_Source_Intelligence_Handbook.
14. Bianco, D. (2014). *The pyramid of pain*. Retrieved June 1, 2019 from http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html.
15. OASIS. *TTP (Techniques, Tactics and Procedures by STIX™*. Retrieved June 1, 2019 from https://stixproject.github.io/getting-started/whitepaper/#tactics-techniques-and-procedures-ttp.
16. ERC20 token *IEIP20 - ERC20 standard token*. Retrieved June 1, 2019 from https://eips.ethereum.org/EIPS/eip-20.

17. W3C. *Reasoner*. Retrieved June 1, 2019 from https://www.w3.org/2001/sw/wiki/Category:Reasoner.
18. OASIS. *STIX™2.0 specifications*. Retrieved June 1, 2019 from https://oasis-open.github.io/cti-documentation/stix/intro#stix-2-objects.
19. W3C. *OWL*. Retrieved June 1, 2019 from https://www.w3.org/OWL/.
20. W3C. *Ontology*. Retrieved June 1, 2019 from https://www.w3.org/standards/semanticweb/ontology.
21. W3C. *SWRL semantic web rule language*. Retrieved June 1, 2019 from https://www.w3.org/Submission/SWRL/.
22. Nath, I. (2016). Data exchange platform to fight insurance fraud on blockchain. In *IEEE 16th international conference on data mining workshops (ICDMW)*. https://doi.org/10.1109/ICDMW.2016.0121.
23. Polyswarm. Polyswarm decentralized threat detection marketplace. Retrieved June 1, 2019 from https://polyswarm.io.
24. Graf, R., & King, R. (2018). Neural network and blockchain based technique for cyber threat intelligence and situational awareness. In *International conference on cyber conflict, CYCON*. https://doi.org/10.23919/CYCON.2018.8405028.
25. OASIS. *STIX™White paper*. Retrieved June 1, 2019 from https://stixproject.github.io/about/STIX_Whitepaper_v1.1.pdf.
26. Ravsan, S.K. (2018). Utility tokens: Discussion, economic model and simulation in R, Hackernoon. Retrieved June 1, 2019 from https://hackernoon.com/utility-tokens-discussion-economic-model-and-simulation-in-r-798c0ff3d26c.
27. Ciaian, P., Rajcaniova, M., & Kancs, A. (2016). The economics of BitCoin price formation. *Applied Economics*, *48*(19), 1799–1815. https://doi.org/10.1080/00036846.2015.1109038.
28. Zheng, Q., Li, Y., Chen, P. & Dong, X. (2018). An innovative IPFS-based storage model for blockchain. In *IEEE/WIC/ACM international conference on web intelligence, WI*. https://doi.org/10.1109/WI.2018.000-8.
29. Ethereum contract storage. *How to read Ethereum contract storage*. Retrieved June 1, 2019 from https://medium.com/aigang-network/how-to-read-ethereum-contract-storage-44252c8af925.