ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov726

Blockchain-Based Threat Intelligence Sharing Platform

Haripriya M.[1]; Yousuf Al-Mazrouai[2]; Almuntaser Alghafri[3]; Ibrahim Alharmali[4]

[1]College of Computing and Information Sciences University of Technology and Applied Sciences – Al-Mussanah Sultanate of Oman
[2]College of Computing and Information Sciences University of Technology and Applied Sciences – Al-Mussanah Sultanate of Oman
[3]College of Computing and Information Sciences University of Technology and Applied Sciences – Al-Mussanah Sultanate of Oman
[4]College of Computing and Information Sciences University of Technology and Applied Sciences – Al-Mussanah Sultanate of Oman

Publication Date: 2026/11/23

Abstract: The secure and voluntary exchange of Cyber Threat Intelligence (CTI) among organizations is a widely recognized prerequisite for effective collective cybersecurity defense. However, existing platforms based on centralized architectures suffer from critical structural weaknesses including single points of failure, lack of trust enforcement, and exposure of organizational identity. This paper proposes a Blockchain-Based Threat Intelligence Sharing Platform that replaces the central server model with a fully decentralized Ethereum blockchain architecture governed by a Vyper smart contract. The platform enables organizations to submit Indicators of Compromise (IoCs)—such as malicious IP addresses, file hashes, and phishing URLs—which are validated and stored permanently on-chain through a reputation-based mechanism. Organizations earn on-chain reputation scores by submitting and validating threat data, creating an incentive structure that encourages participation without relying on monetary rewards. A web-based user interface built with HTML, CSS, and JavaScript allows organizations to register, submit threats, view the live shared threat feed, and validate submissions. A Node.js or Python backend API bridges the frontend and the blockchain, with user accounts managed in a MongoDB or MySQL database. The system is deployed on the Ethereum Sepolia public test network, making it the first open prototype in this domain to cover all architectural layers—smart contract, API, database, and user interface—identified as missing by the existing literature.

Keywords: *Blockchain, Cyber Threat Intelligence, Smart Contract, Vyper, Ethereum, Indicators of Compromise, Decentralized Platform, Reputation System.*

How to Cite: Haripriya M.; Yousuf Al-Mazrouai; Almuntaser Alghafri; Ibrahim Alharmali (2026) Blockchain-Based Threat Intelligence Sharing Platform. *International Journal of Innovative Science and Research Technology*, 11(11), 1131-1137. https://doi.org/10.38124/ijisrt/26nov726

I. INTRODUCTION

➢ *Background*

Cyberattacks are growing faster and becoming harder to stop. When one company is attacked, others often do not hear about it until it is too late. Cyber Threat Intelligence (CTI) is information about these attacks—for example, which IP addresses hackers use, what malicious files look like, and which websites trick people into giving away their passwords. When organizations share this kind of information with each other quickly, they can all protect themselves before the same attack reaches them [1]. However, most current sharing platforms are built around a central server, which creates a single point of failure—if that server is hacked or goes down, the entire sharing network stops working [2].

Platforms like MISP, STIX, and TAXII help organizations organize threat data, but they still depend on a trusted central administrator. This means one organization has full control over the data, which makes others hesitant to share sensitive information [3]. There is also no automatic way to check if the information shared is real or fake, allowing bad actors to inject false threat data into the system.

➢ *Motivation and Problem Statement*

The main motivation behind this project is the observation that organizations do not trust each other enough to share threat data freely. Technical standards like STIX/TAXII organize the format of the data, but they do not verify whether the person sharing the data is trustworthy [4]. A central platform administrator could also be bribed, hacked, or make mistakes that corrupt the whole shared database. This

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov726

project replaces the central administrator with a blockchain, where rules are enforced automatically by smart contract code that no one can change after it is deployed.

➢ *Problem Definition*

This project addresses three specific problems. First, existing CTI platforms break if the central server is attacked. Second, there is no automated way to check whether submitted threat data is genuine before it reaches other users. Third, organizations must reveal their identity when using most platforms, which discourages participation in competitive industries. This project solves all three problems at once by using a decentralized blockchain, automated smart contract validation, and anonymous wallet-based identity.

➢ *Research Gap*

Many researchers have proposed blockchain-based CTI systems, including frameworks by Gong and Lee [5], Badsha et al. [6], and Nguyen et al. [7]. However, a comprehensive review by Chatziamanetoglou and Rantos [8] confirmed that none of these works produced a fully working system that includes a website, user login, database, and live public blockchain deployment together. This project fills that gap.

➢ *Objectives*

This project has three main objectives. The first is Decentralized Ledger Development—building a peer-to-peer Ethereum network so threat data is stored across many nodes with no single point of failure. The second is Smart Contract Verification—using a Vyper smart contract to automatically check the quality and credibility of submitted threat data before it is saved permanently. The third is Secure Data Access—allowing organizations to share threat details such as malicious IP addresses using anonymous wallet identities, without revealing who they are.

➢ *Organization of the Paper*

Section II reviews related research and explains how this project builds upon existing work. Section III describes the proposed system design. Section IV presents the test results and discussion. Section V concludes the paper.

II. LITERATURE REVIEW

Sharing CTI between organizations has long been recognized as a key part of protecting against cyberattacks. Traditional platforms based on the STIX/TAXII standard have well-known weaknesses: they rely on a central server that can be hacked, and they allow one entity to control all the shared data [4]. Johnson and White [3] showed that these platforms also put organizations at risk because the platform owner can see which threats each member is facing, making organizations reluctant to share. This project directly addresses these weaknesses by removing the central server and replacing it with a blockchain that no single party controls.

Blockchain technology is widely proposed as a solution to the trust problem in CTI sharing. Its key properties—permanent records, no central owner, and public verification—fix the exact problems of centralized platforms [9]. Pahlevan and Ionita [4] demonstrated this by building a working private blockchain testbed where organizations exchanged TAXII-formatted threat data without trusting a central intermediary. Moubarak et al. [10] tested Hyperledger's performance for CTI sharing and found it fast enough for practical use. This project takes the same idea further by using Ethereum and the Vyper smart contract language, which is simpler and safer to audit than Solidity [11].

A major challenge in open CTI sharing is making sure the submitted data is real and trustworthy. Gong and Lee [5] tackled this with BLOCIS, a blockchain framework that gives each contributor a reputation score stored permanently on the blockchain, making it harder for attackers to flood the system with fake data. Chatziamanetoglou and Rantos [12] went further with a Proof-of-Quality consensus algorithm where validators are chosen based on their past reputation scores, ensuring that only trusted members confirm new threat data. Their later study [13] proved through simulation that this approach stays reliable even when 40% of validators are acting maliciously. This project's validateThreat() function is directly inspired by this three-validation threshold model.

Keeping organizations' identities private while still sharing useful threat data is one of the hardest design challenges. Badsha et al. [6] addressed this with BloCyNfo-Share, which uses advanced encryption techniques to allow partners to share threat indicators without revealing sensitive operational details. Liang et al. [14] showed that simply using anonymous wallet addresses—like Ethereum does—is enough to protect identity in most cases without needing complex encryption. This project adopts the wallet address approach for simplicity while still achieving the privacy goal.

Getting organizations to actually share threat data requires more than just technology—they need a reason to participate. Nguyen et al. [7] designed a framework that gives reputation and financial rewards to organizations that share verified threat data in industrial systems. Purohit et al. [15], [19] built DefenseChain, a consortium blockchain system where contribution quality determines an organization's reputation score and threat information is spread to all members within five seconds. Preuveneers et al. [16] showed that transparent, community-managed reputation scores naturally reduce free-riding—where organizations take information but do not contribute. This project applies these lessons by giving +1 reputation for submitting a threat and +2 for each successful validation received.

Storing large amounts of threat data directly on a blockchain is expensive. Neto et al. [17] and Menges et al. [18] both recommended storing only short identifiers and hashes on-chain and keeping full reports in a separate storage system. This project balances this by storing short IoC values—IP addresses, file hashes, and URLs—directly on-chain where cost is acceptable, while keeping user account data in a separate database for faster loading. Recent frameworks such as TrustShare by Ali et al. [1] and Menges et al. [18] also emphasized that aligning blockchain CTI systems with the STIX 2.1 standard improves compatibility with existing security tools, which is why this project's threat types are designed to be mappable to STIX 2.1 objects in the future.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov726

➢ *Comparative Summary of Related Literature*

A synthesis of the reviewed literature reveals four distinct architectural and operational patterns across existing blockchain-based CTI studies, alongside recurring limitations that motivate our proposed system:

- Permissioned Consortium Frameworks: Studies utilizing Hyperledger Fabric and private blockchains [1], [4], [10] achieve high transaction throughput and channel-level access control for STIX/TAXII dissemination. However, their reliance on closed permissioned consortia restricts open cross-sector participation, introduces complex certificate management overhead, and omits public user-facing web interfaces.
- Reputation and Quality Consensus Models: Frameworks such as BLOCIS [5], Proof-of-Quality (PoQ) [12], [13], and distributed reputation ledgers [16] effectively mitigate Sybil attacks and free-riding through weighted validator selection and historical quality tracking. Despite strong theoretical resilience (sustaining up to 40% malicious validators), these contributions remain confined to mathematical models or simulation environments without live blockchain deployment.
- Cryptographic Access Control and Token Incentives: Solutions incorporating Attribute-Based Encryption (ABE), proxy re-encryption, and IPFS off-chain storage [6], [7], [15], [19] preserve confidentiality and incentivize industrial sharing, but impose heavy cryptographic latency, complex token-economy administration, or cloud vendor lock-in.
- Positioning of the Proposed System: As confirmed by the systematic review of over 50 papers by Chatziamanetoglou and Rantos [8], no prior work delivers an open, end-to-end working prototype that unifies all four architectural layers. Our proposed platform bridges this gap by combining an auditable Vyper 0.3.10 smart contract deployed on the public Ethereum Sepolia test network, a token-free reputation mechanism (+1 per submission, +2 per confirmed validation), a RESTful backend API with off-chain database caching, and a responsive Bootstrap 5 web interface integrated with MetaMask.

III. PROPOSED WORK

The proposed system is a full-stack, decentralized CTI sharing platform with three working layers: a Vyper smart contract on the Ethereum blockchain, a backend server API, and a plain HTML website. Each layer has a clear role and they work together to meet all three project objectives.

The choice of Vyper over Solidity is based on its security-focused design philosophy. Vyper has a simpler syntax, fewer features, and avoids complex constructs such as inheritance and modifiers, which reduces the likelihood of vulnerabilities. This makes the smart contract easier to audit and verify. Additionally, Vyper enforces explicit coding practices and minimizes hidden behaviors, which improves code readability and predictability. These characteristics are critical in security-sensitive applications such as CTI sharing, where trust and correctness are essential.

➢ *Smart Contract Layer*

The Smart Contract Layer is the core of the system. A Vyper 0.3.10 smart contract named ThreatIntelligence.vy is deployed on the Ethereum Sepolia public test network. The contract stores four types of on-chain data: threat records (linked by a unique ID number), reputation scores for each organization's wallet address, a registration list showing which wallets have joined the platform, and a double-validation map that stops any organization from confirming the same threat twice. The contract has six functions. registerOrganization() gives a new wallet an initial reputation score of 1 and marks it as registered. submitThreat() accepts a threat type, the actual indicator value, a severity level from 1 to 5, and a short description—then saves the record permanently on-chain and gives the submitter +1 reputation. validateThreat() lets registered organizations confirm that another organization's threat is real—and when a threat receives three independent confirmations, it is automatically marked as Verified and the original submitter receives +2 reputation. The three read-only query functions (getThreat(), getThreatCount(), and getReputation()) return stored data without any transaction cost.

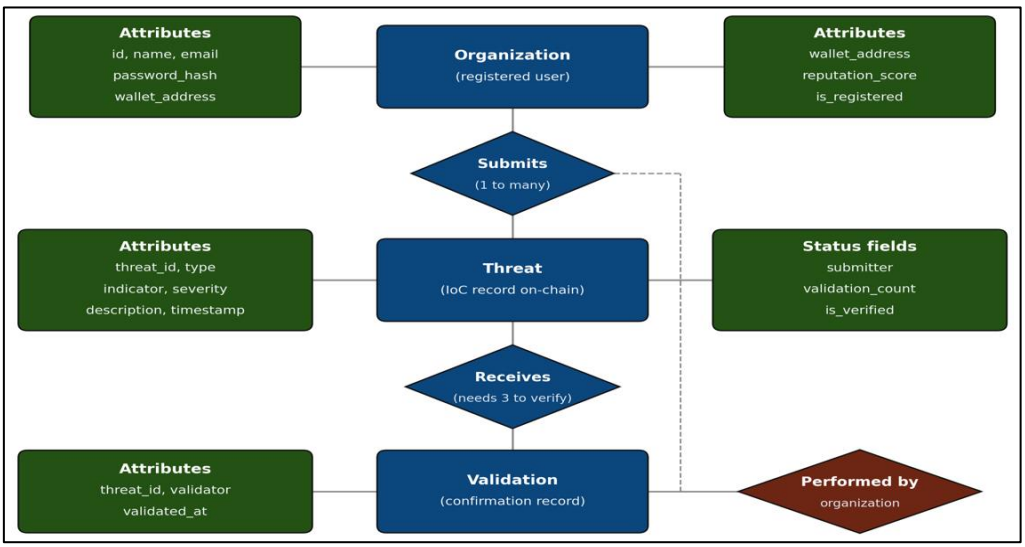


Fig 1 Entity-Relationship Diagram (ERD) of the CTI Platform.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov726

➢ *Backend API Layer*

The Backend API Layer connects the website to the blockchain. It is built using either Node.js with Express.js or Python with Flask, giving the team flexibility in language choice. The backend handles eight REST API routes: user registration, login, on-chain wallet registration, threat submission, threat retrieval (all or one), threat validation, and reputation queries. Passwords are hashed using bcrypt before storage, and every protected route requires a valid JSON Web Token (JWT) obtained at login. The Ethers.js library (Node.js) or web3.py (Python) connects the backend to the smart contract using the contract's ABI and Sepolia deployment address. A MongoDB or MySQL database stores user account information and caches threat records for fast page loading.

➢ *Frontend Layer*

The Frontend Layer is a simple website built with plain HTML, CSS, and Vanilla JavaScript, styled using Bootstrap 5 loaded from a CDN link—no installation or build step needed. MetaMask, a browser extension Ethereum wallet, connects users to the blockchain using the window.ethereum API built into modern browsers. The site has six pages: a login page, a registration page that auto-fills the user's wallet address from MetaMask, a dashboard showing reputation scores and platform statistics, a threat submission form with a dropdown for threat type and a slider for severity, a live threat feed table with search and filter tools, and a validate button on each feed row that calls the smart contract directly. The frontend is hosted free of charge on GitHub Pages or Netlify. Fig. 2 illustrates the main dashboard and live threat intelligence feed interface, while Fig. 3 shows the threat indicator submission page interacting directly with the MetaMask wallet extension on the Sepolia test network.



Fig 2 MetaMask Wallet Accounts Representing Four Simulated Organizations (Org A–D) in the Local Blockchain Prototype.

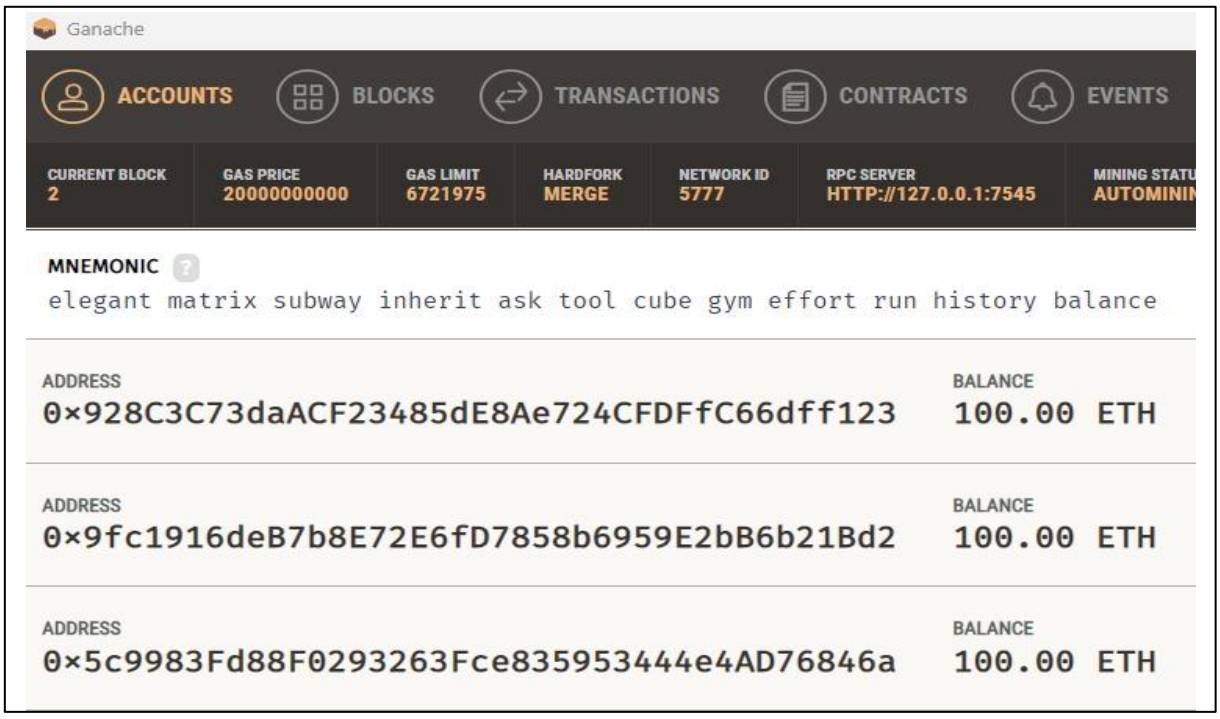


Fig 3 Ganache Local Ethereum Environment Displaying Test Accounts with Balances of 100 Test ETH Each and the RPC Endpoint http://127.0.0.1:7545.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov726

➢ *Data Model*

Fig. 1 shows the main data stored in the Blockchain CTI Platform and how it connects. There are three main entities: Organization (the registered users with their wallet and reputation), Threat (the shared cyber threat data such as malicious IPs or file hashes), and Validation (the record of one organization confirming another's threat). An organization submits many threats, each threat receives validations from other organizations, and every validation is performed by an organization. Once a threat receives three validations, it is automatically verified.

IV. EXPECTED RESULTS AND DISCUSSION

The platform will be tested in three stages: functional testing, security testing, and quantitative performance evaluation. The smart contract will first be tested on a local Ganache blockchain during development and then deployed to the Ethereum Sepolia public test network for the final evaluation.

➢ *Functional Testing—End-to-End*

A ten-step user journey will be tested from start to finish. Every step is expected to be completed successfully, showing that all three layers—smart contract, backend, and website—work together correctly. Table 1 shows each step and its expected result across the user interface (Fig. 2 and Fig. 3), REST API, and Sepolia blockchain.

Table 1 Functional Test Expected Results

Step	Action	Expected Result
1	Connect MetaMask wallet to the website	MetaMask popup appeared and connection was approved
2	Register as a new organization (Org A)	Account saved; wallet registered on-chain; reputation set to 1
3	Login with email and password	JWT token returned; redirected to dashboard page
4	Dashboard shows reputation = 1	Correct score displayed immediately after login
5	Submit a threat (Type: IP Address, Value: 1.2.3.4, Severity: 3)	Threat saved on-chain; transaction appeared in Ganache; transaction hash shown on page
6	Threat feed displays the submitted threat	Table row appeared with correct type, indicator, severity, and status = Pending
7	Switch to Org B wallet and register	Org B registered successfully with reputation = 1
8	Org B validates Org A's threat	Validation count increased to 1; Org A's reputation increased to 4
9	Sepolia Etherscan shows all transactions	Contract address and calls visible publicly on sepolia.etherscan.io
10	Threat marked Verified after 3 validations	isVerified flag = True; badge changed from Pending to Verified in feed

➢ *Security Testing—Boundary Enforcement*

Table 2 shows the expected results of the security boundary tests, verifying that unauthorized API calls, unregistered wallets, self-validations, and duplicate confirmations are deterministically rejected.

Table 2 Security Test Expected Results

Test	Action Attempted	Expected Result
1	API call without login token	Backend returns HTTP 401 Unauthorized
2	Unregistered wallet tries to submit a threat	Smart contract rejects the transaction call
3	Org A tries to validate its own threat	Smart contract rejects self-validation (revert)
4	Org B tries to validate the same threat twice	Smart contract rejects duplicate validation (revert)

➢ *Quantitative Performance Evaluation*

To evaluate the system performance, several quantitative metrics will be observed during testing on the Sepolia test network. The average transaction confirmation time is expected to range between 10 to 15 seconds per operation, based on typical Sepolia network behaviour documented in related work.

The system is expected to complete all functional test scenarios successfully, including threat submission, validation, and reputation updates. Each submitted threat will require exactly three independent validations to be marked as verified, ensuring data reliability.

Additionally, the reputation system is designed to behave consistently: submitting a threat will increase reputation by +1 and each successful validation received will result in a +2 increase. The platform is expected to handle multiple user interactions sequentially without errors.

These expected results indicate that the proposed system is designed to be stable, reliable, and suitable for practical CTI sharing, although further optimization may be required for large-scale deployment.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov726

V. CONCLUSION

This paper presented a Blockchain-Based Threat Intelligence Sharing Platform that solves three key problems in CTI sharing: fragile central servers, no automatic data quality checking, and identity exposure that prevents organizations from contributing. The platform replaces the central server with a decentralized Ethereum blockchain, uses a Vyper smart contract to automatically validate threat data through a reputation system, and lets organizations share safely using anonymous wallet addresses.

Based on the proposed design, all system functions are expected to work correctly, all four security rules will be enforced by the smart contract without any bypass, and transaction times are anticipated to be within acceptable ranges for a non-real-time sharing platform. The result will be the first student-level open prototype that covers all four architectural layers that previous research was missing: smart contract, REST API, web interface, and live public blockchain deployment. The platform supports Oman Vision 2040's Digital Society objective by providing a foundation for Omani government and financial organizations to share cyber defenses without depending on foreign closed-source platforms.

Future improvements include integrating Layer-2 scaling solutions such as Optimism or Arbitrum to reduce transaction costs and improve performance. Additionally, the system can be extended to support full compatibility with CTI standards such as STIX 2.1 and TAXII for better interoperability with existing cybersecurity platforms. Another important enhancement is the integration of zero-knowledge proofs (ZKP), which would allow organizations to verify the authenticity of threat intelligence without revealing sensitive details. This would significantly enhance privacy while maintaining trust.

Furthermore, future work may include implementing automated threat scoring using machine learning techniques and expanding the system to support cross-chain interoperability, enabling communication between different blockchain networks.

REFERENCES

[1]. H. Ali, W. J. Buchanan, J. Ahmad, M. Abubakar, M. S. Khan, and I. Wadhaj, “TrustShare: Secure and trusted blockchain framework for threat intelligence sharing,” Future Internet, vol. 17, no. 7, Art. no. 289, 2025, doi: 10.3390/fi17070289.

[2]. Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A methodical examination of machine learning and deep learning methodologies,” Trans. Emerg. Telecommun. Technol., vol. 32, no. 1, Art. no. e4150, 2021, doi: 10.1002/ett.4150.

[3]. R. Johnson and T. White, “Privacy risks in centralized threat intelligence repositories: An empirical investigation,” Cybersecurity J., vol. 6, no. 1, Art. no. tyaa014, 2020.

[4]. M. Pahlevan and V. Ionita, “Secure and efficient exchange of threat information using blockchain technology,” Information, vol. 13, no. 10, Art. no. 463, 2022, doi: 10.3390/info13100463.

[5]. S. Gong and C. Lee, “BLOCIS: A framework for sharing cyber threat intelligence based on blockchain that is resistant to Sybil attacks,” Electronics, vol. 9, no. 3, Art. no. 521, 2020, doi: 10.3390/electronics9030521.

[6]. S. Badsha, I. Vakilinia, and S. Sengupta, “BloCyNfo-Share: A way to share cybersecurity information based on blockchain technology with very specific access control,” in Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC), 2020, pp. 317–323, doi: 10.1109/CCWC47524.2020.9031164.

[7]. K. Nguyen, S. Pal, Z. Jadidi, A. Dorri, and R. Jurdak, “An incentivized framework for sharing cyber threat intelligence in ICS that uses blockchain technology,” in Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops), 2022, pp. 261–266, doi: 10.1109/PerComWorkshops53856.2022.9767226.

[8]. D. Chatziamanetoglou and K. Rantos, “Cyber threat intelligence on blockchain: A methodical review of the literature,” Computers, vol. 13, no. 3, Art. no. 60, 2024, doi: 10.3390/computers13030060.

[9]. X. Ma et al., “A blockchain-based incentive mechanism for sharing cyber threat intelligence,” Electronics, vol. 12, no. 11, Art. no. 2454, 2023, doi: 10.3390/electronics12112454.

[10]. J. Moubarak, C. Bassil, and J. Antoun, “On the distribution of cyber threat intelligence via Hyperledger,” in Proc. 17th Int. Conf. Design Reliable Commun. Netw. (DRCN), 2021, pp. 1–6, doi: 10.1109/DRCN51428.2021.9435486.

[11]. Vyper Documentation, “Vyper programming language reference (Version 0.3.10),” 2023. [Online]. Available: https://vyper.readthedocs.io

[12]. D. Chatziamanetoglou and K. Rantos, “CTI blockchain-based sharing utilizing the Proof-of-Quality consensus algorithm,” in Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR), 2021, pp. 331–336, doi: 10.1109/CSR51186.2021.9527959.

[13]. D. Chatziamanetoglou and K. Rantos, “Sharing cyber threat intelligence based on blockchain using Proof-of-Quality consensus,” Secur. Commun. Netw., vol. 2023, Art. no. 3303122, 2023, doi: 10.1155/2023/3303122.

[14]. X. Liang, S. Shetty, and D. Tosh, “Privacy-preserving and accountable multi-authority attribute-based data access control for the smart grid utilizing blockchain technology,” IEEE Trans. Dependable Secure Comput., vol. 19, no. 4, pp. 2236–2248, 2022, doi: 10.1109/TDSC.2021.3054766.

[15]. S. Purohit, P. Calyam, S. Wang, R. Yempalla, and J. Varghese, “DefenseChain: Consortium blockchain for cyber threat intelligence sharing and defense,” in Proc. 2nd IEEE Conf. Blockchain Res. Appl. Innov. Netw. Services (BRAINS), 2020, pp. 112–119, doi: 10.1109/BRAINS49436.2020.9223313.

[16]. D. Preuveneers, W. Joosen, J. Bernal Bernabe, and A. Skarmeta, “A distributed security framework for the safe sharing of threat intelligence,” Secur. Commun.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov726

Netw., vol. 2020, Art. no. 8833765, 2020, doi: 10.1155/2020/8833765.

[17]. E. C. P. Neto, S. Dadkhah, R. Ferreira, M. Dorodchi, and A. A. Ghorbani, "Toward a blockchain-based cybersecurity stance and threat intelligence sharing," in Proc. IEEE Conf. Commun. Netw. Secur. (CNS), 2020, pp. 1–2, doi: 10.1109/CNS48642.2020.9162266.

[18]. F. Menges, C. Sperl, and G. Pernul, "Bringing together cyber threat intelligence," Comput. Secur., vol. 106, Art. no. 102304, 2021, doi: 10.1016/j.cose.2021.102304.

[19]. S. Purohit et al., "Cyber threat intelligence sharing for co-operative defense in multi-domain entities," IEEE Trans. Dependable Secure Comput., vol. 20, no. 5, pp. 4273–4290, 2023, doi: 10.1109/TDSC.2022.3210002.