CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

| | |
|---|---|
| Title | Obfuscating network structure from blockchain analysis |
| Authors | Khalid, Asfa;Murphy, Seán Óg;Sreenan, Cormac J.;Roedig, Utz |
| Publication date | 2025 |
| Original Citation | Khalid, A., Murphy, S. Ó., Sreenan, C. J., Roedig, U. (2025) 'Obfuscating network structure from blockchain analysis', in Barolli, L. (ed.) Advanced Information Networking and Applications. AINA 2025. Lecture Notes on Data Engineering and Communications Technologies, 251. pp. 95-104. Springer, Cham. https://doi.org/10.1007/978-3-031-87781-0_11 |
| Type of publication | Conference item;book-chapter |
| Link to publisher's version | 10.1007/978-3-031-87781-0_11 |
| Rights | © 2025, the Authors. The Version of Record is under exclusive license to Springer Nature Switzerland AG. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. - https://creativecommons.org/licenses/by/4.0/ |
| Download date | 2026-11-17 11:28:00 |
| Item downloaded from | https://hdl.handle.net/10468/18334 |

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

# Obfuscating Network Structure from Blockchain Analysis

Asfa Khalid, Seán Óg Murphy, Cormac J. Sreenan, and Utz Roedig

School of Computer Science and Information Technology (CSIT),
University College Cork, Cork, Ireland
123111938@umail.ucc.ie, {seanogmurphy, cormac.sreenan, u.roedig}@ucc.ie
https://ucc.ie/nascresearch/

**Abstract.** In the context of large-scale data collection like in the Internet of Things, Data Confidence Fabrics are expected to play an essential role in verifying and authenticating sensor data. To this end, metadata is generated at each network node and stored on the blockchain. However, storing metadata on the blockchain introduces significant privacy risks, as it can be exploited to reveal sensitive information, such as network structures and communication paths. This paper addresses these challenges by proposing two novel schemes to protect network structures: *Hostname Mapping* and *Hostname Encryption*. Our work demonstrates that the Hostname Mapping approach effectively conceals network patterns but introduces inefficiencies due to additional table storage and computational overhead. In contrast, the Hostname Encryption method eliminates the need for additional table management, offering a more efficient and secure alternative. Despite these advancements, the timestamp field in metadata could still allow attackers to infer patterns using machine learning, highlighting the need for further research to fully secure metadata. By combining encryption with some enhancements to timestamp obfuscation, our approach lays the foundation for additional privacy protection against metadata exploitation.

**Keywords:** Network Security, Data Confidence Fabrics, Blockchain forensics, Metadata Privacy, Obfuscation techniques

## 1 Introduction

Data Confidence Fabrics (DCFs) have emerged as a promising solution for facilitating data authentication while ensuring the traceability of its journey across networks, prominent examples being the Linux Foundation's Alvarium [8,12] and IOTA [6]. DCFs operate by generating metadata (annotations) at each node the data passes through, meticulously documenting its journey [7]. This metadata includes information such as host names and timestamps, the origin and destination of data packets, and processing details. To ensure authentication and verification, this metadata is stored securely on a Distributed Ledger Technology (DLT) such as blockchain [6] [3]. This approach allows organizations to verify

the authenticity and origin of data, even in complex distributed systems. DCFs are especially attractive for the large-scale distributed Internet of Things, where sensor data is generated with heterogeneous provenance.

However, while DCFs excel in facilitating data verification and providing a trusted framework, they introduce critical challenges, particularly concerning the privacy of metadata. Since metadata stored on a blockchain is accessible to multiple parties, including third parties within a distributed system, it inherently exposes sensitive details about the network. This includes the structure of the network, its critical nodes, and even communication patterns. Malicious actors could exploit this information to identify key nodes, predict network behaviour, or even reconstruct the network's architecture. Such insights could be leveraged for targeted attacks or system manipulation, severely compromising the confidentiality and security of the organization's operations. While the data itself remains secure in private databases, the associated metadata's exposure poses a significant risk to the overall integrity of the system.

This paper introduces a novel approach to mitigate these risks by securing the metadata in a manner that preserves its utility for authentication, while obscuring sensitive network information from unauthorized access. Specifically, we present two innovative methods: *Hostname Mapping* and *Hostname Encryption*. These approaches are designed to enhance the privacy of metadata without compromising the functional benefits of DCFs. The Hostname Mapping approach involves generating unique, secret names for network nodes, stored securely in a lookup table inaccessible to unauthorized parties. On the other hand, the Hostname Encryption method uses the data itself as a password for encrypting host names, ensuring that only authorized entities can decode the metadata and reconstruct the network structure. By integrating these methods, this research enhances the security framework of DCFs, maintaining a balance between privacy and operational efficiency. These advancements pave the way for widespread and secure adoption of DCFs, strengthening metadata privacy in heterogeneous distributed systems.

## 2 Motivation

In a DCF, data generated by a sensor device is accompanied by metadata (annotations) that records essential details of its journey. Metadata typically includes fields such as id, key, host, kind, signature, isSatisfied and timestamp, as shown in Table 1, and is published to the blockchain for data authentication and verification. As data passes through nodes like gateways, edge devices, intermediary points, and the cloud, each node generates and publishes similar metadata to the blockchain. This process ensures traceability and authenticity, while the actual data is securely stored in the organization's private database. However, while blockchain's openness facilitates authentication, it also introduces challenges, particularly regarding metadata privacy and network security.

Anyone can easily retrieve and analyse metadata which poses risks, as malicious actors with computational tools can exploit metadata annotations to extract

**Table 1.** Description of Annotation Fields

| Field | Description |
|---|---|
| `id` | A unique identifier for the metadata, generated using a Universal Lexicographical ID generator (ULID). |
| `key` | The result of applying a hash algorithm to the data. |
| `hash` | The name of the hash algorithm used to create the key. |
| `host` | The identifier of the device generating the annotation. |
| `kind` | The security feature being annotated. |
| `signature` | A cryptographic signature applied to the metadata using the device's private key. |
| `isSatisfied` | Indicates whether the criteria for creating the annotation were met. |
| `timestamp` | The time when the metadata was generated. |

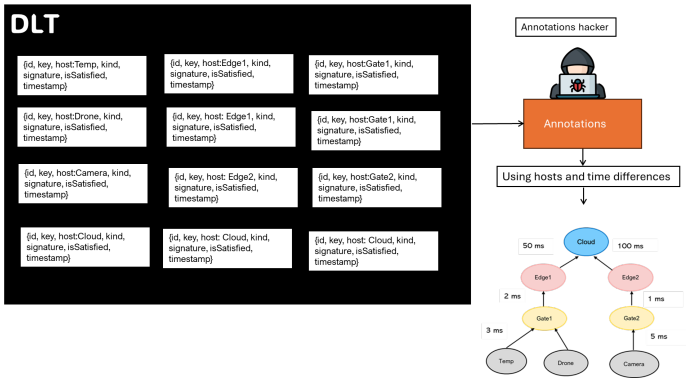



**Fig. 1.** Predicting the structure of the network using metadata.

valuable insights into the network. For example, a hacker could analyse these annotations, such as timestamps and host details, to identify strong and weak nodes in the network like hubs with high traffic, or less critical nodes. They could also uncover frequently used routes, backup paths, and node usage trends. Moreover, they can reveal peak utilization periods and even identify nodes that have failed security tests like Transport Layer Security (TLS), Transport Platform Module (TPM), or Public Key Infrastructure (PKI) checks, indicating potential vulnerabilities. Such insights into the network can provide attackers with the information needed to target critical nodes with precision. For instance, identifying nodes that handle substantial traffic or operate as backups enables attackers to disrupt data flow by launching denial-of-service (DoS) attacks or exploiting weak security points. Furthermore, timestamps combined with host names can be used to reconstruct the entire network structure. By calculating time differences between data transmissions across nodes, a hacker could map out the sequence and relationships between nodes. This constructed network map would expose the architecture and enable attackers to compromise the system strategically as shown in Figure 1.

Such metadata exploitation highlights a strong motivation for developing methods to safeguard the network's structure. These methods must not only obfuscate metadata to protect against exploitation but also ensure its usability for essential functions such as data verification and traceability. Our proposed novel schemes address these dual requirements, ensuring both security and functionality in metadata handling.

## 3 Proposed secret naming scheme for network structure obfuscation

Our proposed schemes obscure network structures by anonymizing host identifiers using *Hostname Mapping* and *Hostname Encryption*, balancing security with metadata usability and allowing authorized authorities to reverse obfuscation for auditing or operations.

### 3.1 Hostname Mapping

The Hostname Mapping approach is proposed for protecting metadata privacy in distributed systems by anonymizing host names. The approach involves creating a secure mapping table that replaces actual host names with unique secret names. This mapping table is stored in a secure, private database accessible only to authorized entities. Each time a network node generates metadata, the real host name is substituted with its corresponding secret name from the table before the metadata is published to the blockchain. For instance, consider a network node labelled "Cloud". The mapping table keep multiple secret names for the Cloud node. When data passes through the Cloud node, the metadata will replace "Cloud" with either "rwv98l" or "wu5v08", as depicted in Figure 2. The secret name used is chosen according to specific rules, ensuring variability. As a result, even if attackers retrieve the metadata from the DLT, they cannot discern the identity of the "Cloud" node. By obfuscating real host names, the method ensures that critical details of the network, such as its structure: node relationships, and usage trends, remain hidden. This obfuscation protects the network from exploitation, especially in scenarios where sensitive data passes through critical nodes.

The Hostname Mapping is an effective approach for safeguarding metadata in distributed systems by replacing real host names with secret identifiers, it obscures critical network details, reducing the risk of attacks and exploitation. However, the method is not without challenges, including table size management, the risk of collisions and security maintenance. In large-scale systems with thousands of nodes, maintaining a mapping table with unique entries for each host name requires considerable storage resources. The administrative overhead of updating, managing, and securing this table also grows proportionally. Failure to keep the table up to date can lead to inefficiencies, such as incorrect mappings or mismatches in metadata. In addition, there is also a risk of collisions or repetitions of secret names. Another critical challenge is the secure maintenance

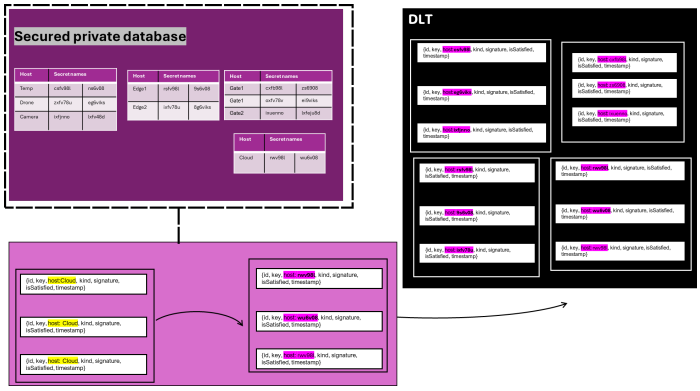



**Fig. 2.** Hostname Mapping approach

of the secret mapping table. Since the table contains the actual host-to-secret mappings, any unauthorized access to it would compromise the entire network's privacy. A compromised table would allow attackers to reverse the obfuscation process, revealing the true identities of the hosts. To address these challenges, we propose a more efficient and secure method for encrypting host names by utilizing data as an encryption key.

### 3.2 Hostname Encryption

Recognizing the scalability limitations inherent in the Hostname Mapping approach for obfuscating host names, this research proposes an encryption-based method. In this approach, host names are encrypted using keys derived directly from the associated data. By linking the encryption process to the data itself, this method eliminates the need for a separate mapping table, thereby improving both scalability and security. When a piece of data passes through a network node, a secure hashing algorithm is applied to generate an encryption key. This key (as password) is then used to encrypt the host name, which is embedded in the metadata and published to the blockchain. Meanwhile, the original data continues its journey through the network and is securely stored in the organization's database. Since the encryption key is linked to the data, only users with access to the original data can regenerate the key and decrypt the host name. This data-driven encryption mechanism ensures that network metadata remains highly secure against unauthorized access.

For instance, consider data passing through a node labeled "Temp." As metadata is generated at this node, the data itself is hashed to create an encryption key, such as "1Ma5HoizYnSDF5." Using this key, the host name "Temp" is encrypted (via the Advanced Encryption Standard (AES)) [4] into a string like "Tw2yMWYZKjyu." This encrypted string replaces the original host name in the metadata published to the blockchain. If an unauthorized party gains access to the metadata, they will see only the encrypted string instead of the actual

host name. To decrypt the host name, authorized users must access the original data to regenerate the key, as illustrated in Figure 3. Without this key, the encrypted host name is indecipherable, even if attackers retrieve the metadata from the DLT. This approach ensures that metadata remains secure even in an open, distributed environment. For instance, even if metadata for "Temp" appears repeatedly with different encryption outputs, the lack of a predictable pattern prevents attackers from deducing relationships or identifying critical nodes.

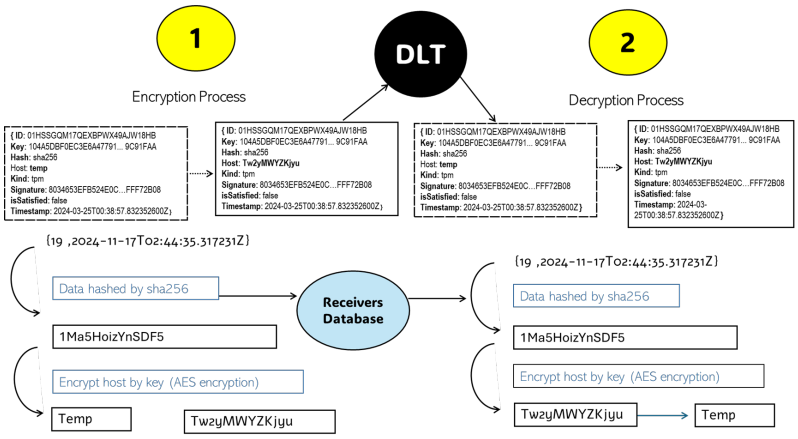



**Fig. 3.** Encryption of hostname using data as password

The Hostname Encryption method offers multiple advantages over the Hostname Mapping approach, particularly in terms of scalability and security. One key benefit is the elimination of the mapping table. By tying host name encryption directly to the data, this method removes the need for maintaining a central table with host-to-secret mappings. This reduces the storage requirements and management overhead associated with the table, making the system more scalable. It is particularly useful in large-scale networks where maintaining a central table for secret mappings becomes increasingly challenging and environments where real-time metadata publishing is essential, such as Internet of Things (IoT) networks. Additionally, this method minimizes the risk of collisions or repetitions in secret name identifiers. Each encryption key is uniquely derived from the data, ensuring that the encrypted host names remain distinct, even in large-scale distributed networks. Unlike the Hostname Mapping approach, where the same host's name might be assigned the same secret name across different metadata transactions to blockchain, the encryption-based method generates unique encrypted names for each metadata instance. This ensures that attackers cannot exploit patterns in the metadata to infer relationships or reconstruct the network topology. From a security perspective, the encryption-based method adds an additional layer of protection. Even if unauthorized parties gain access

to the metadata, they cannot decrypt the host names without the original data. This tightly makes dependency between the encryption process and the data itself significantly enhances the privacy of metadata.

## 4 Evaluation

To assess the effectiveness of our secret naming scheme, we conducted a series of experiments that compared the network's state before and after implementing the proposed methods. The primary objective was to evaluate how well the schemes obscured the network structure and reduced its predictability to potential attackers. Initially, we examined the metadata annotations generated by the Alvarium SDK [1] without any modifications. This served as a baseline, enabling us to identify clear patterns, paths, and relationships by analyzing the Host and Timestamp fields, as illustrated in Figure 4. Subsequently, we applied the proposed secret naming schemes to obfuscate metadata and repeated the analysis. This allowed us to compare the network structure before and after the obfuscation process.

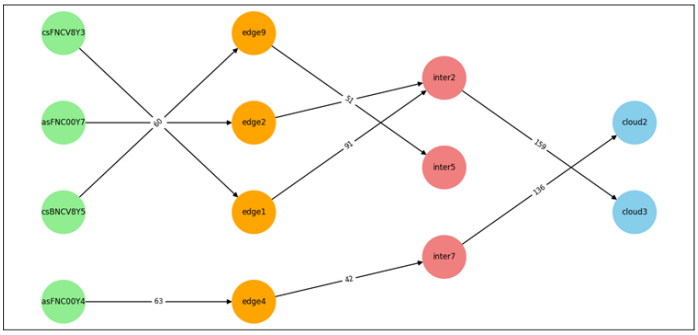



**Fig. 4.** The baseline network structure visualized using metadata fields: Host and Timestamp, reveals clear and easily identifiable patterns, relationships, and paths within the network.

After implementing the schemes, the results clearly revealed the complete network structure, including the identities of nodes and paths, addressing our initial concern. After implementing the secret naming schemes, the reconstructed network structure became significantly challenging to interpret. Visual analysis through generated graphs revealed that the recognizable patterns present in the unprotected metadata were effectively concealed as shown in Figure 5.

To further quantify the effectiveness of the schemes, we performed a comparative analysis using correlation matrices. The results showed that, in the unprotected metadata, strong correlations between node activities and network paths were easily discernible. However, after applying the obfuscation schemes,

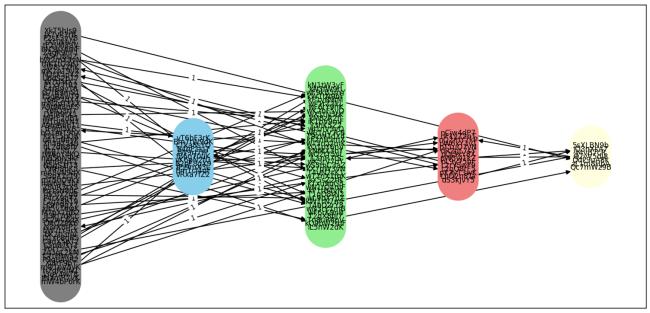


**Fig. 5.** Hiding network structure: nodes identity and paths by metadata obfuscation schemes. This figure illustrates the reconstructed network structure after applying the secret naming schemes, highlighting the transformation from recognizable patterns to a significantly obfuscated and challenging-to-interpret network.

these correlations were significantly reduced, as shown in Figure 6. This reduction demonstrated that the network nodes relationships and paths were no longer easily inferred from the metadata, confirming the success of the schemes in hiding the structure.

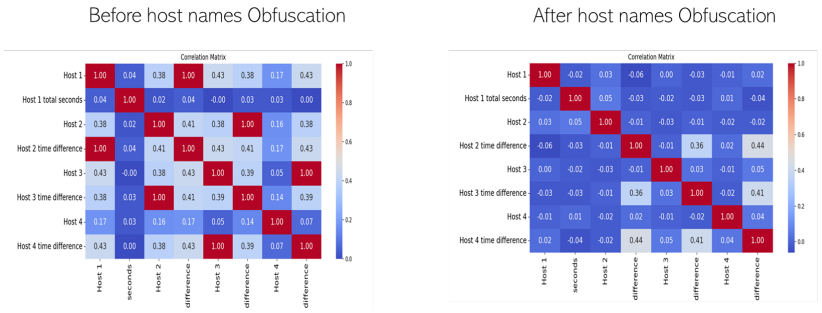



**Fig. 6.** Correlation analysis: before and after using metadata obfuscation schemes. This figure compares correlation matrices derived from the metadata before and after obfuscation, showing a substantial reduction in identifiable correlations, which underscores the effectiveness of the schemes in concealing network relationships.

## 5 Related Work

Data transmitted across networks, including social media platforms and financial systems, often leak sensitive information due to weak privacy measures and metadata vulnerabilities. Metadata fields like timestamps, activity logs, and network

topology, can be exploited to identify individuals, analyze behaviors, or expose organizational structures. Related papers include [10] [11] [5] [9].

Research by Wu et al. [13] introduced an innovative Bitcoin Transaction Network (BTN) model, based on extended safe Petri Nets, to enhance blockchain forensic investigations. The model analyzes static and dynamic transaction features and integrates a concept called "Bitcoin gene" to enable precise tracking and pattern recognition of cryptocurrency flows, helping to identify suspicious activities and improve financial security. Similarly, Ayoob et al. [2] developed a rule-based forensic tool for detecting illegal Bitcoin transactions using predefined patterns and thresholds rather than complex machine learning models. Leveraging the Elliptic++ dataset, they implemented the framework in Python to analyze transaction attributes like value, frequency, and connections to flagged addresses, achieving an 88% accuracy in identifying illicit transactions. These approaches demonstrate practical, efficient tools for combating illegal activities in cryptocurrency networks.

Our work differs from these studies in its focus on security through hiding the network structure. While these papers generally explore how to predict or infer network properties from metadata, our approach addresses the risk of attackers exploiting this metadata to reconstruct the structure. Specifically, we propose methods to obscure host names and prevent the prediction of the network topology by using secret names and encryption, which is a more security-focused approach compared to the network reconstruction strategies presented in these studies. Additionally, while some of the previous work discusses clustering for prediction, our work directly targets metadata manipulation to block attackers, to infer the network structure.

## 6 Conclusions

Our research is motivated by a critical security concern: while security metadata annotations may be a powerful tool for securing network data and ensuring its authenticity, they may reveal information that could be exploited to compromise the privacy of a network and its activities. To address this, we propose methods to assign secret names to hosts present in these annotations in order to obfuscate the history of data as it traverses the network. Through our experiments, we found that replacing real host names with these secret/encrypted names we can frustrate potential attackers from reconstructing the network structure. We also propose an approach using the access to the data itself as the basis for an accredited scheme for encrypting and decrypting host names. This method eliminates the need for a separate secret table by generating encryption keys from the data itself and using them to encrypt and decrypt the host name. Please note that hostname obfuscation does not prevent data processing, including the use for ML/AI algorithms, as the data processor will be able to restore the original hostnames. In future work, we will explore using the timestamp fields as the basis for pattern analysis – in addition to the approaches outlined in this paper – to determine if such analysis is effective. If such analysis proves effective, we

will explore further obfuscation techniques to attempt to defend against these attacks as well.

## 7 Acknowledgments

This work was conducted as part of the CLEVER project, EU Grant Number 101097560 and EI No: IR-2022-0065, and supported in part by Taighde Éireann – Research Ireland under Grant number 13/RC/2077_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

## References

1. GitHub - Project-Alvarium: Repository containing work related to the forthcoming Java SDK implementation. `https://github.com/project-alvarium/alvarium-sdk-java`, [Accessed 23-January-2025]
2. Ayoob, F.M., Jose, M., Kumar, S.U.: Forensic analysis and detection of illicit transactions in bitcoin network. In: 2024 5th International Conference on Smart Electronics and Communication (ICOSEC). pp. 673–679. IEEE (2024)
3. Baird, L., Harmon, M., Madsen, P.: Hedera: A public hashgraph network & governing council. White paper at hedera.com (2020), [Online; accessed 23-January-2025]
4. Bos, J.W., Özen, O., Stam, M.: Efficient hashing using the aes instruction set. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 507–522. Springer (2011)
5. Hric, D., Peixoto, T.P., Fortunato, S.: Network structure, metadata, and the prediction of missing nodes and annotations. Physical Review X **6**(3), 031038 (2016)
6. IOTA Foundation: IOTA Tangle. `https://www.iota.org/get-started/what-is-iota/`, [Online; accessed 23-January-2025]
7. Khalid, A., Murphy, S., Sreenan, C., Roedig, U.: Efficient data confidence fabrics with compact annotations. In: Proc. of Cloud-Edge Continuum Workshop at IEEE Intl. Conf. on Network Protocols (ICNP) (2024)
8. Lovato, J.: New Linux Foundation effort to focus on data confidence fabrics to scale digital transformation initiatives. Press release at linuxfoundation.org (2019), [Online; accessed 23-January-2025]
9. Meier, R.: Improving Network Security through Obfuscation. Ph.D. thesis, ETH Zurich (2022)
10. Salem, A., Banescu, S.: Metadata recovery from obfuscated programs using machine learning. In: Proceedings of the 6th Workshop on Software Security, Protection, and Reverse Engineering. pp. 1–11 (2016)
11. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE symposium on security and privacy. pp. 305–316. IEEE (2010)
12. Steve Todd, Dell Technologies Inc.: Project Alvarium: The future of edge data. Available from learning.dell.com (2020), [Online; accessed 23-January-2025]
13. Wu, Y., Tao, F., Liu, L., Gu, J., Panneerselvam, J., Zhu, R., Shahzad, M.N.: A bitcoin transaction network analytic method for future blockchain forensic investigation. IEEE Transactions on Network Science and Engineering **8**(2), 1230–1241 (2020)