CURRENT RESEARCH AND FUTURE POTENTIAL OF BLOCKCHAIN SEARCH ENGINES FOR IOT NETWORKS

[1]V. Vijayabhaskar, Assistant Professor, Department of Science and Humanities R.M.K. Engineering College, Kavaraipettai, Thiruvallur District, Tamil Nadu 601 206, vvbhaskar75@gmail.com

[2]A.Prithiviraj, Assistant Professor (Sr.Grade), Department of CSE, Sona College of Technology, Salem-5.prithivipranav@gmail.com

[3]D Balamurugan, Associate Professor, Computer Science and Engineering, Sona College of Technology, Salem, balamurugand_81@yahoo.com

[4]R.C.Narayanan, Associate Professor,Computer Science and Engineering, Sona College of Technology, Salem 5, rc.narayanan@gmail.com

[5]Kamatam Hari Prasad, Associate Professor, Department of Physics, Institute of Aeronautical Engineering, Hyderabad – 500003, INDIA, hariprasad.kamatam@gmail.com

[6]Swapna Datta Khan, Associate Professor, Globsyn Business School, Kolkata, captsdk@gmail.com

[7]G. Purushothaman, St. Joseph's College of Engineering, Chennai -119, gpmanphd@gmail.com

[8]B. Kannadasan, Assistant Professor, Civil Engineering, B.S. Abdur Rahman Crescent Institute of Science and Technology, Vandalur, Chennai-600048, Tamil Nādu, India, kdasan9@gmail.com

ABSTRACT:

Due to the proliferation of cryptocurrencies and decentralized apps, blockchain has recently sparked widespread attention and worry. It functions as a database and is used to keep track of a huge number of entries in a time-ordered fashion for use in data-heavy applications. It is proposed in such a scenario that search engines be used to retrieve the data stored in a blockchain and then analysed for various reasons. However, there hasn't been any comprehensive study of blockchain search engines yet, and the field is still in its infancy. Both the present state of blockchain search engine development and its potential in the Internet of Things (IoT) space are examined in this article. The paper's primary focus, from a research standpoint, is on the categorization of blockchain search engine techniques and a comparison of their relative performance. Current search works connected to IoT domains are introduced with the future potential of blockchain search engines in this space. The remainder of the paper provides a history of blockchain technology. Then, the search necessities are laid down, such as the basic rules of categorization. We then provide a problem analysis of blockchain search in IoT domains and a comprehensive overview of previous work in this area. We conclude with a discussion of the difficulties inherent in blockchain-based searches and an outlook for future research initiatives in IoT-related fields.

Keywords: Blockchain search engine Search fundamentals Search requirements.

DOI Number: 10.14704/nq.2022.20.12.NQ77013 **NeuroQuantology 2022; 20(12): 120-150**

1. INTRODUCTION

The blockchain technique that forms the backbone of cryptocurrencies like Ethereum or Bitcoin [2] can be credited for the rapid and amazing explosion of technological advancement in decentralized computing. This system is based on a peer-to-peer (P2P) distributed network with a wide variety of cryptographic primitives (such as digital signatures and asymmetric encryption). Users that don't trust each other can freely engage, maintain track of financial activities, and agree on communal wealth using blockchain because no third party is required for mediation or verification [6,7]. This means that the blockchain network can potentially use a consensus mechanism to build a reliable, append-only database. Data is added

eISSN 1303-5150

to the block and sent to nodes in the network only after it has been confirmed and verified. Smart contracts have enabled the expansion of blockchain technology beyond the domain of cryptocurrencies into fields as diverse as medicine, government, supply chain management, the energy sector (through the smart grid), the Internet of Things (IoT), implantable medical devices, and more [9,10]. The current state of blockchain-based systems has great promise for a variety of applications, including the reduction of operational expenses, the identification and prevention of fraud, the prevention of tampering, and the enforcement of contracts [11].

Transactions involving the exchange of money can now be recorded and kept in a decentralized ledger known as a blockchain. The hash of the previous transaction and other metadata may vary between blockchain implementations and make up a node's digital signature. For instance, the hash value of a previous transaction and an index to identify the output are both included in the input of a signed transaction in Bitcoin and other blockchains that follow the Unspent Transaction Output (UTXO) paradigm. This guarantees that monetary transactions are secure against corruption or destruction. The analytical tool can quickly discover and verify all transactions that have taken place at a certain address, such as those in a wallet, because the blockchain acts as a public record. Although most blockchain-based schemes only permit limited search capabilities at the moment, blockchain technology may be used to eradicate the risk of fraud in data storage. More specifically, the blockchain keeps its records in order of time.

All per participant nodes need to either comb through every block in the blockchain or use metadata/index of blocks to acquire a comprehensive search result in response to a query. The records can be retrieved by randomly searching the disk's vast number of files or by adopting a rudimentary block index, but this is an extremely time-consuming search technique. Data providers collect a greater and larger data set from which to draw conclusions as more and more smart devices reach the market. The apparent drawback is that it complicates targeted, rapid, and privacy-preserving searches for chosen data and datasets among the mountain of data stored on blockchain. Even if there are a plethora of large datasets to choose from, not all of them are conducive to guiding decision-making. Furthermore, the analysis process calls for one or more custom-built datasets or dataset combinations to accomplish one or more custom-built goals in any number of different application domains.

This emphasizes the need of providing a means of sifting through data on a blockchain via a search interface. Due to blockchain's intrinsic restrictions in throughput, storage capacity, scalability and flexibility, latency, size, and bandwidth, how to efficiently search data is a significant topic in the field of blockchain systems study. Security of object data in a spatial database or cloud is becoming increasingly important as more and more things are connected to the internet of things (IoT) [12]. However, the volume of data generated by these networks and stored in various data warehouses and clouds [13,14] is becoming increasingly problematic. With the use of the cloud, blockchain may be used to store information from a wide variety of IoT-related application domains, not only that provided by the devices themselves. As shown in Fig. 1, edge devices may encrypt and collect data from sources including sensors, monitors, and cyber-physical systems by creating and storing various local spatial indexes. Next, the edge devices verify the data transfer to the warehouse using the blockchain, which stores the smart IoT device's and edge device's identifiers in addition to the storage address.

eISSN 1303-5150

A user's identity is validated and the transaction is permitted by the blockchain before it is sent from their device to the edge node. The edge device will then receive the query results using the storage address and provide them to the requesting node after the transaction has been verified and recorded into a block. It is important to note that the edge device at the fog layer acts as a proxy between the IoT device layer and the cloud layer because to its better compute, communication, energy, and memory capabilities. For optimum connectivity with the edge device, we've set up a blockchain and a warehouse in the cloud. Therefore, an edge device may allow for low-latency interaction between IoT devices, blockchain, and the warehouse. Create a robust blockchain search engine that can be used in IoT environments, since the storage of IoT data via blockchain is on the rise.

To swiftly and anonymously access block or transaction information in answer to a given query, a blockchain search engine comparable to existing search engines is necessary. Without a doubt, the index structures that fuel search engines are indispensable. However, due to blockchain's intrinsic limits, such as those on throughput, storage capacity, scalability and flexibility, latency, file size and bandwidth, wasted resources, and usability, creating an efficient index structure in blockchain is difficult. It's impossible to do difficult jobs in a timely manner when a large amount of data is being received.

This makes it difficult to integrate blockchain data into other systems and make use of it elsewhere. Sometimes it's hard for users to get their hands on blockchain data, much less figure out how to use it. While there are now several well-known blockchain search engines, they all have the drawbacks common to research at an early level and are not yet being directly used to blockchain search in IoT-related application areas. To the best of our knowledge, there is not yet a synthesis of the current status of search research in blockchain that can be found in existing studies and surveys. We give a comprehensive literature analysis on the studies that are pertinent to this study, and we offer a look at potential roadblocks and future research possibilities in the areas of application that are connected to the Internet of Things.

As a quick recap of this paper's findings, here they are:

• An introduction to the search literature pertinent to the Internet of Things is provided in the first part of this research.

• Then, we go over the basics of blockchain technology, including search.

• In what follows, we conduct an extensive literature review of blockchain search engines already in existence. While there are advantages to using blockchain technology for reliable data storage, we also detail the drawbacks of this approach to research. We use this to analyse various issues with blockchain-based searches in Internet of Things settings.

• We make some suggestions for future study into blockchain search in IoT application domains, taking into account the recent ongoing of search engine in different domains.

eISSN 1303-5150

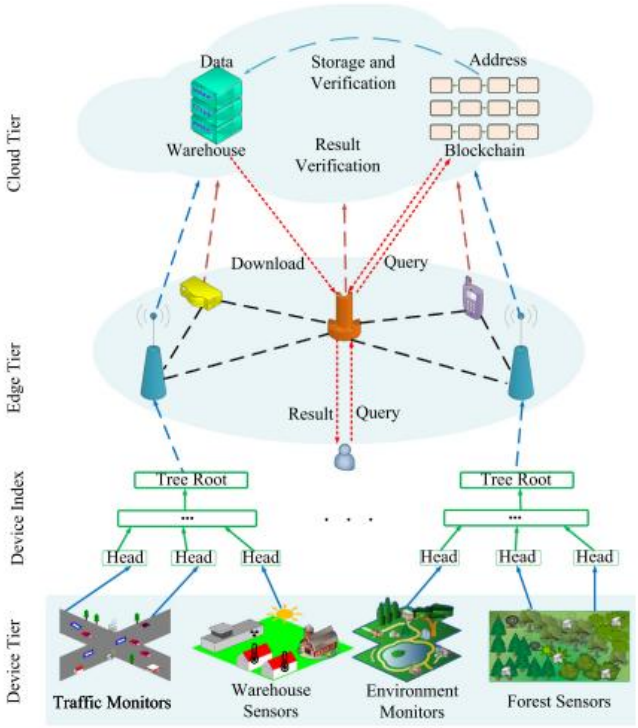

Fig. 1. System structure for blockchain storage and search in IoT-Edge-Cloud collaboration [1]

2. SEARCH WORKS RELEVANT WITH IOT NETWORK

For this part, we take into account I the spatial-temporal feature of IoT objects, and (ii) the relevant IoT-Edge-Cloud environment, and examine the state-of-the-art search operations pertinent with IoT network.Finding Information in a Geographic Database: For example, "reviews of Los Angeles restaurants in March," "hotels and bars in Chicago in the summer," "San Francisco's weather in spring," and so on are all examples of real-world queries that people are interested in information about certain locations, time intervals, and keywords associated to. Searches that combine textual, geographical, and temporal criteria are known as spatial-temporal-keyword inquiries. The B-tree [15,16], R-tree [17], KD-tree [18], and its variations are all examples of hierarchical index structures that are commonly used to record spatial locations for the query only with spatial constraint.

The B-tree is a helpful tool for indexing and searching one-dimensional data since it maintains data order and permits sequential access, insertions, and deletions in logarithmic time. Indexing and searching multi-dimensional data like geographical coordinates, rectangles, or polygons may be done with the R-tree, an extension of the B-tree. Assuming an initial bulk is loaded, and then no changes occur in the succeeding data, the KD-tree is optimal for scenarios when the data is low-dimensional and static in practice. For spatial data in two or more dimensions, various methods translate the keys to a single-dimensional key based on space-filling curves (e.g., Z-order, Hilbert, Peano) [19,20].

Using this, a spatial-temporal-keyword index is created for accessing data items satisfying a spatial-temporal or keyword-related query constraint, such as an IR-tree [21], SIF (Spatial Inverted File) [22], a decaying inverted quadtree [23], and so on. To retrieve documents, IR-tree [21] uses a unified representation of spatial and textual relevance's. Using a block-based inverted file, SIF [22] expands IRtree to handle top-k spatial-textual searches in bulk. The spatial-temporal-keyword queries suggested in [23] can be executed in real time using the index structure of a decaying inverted quadtree. Using a quadtree structure, the scalable

eISSN 1303-5150

integrated inverted index [24] partitions the data space into cells containing relevant keywords and thereby captures the keywords' spatial proximity. In [25], researchers introduced a new type of index structure, which uses an inverted list to combine the textual and temporal aspects of documents. In [26], a shallow tree is used to narrow the search space in the temporal dimension by using time stamps, in addition to scanning the spatial and textual dimensions. When combined with keyword information and the temporal dimension, the two hybrid index structures introduced in [27] for processing spatial-temporal and spatial-keyword searches can also enable spatial-temporal-keyword inquiries. In [28], the spatial, temporal, and keyword aspects are all handled inside a single index structure, allowing for quick and easy resolution of spatial-temporal-keyword searches. In [29], the authors examine a geographic keyword query with temporal awareness, and they develop an efficient index structure and related algorithms to reduce the search area in terms of both spatial-temporal and textual characteristics. Unfortunately, the results of these research cannot be immediately used to the search of spatial-temporal-keyword associated items stored in blockchain since the index size may not be sufficient to the restricted blockchain storage.

Roadside social-network-based querying: In order to meet the rising demand for these sorts of services, location-based social network programs like Yelp, Google Maps, and Facebook Places have sprung up in recent years. The users of the social network are always linked to their road network locations, such as stores, homes, workplaces, etc., in such apps, creating a hybrid road-social network. Distance and range inquiries [33], reverse top-k Boolean spatial keyword queries [34], collective spatial keyword queries [35], and aggregate closest neighbour queries [36] are all common forms of queries performed on road networks. Similar approaches for searching road networks provide up exciting new possibilities for users of social networks to quickly organize and participate in spontaneous offline activities. The goal of the cohesion group closest neighbour query [37,38] is to return the query users with the highest proximity and the matching top-k items on a road network by leveraging social cohesion between them.

In order to find a collection of skyline cohesive groups where no one group may dominate the others, researchers have developed the skyline cohesive group query [39]. The authors in [40] propose a novel index structure PIM-tree for algorithm design and pruning, which combines a Partitioned Road network, an Inverted Intersection Union (I2U) file, and a Check-in and Friendship Matrix, and uses a top-k geo-social keyword query to retrieve the objects with the spatial-social-textual constraint (CFM). The index size may not be acceptable to the restricted blockchain storage, thus while these studies do answer certain issues associated with geographical, social, or keyword limitations on road-social networks, they cannot be directly utilized to search spatial-social-keyword related items stored in blockchain.

IoT-related keyword: The Internet of Things (IoT) paradigm links physical things in the real world to the virtual world, paving the way for the development of smart environments and applications [41-44]. Due to their interconnectivity and data-processing capabilities, the billions of smart devices that make up an IoT network generate a vast quantity of data [45-47]. In light of this, analysing such a massive dataset is a crucial but challenging endeavour. By employing search methods, a program may cherry-pick the information that best suits its needs [48, 49]. Search methods are therefore essential to the IoT, but they also present a number of difficulties, including the aforementioned huge volume, heterogeneity of devices,

eISSN 1303-5150

dynamic availability, resource restriction, and real-time data created in a wide variety of forms and kinds [50-53]. Data searches in IoT sensing networks need to account for query efficiency, latency, energy usage, and communication cost.

Most of the existing literature on IoT search engines uses a layered architecture [54–56] to handle queries with basic and restricted limitations, such as keyword searches [54–56], location-based searches [55–57], and state-based searches [58]. Micro search [50] and google [55] both employ a keyword-formatted sensor node to store entities, from which users can then query matching entities by using the keywords. However, the suggested method suffers from severe limitations in its data transfer manner, making it unfit for use in a highly dynamic, expansive network setting. The MAX [56] search system can detect real-world objects through tags and adjust to mobile queries with dynamic information. Broadcasting messages, however, creates significant communication cost in large-scale network setups. For real-world entities in a given state, Dyser [58] uses keyword matching.

Adopting a predictive method can boost searching efficiency while cutting down on expenses. As a result, Dyser works well in environments where there are constraints on the available networking resources. In recent years, researchers have been studying the spatial-temporal restriction in addition to the keyword limitation in IoT search engines. To address this issue, the developers of IoT-SVKSearch [59] created a master index server and secondary index node servers. In order to index the full-text keywords or the time-stamped, dynamic positions of moving objects, a collection of hierarchical trees is generated in each index node server. In order to handle range and (improved) k closest neighbour queries, the SMPKR (Search Mechanism over PKR-tree) search engine [60] creates an encoding-enabled index approach that takes into account the spatial-temporal-keyword closeness. Unfortunately, because to the storage limitations of blockchain, these findings cannot be immediately applied to the search of IoT data stored in blockchain.

IoT edge and cloud computing search collaboration: Furthermore, it is necessary to execute data analysis as close to the smart devices or base station as feasible to decrease the processing and transmission latency in IoT networks. Fog computing [61–63] refers to distributed computing that makes use of locally available resources, and it has the potential to extend cloud services to the network's periphery, where they may be used for things like low-latency online analysis of large amounts of data in real time. The method suggested in [67] uses fog computing and a priority-aware index tree to process a flood of continuous multi-dimensional top-k searches in WSNs. When the cloud and WSNs are linked via a fog layer, near-real-time data transfer from WSNs to the cloud is possible. Two types of spatial indexes are built in CECSE (Collaborative Edge-cloud Cache Search Engine) [68] to improve search efficiency and decrease search latency by identifying and classifying activity regions with different frequencies.

The reasoning for this is because using fog and cloud caching may greatly reduce network traffic and service latency in IoT networks [69-71]. In [72], the authors offer a new method for cloud-edge storage that allows for safe data searching and sharing. While the system does provide for more security, it does cause additional computation and communication overhead when uploading and exchanging data. Generation of keyword search trapdoors reduces data search's communication and computational cost. To provide fine-grained access control and search, LFGS (Lightweight Fine-Grained Search) [73] moves some of the storage and compute burden from end users to strategically placed edge nodes. In [74], we

eISSN 1303-5150

find a data processing system that uses both fog and cloud computing to centrally manage data preparation, storage, and search.

At initially, the edge node processes all of the unprocessed data. Data that can't wait are used and stored locally, while data that can be searched using a retrieval feature tree (RF-tree) or a hash value tree are uploaded to the cloud (ID-AVL tree). As a result, the suggested method increases retrieval efficiency while decreasing network and cloud data transmission and storage needs. To ensure the semantic security of outsourced ciphertexts, ESPE (Edge-aided Searchable Public-key Encryption) [75] takes use of the edge-cloud architecture and provides a searchable public-key encryption technique with the help of edge. With this method, the expensive cryptographic operations of industrial IoT devices may be offloaded to the neighbouring edge node for rapid computing, drastically lowering the costs of transmission and decryption. Unfortunately, because to blockchain's storage limitations, these research' conclusions about improving the efficiency, latency, security, and privacy of search in collaborative IoT-Edge-Cloud storage cannot be extended directly to searching IoT data stored in blockchain.

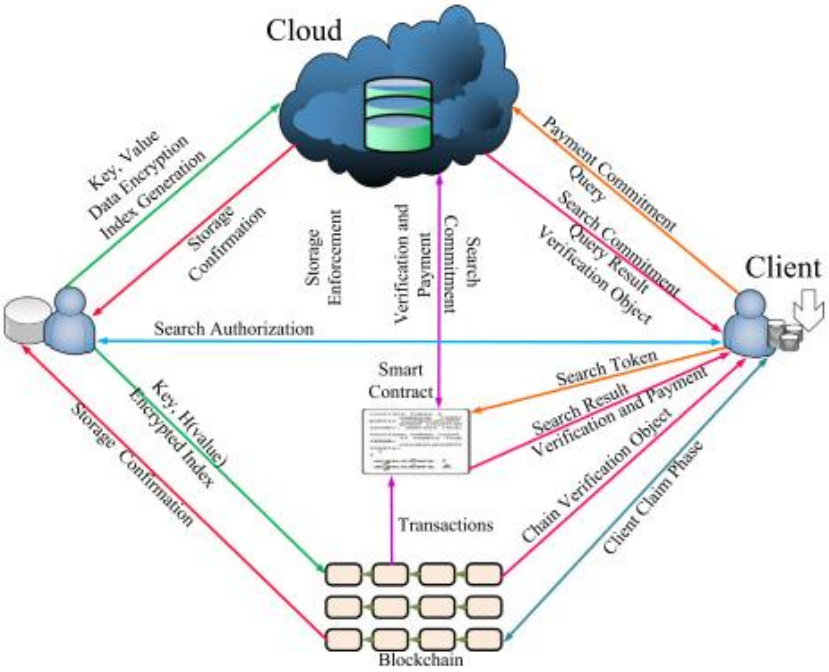

Fig. 2. System structure for search in cloud computing via blockchain. [1]

A growing number of people and businesses are eager to transfer their data to cloud computing platforms due to its endless storage and processing resources, low costs, and ease of use [76–79]. It is necessary to encrypt data prior to outsourcing it to the cloud in order to protect user privacy and maintain search capabilities [80–82]. Many essential and difficult challenges, such as user-server-side verifiability and a fair payment mechanism without a trusted third party, have not been thoroughly studied in existing search algorithms over encrypted data. As an illustration, in the universal payment model, a malevolent cloud may deliver incorrect results or even mismatched papers in order to cheat for financial gain, while the client would be left holding the bag for not receiving the right result and hence incurring financial loss [83]. This seems to necessitate a more secure search technique over encrypted data, one that can not only identify illicit cloud activity but also safeguard client benefits with an in-built fairness mechanism.

To combat issues like hostile or hacked nodes [84], online keyword guessing

eISSN 1303-5150

attacks [85], unfair payment [86-88], data sharing and key leakage [88-90], and data integrity and security [91-93], blockchain approaches are increasingly being considered. In Fig. 2 we see the four components of the search system: the data owner, the client, the cloud server, and the blockchain. When storing information in the cloud, the data's rightful owner generates a password-protected index and encrypted copy of the file. With the aid of blockchain, the client conducts a search query and uploads it to the cloud for result return, where it is verified and fair payment is achieved. All these projects use blockchain to address issues with trust and compensation. However, this creates expenses for blockchain in order to ensure safety and equity. we survey the related researches on search in blockchain, as the application of blockchain technology can bring some benefits for search in cloud computing, but it is more efficient and possible to ensure these search merits directly from blockchain storage because of the following advantages.

The blockchain technology functions normally in a decentralized network where no central authority is required.

- All users may inquire about past transactions, making Blockchain a trustworthy system. Pseudonymity is achieved in the Blockchain system by recording transactions using publicly available pseudonymous addresses and by concealing the identities of nodes. The decentralized, peer-to-peer nature of the blockchain system makes it possible for all nodes to equally weigh in on choices. A blockchain's smart contracts can execute tasks including generating transactions, making decisions, and storing data mechanically. Despite its decentralized nature, the Blockchain system guarantees reliability through the use of a consensus procedure executed by all participating nodes. Payment integrity is ensured using the Blockchain technology, which eliminates the need for a trusted third party.

3. BLOCKCHAIN BACKGROUND

Blockchain is a technique for facilitating consensus between numerous parties in regards to incremented blocks of data. To create the blockchain's underlying data structure—the hash chain—each block must first be hashed. This process may be encrypted using a hashing function in cryptography.

Constituent parts: It is important to remember that the user creates a transaction by feeding in a sequence of UTXO, which reflect the current state of the Bitcoin network. Meanwhile, Ethereum uses a 160-bit identifier called the State Root to describe the global state. Blocks in a chained structure, which are built using identical blocks, often contain the following data fields, as shown in Fig. 3. In the first, known as the block header, the following components predominate:

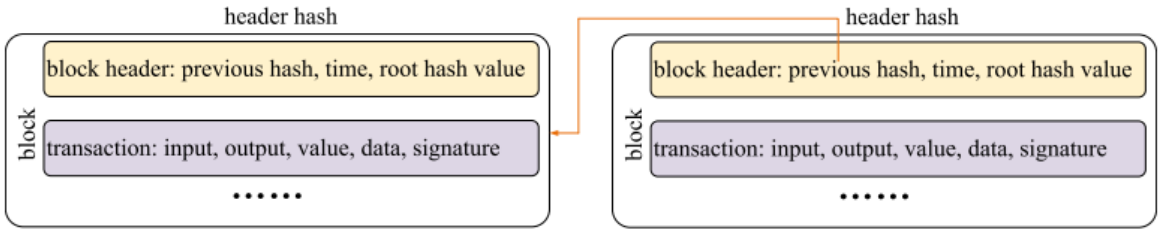

Fig. 3. A chained structure of blockchain [1]

eISSN 1303-5150

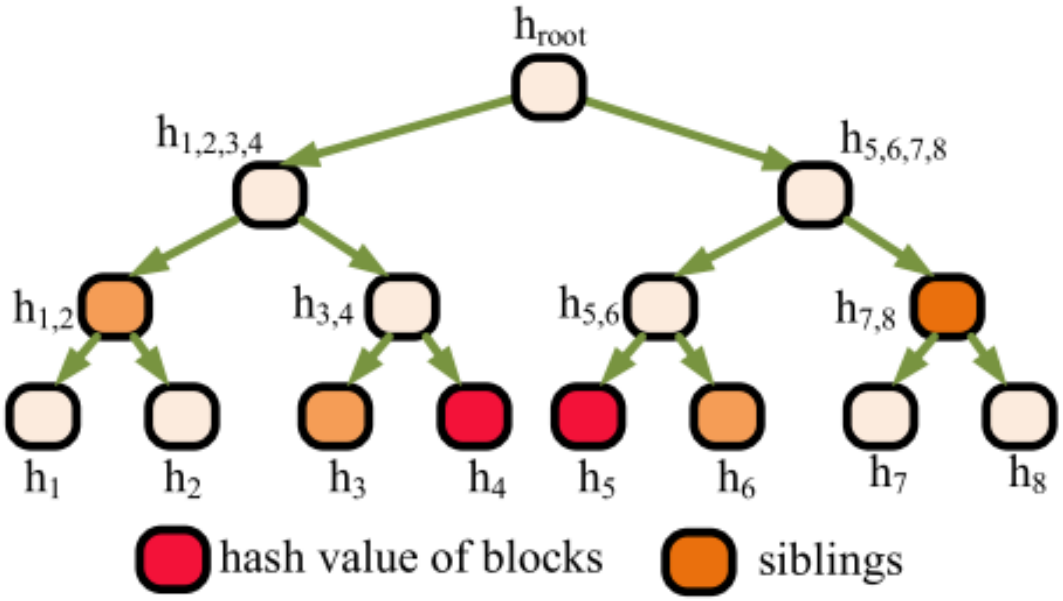

Fig. 4. Merkle Hash Tree [1]

For the hash chain to work, the "previous hash" must be the hash value of the prior block.The time stamp indicates the precise moment the block was inserted into the chain. MerkleRoot: It is the root hash value computed from all the transactions in the current Bitcoin block using the Merkle Hash Tree, as illustrated in Fig. 4, or its variations such as the Merkle Patricia Trie Tree [94], the Merkle B-tree [95], or the Merkle KD-tree [96].

• StateRoot: This is the hash value at the tree's root that is calculated using a Merkle Patricia Trie Tree from the hashes of all Ethereum transfer transactions. TxRoot is the root hash value of the current Ethereum block's transactions, calculated using a Merkle Tree. ReceiptRoot is the Merkle Tree root hash value of all the receipt data in the current Ethereum block.

It is important to remember that the user creates a transaction by feeding in a sequence of UTXO, which reflect the current state of the Bitcoin network. For Ethereum, the state is represented by a "wallet," a unique identifier that is 160 bits in length. The second provides the current block's worth of transactions, which mostly consists of the following: The input is the address where the communication is beginning. In other words, the output is the final destination's physical address. The number of input tokens that were successfully transmitted, denoted by "Value." What we mean by "data" in this context is the information provided in the input's attachments. After a transaction has been verified and recorded in a blockchain, the value of the data it represents can no longer be changed or falsified. The length of transaction data in Bitcoin varies with the value of the initial byte, but with Ethereum, there is no such restriction. Signature is a value attached to a specific transaction that has been signed using the input key. In addition, there are several blockchain architectures that enhance blockchain functionality in the following ways: I knew block types, such as Bitcoin-NG [97], which seeks to increase Bitcoin's throughput by implementing Key blocks and Macroblocks; (ii) new block structures, such as Fruit chains [99], which provides an alternative structure to the blockchain and separates the processing of transactions and blocks; (iii) new block types, such as Bitcoin-NG [97], which seeks to increase Bitcoin's throughput by implementing Key blocks and Macroblocks; (iv) new

eISSN 1303-5150

Mining blocks: Data is added to the blockchain through the mining process, which consists of producing blocks. The most common method of mining in practice is the proof of work [103], which seals off the blocks and makes the transaction history unchangeable. Complex mathematical puzzles, which are updated periodically, must be solved by mining nodes and accepted by all miners. After the node verifies the transactions and solves the puzzle, the block is sent to the blockchain. To further ensure that the deliverer is not forging the block, other mining nodes verify its integrity. After the miners validate a block, it is added to the blockchain and the sender receives payment in bitcoin. In this case, consensus has been reached by the greater number. Nakamoto's Consensus [104], the GHOST (Greedy Heaviest Observed Subtree) protocol [105], the POS (Proof of Stake) protocol [106], and the DPoS (Delegated Proof of Stake) protocol [107] are all examples of consensus protocols used to verify nodes in the public blockchain before they can participate in block mining. However, in consortium blockchains or private blockchains that don't rely on the monetary system, the nodes have stricter standards for the reliability of each other. Because of this, the classic consensus algorithms like PBFT (Practical Byzantine Fault Tolerance) [108], Paxos [109], Raft [110], etc., have largely replaced their more modern counterparts. The blocks that include the transactions and state transfers are generated by the consensus algorithm, making it crucial to the blockchain's functionality [111]. Unless more than half of the mining nodes are hacked, it is difficult to forge using this method. Therefore, blockchain creates a trustworthy network where members may jointly preserve the block information without having to put their faith in any one node.

Conventional Contracting Meets the Future. Nick Szabo initially proposed the notion of a smart contract in 1997 [112]. A smart contract is a computer program consisting of a set of trustworthy symbolic protocols. There is no need for an impartial third party to oversee the operation of the contract. Smart contracts provide a large number of APIs and are designed to handle increasingly complex scenarios. Using these APIs, programmers may create sophisticated protocols for communication between nodes [113]. The miners run the code, and the blockchain consensus process ensures the validity, semantic equivalence, and non-tampering of the contracts they execute. Ethereum's smart contracts run on the blockchain in a bytecode format and are executed by a Virtual Machine [114]. In Ethereum, where data cannot be rejected or altered, miners must agree on how to store data and carry out calculations. Additionally, the smart contract is implemented to lessen the burden of administration and service expenses while simultaneously increasing safety.

Credible information architecture: There has been much research on verifiable query processing algorithms [115] to ensure the integrity of the query result that is challenged by a trust less service provider. Generally speaking, methods based on an authenticated data structure are more effective because of how they optimize for individual requests. Digital signature [116] and Merkle Hash Tree [117] are two of the most fundamental data structures that are used here to act as authenticated data structures. A digital signature is a kind of message authentication that is generated using asymmetric cryptography. It is not scalable to huge datasets since each data record must be signed in order to provide verified searches. The structure of a Merkle Hash Tree is analogous to that of a tree hierarchy. Each entry in a leaf node is given the data record's hash digest, whereas each entry in an internal node is given the digest computed from its children.

eISSN 1303-5150

Meanwhile, the data owners sign the root digest of the Merkle Hash Tree. Using this, a minimal set of verification objects or hashes may be provided to confirm the authenticity and integrity of the data records included inside blocks. Because of this, with some tweaks, Merkle Hash Tree may be used in blockchain systems. [118] Access to the blockchain: There are two broad kinds of blockchain systems based on how they manage user access to the network: Bitcoin and Ethereum are examples of permissionless blockchains, whereas Hyperledger-Fabric and Ripple are examples of permissioned blockchains, which restrict access to the network and its contents based on a set of access restrictions. Since each blockchain uses a unique database and data storage method, a search that works for one blockchain may not work for another. We'll talk more about this as part of our analysis of relevant search works below.

4. SEARCH FUNDAMENTALS

4.1. Classification of search techniques

This section provides an overview of how blockchain search engines are currently being studied and categorized. Current works mostly employ the following three structures to boost search performance, all of which are based on the core data organization and storage structure. Improved search efficiency is achieved through the use of a fingerprint/tag/pointer structure to obtain and return query results from the distributed data storage. In order to enhance search performance, we have restructured our databases and now get query results by moving data to new locations.

• Authenticated index structure: Authenticated index structures are used to obtain and confirm the soundness and completeness of query results. Current studies mostly concentrate on keyword-based search, along with some additional search contents investigated, such as transaction, history, balance, Boolean range search, time-window search, spatial-temporal search, and so on, depending on the data's applicability and utility in various situations. The retrieved transactions are based on the keywords used in the search. For instance, you may look for all the purchases that included the word "Beijing."

• Search that doesn't rely on keywords to obtain results identifies financial dealings by their structure and other characteristics. For instance, finding the deals at a specific "time instance" that describe "picture" information.

4.2. Search requirements

When a blockchain application is built, it is tailored to a particular industry so that it can carry out the unique activities required by that industry. As a result, it creates a concrete goal that takes into account the needs stemming from the aims of the sector. In order to give better solutions to achieve these goals, search for the blockchain apps should account for search needs. The search system you use should take into account your need for efficiency and offer your ideas that will help you achieve that goal. Time is money;thus, a search engine has to factor in how long it takes to do a search on the blockchain and how long it takes to compute the results.

• Throughput: Search systems need to think about how many requests/transactions they have to execute in a certain amount of time. The price of sending a query to a database and receiving the results back should be kept to a minimum for a search engine to be efficient. The computational effort required to execute and validate certain operations in Ethereum's search system is a good proxy for the gas cost associated with doing so. The search engine must take into account the amount of privacy protection required by the user and the data supplier. It seeks to protect the privacy, security, and accessibility of data stored on a blockchain.

eISSN 1303-5150

This feature ensures that the information stored in a blockchain remains private. When doing a search or gaining access to only certain blocks of data, the search pattern might be concealed.

• Soundness: This feature ensures that any attempt by a server to breach the agreement will be discovered. All of the returned search results must also originate from the data provider and meet the criteria of the original query. This feature protects sensitive information from prying eyes by keeping search requests and data files private. Additionally, it strives to conceal from the attacker whether or not the newly inserted document contains the previously searched-for data.

• Integrity: This ensures that information cannot be altered while it is in storage or in motion.

• Verifiability: This feature is used to check that search results are accurate and comprehensive.

• Access control is the ability to limit who has access to which resources and when. If the clients are legitimate, the data supplier will also provide them the decryption key and permission to access the data. Here, the data supplier has completed say over who may access their information. All parties engaged in a search system need to keep in mind that only when users pay for both their search tasks and access to the data provider's data will they receive accurate results. An examination of the incentive system in financial applications provides the foundation for a fairness conclusion. Loosely speaking, the process of fairness compels all parties to make accurate calculations. If one of the parties breaches this contract, he will receive no benefits: To summarize, in both the single-user and multi-user settings, the client should acquire accurate search results if he pays for his search tasks and the access to the data provider's data, while the executor and data provider should make money if they honestly comply with the agreement. By making the transcription public and selling it for things like search tokens, the data supplier benefits financially.

5. SEARCH IN BLOCKCHAIN

The current state of the art in blockchain search engine research is discussed. We define the associated idea for each of the examined works of literature and elaborate on its storage structure, query type, search strategy, and drawbacks. Due to its inflexibility, lack of scalability, and limited storage capacity, blockchain is unable to handle complicated activities in a timely, cost-effective, and user-friendly manner. In order to facilitate efficient and privacy-preserving query, some researchers have proposed using fingerprint/tag/pointer et al. to access the encrypted data in the decentralized storage [119–124], or reorganizing the block and transaction information into the designed databases [125–128] and then verification [96,129–131].Indexing and retrieval using fingerprint/tag/pointer structure 5.1

An elegant way for a client to access the requested encrypted data is searching based on fingerprint/tag/pointer structure. While the encrypted keyword fingerprint/tag/pointer is kept by the permissionless/permissioned blockchain, the encrypted dataset is saved in a distributed and decentralized way. As a result, the encrypted keyword fingerprint/tag/pointer may be utilized for indexing, allowing for a speedy means of searching for relevant encrypted documents. In order to speed up the search process and prevent downloading unnecessary data, clients interact with the nodes and content in a blockchain to obtain a fingerprint of the exact data content in response to a query. The client then accesses the distributed data store to get the fingerprint/tag/pointer-based encrypted documents. Structure of the relevant search system is depicted in Fig. 5.

Do et al. [119] use the blockchain technology to implement a search engine that provides a safe keyword search service over a network of databases. The content of the paper is not stored in the blockchain. So that clients may connect to the distributed hash table through blockchain, it simply stores the fingerprint generated using the hash function. The blockchain only stores the document that contains the ciphertext keywords. The suggested search strategy checks if the trapdoor keyword matches the ciphertext encrypted by the smart contract or the peers of the blockchain. The system uses distributed proof-of-retrievability to guarantee the reliability of its data storage, and it provides data consumers with a method of obtaining credentials that protects their anonymity. However, no review or comparison of search performance has been provided by the authors. Searchable encrypted data kept on a permissioned and distributed ledger blockchain network is made more efficient via the use of keywords with the help of a unique framework presented by Tahir et al. [120]. To prevent separate attacks and ensure search security at higher levels, the suggested system makes advantage of the trapdoors produced by a master key, an index table, and a smart contract. When using the trapdoors, only an encrypted set of transaction IDs is detected and returned, which must be decrypted in order to recover the transaction id and hence get the appropriate data blocks. If numerous linked transaction ids are returned, however, it is necessary to sequentially scan the connected data blocks, which is a time-consuming process. Additionally, the search performance is uncertain because the authors have not provided any assessment and comparison.

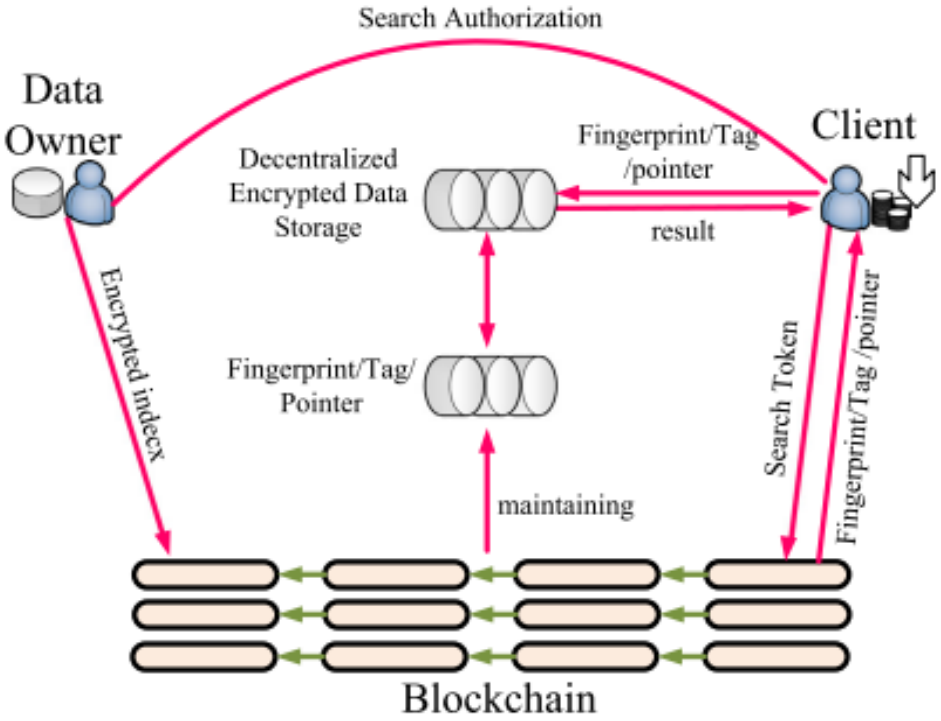

Fig. 5. System structure I for search in blockchain. [1]

Using Ethereum, a public blockchain, Hu et al. [121] offer a decentralized privacy-preserving search strategy to ensure that the data source does not need to complete verifications with the help of a smart contract in order to receive reliable search results. The authors also provide a unique fairness mechanism and develop a fair privacy-preserving search system, in which all users, whether in a single-user or multi-user environment, are treated equally and encouraged to make accurate calculations.

eISSN 1303-5150

However, due to the high cost of eth/etc. required to conduct the sophisticated search work on a smart contract, the aforementioned techniques have a limited scalability and significant overhead.

A decentralized privacy-preserving search technique using the private blockchain is also proposed by Hu et al. [123]. In the proposed system, users may choose from a pool of verified and reputable service providers. In contrast to [121], this method handles massive data sets and allows only verified users to join its private blockchain network. Since the private blockchain makes use of a more efficient consensus process, it is more productive than [121], despite the two using the same encrypted index structure. On the other hand, smart contracts incur additional overhead in terms of time spent finding, updating, and calculating for both public and private blockchains. In light of the fact that multiple queries may result in additional costs for the client, Jiang et al. [122] investigate a new cooperative search scheme in blockchain-based data markets, in which the pricing cost for each query is distributed between I owner, who is compensated for his data ownership, and (ii) miner, who is rewarded by query search, via a smart contract and gas system. With the restriction that the resultant integrated query is not much bigger than any of its original questions in terms of the number of keywords, the cost can be decreased by combining numerous searches into a group.

However, the suggested technique provides just a split of groups based on keywords, which are used as query criteria. As more sophisticated query requirements may increase computing, communication, and gas cost, it is increasingly important to think about other constraint constraints, such as spatial-temporal aspects paired with keywords. A dynamically efficient approach for doing encrypted keyword search on distributed storage is proposed by Cai et al. [133]. Each file and its index will remain in the same service peer under the proposed system, reducing the communication cost of a search query. In addition, the scheme uses the smart contract to store encrypted search logs (evidence) on the blockchain, and it creates a fair protocol to manage disputes and provide fair pay-outs. The suggested system forms an arbiter committee with the presumption that more than two-thirds of the service peers on the committee are trustworthy.

This is unrealistic since it ignores the possibility of coordinated hacking and fraud. In their research, Zhang et al. [132] examine a new method for enabling encrypted keyword searches in a blockchain-based distributed ledger with centralized data storage. The proposed scheme is built around I the Suppressed Merkle inverted index, which has a logarithmic maintenance cost but allows for efficient query processing and result verification, and (ii) the Chameleon inverted index, which uses the chameleon vector commitment to achieve a constant maintenance cost. More than that, the suggested indexes have authenticated query algorithms designed for them. This article includes a security proof and an analysis of the proposed scheme's performance on real-world Ethereum datasets. Experimental findings show that the suggested technique successfully balances high query performance with drastically reduced on-chain maintenance costs. Users must verify the accuracy and completeness of search results and should not blindly trust the system's service providers.

A search that doesn't rely on keywords... Using GPUs to speed up blockchain search is an idea proposed by Morishima et al. [124]. In order to make the most of the blockchain functionality, the authors provide an array-based Patricia tree structure optimized for GPUs. Since the array is built by append-only write queries in a

eISSN 1303-5150

blockchain, it may grow in size dramatically as the number of keys increases. This is because the array stores each vertex of the Patricia tree representing the keys in the order they were added. All the aforementioned research summarizes how search functions on permissioned/permissionless blockchains, where the majority of encrypted indexes or tags are maintained in blockchain and the actual data is obtained.

It is impractical to obtain all the data/block one at a time using the basic pointer structure due to the real-time necessity of a batch of data from a large-scale deployment of the IoT devices. Pointer's storage benefit can be used with index mechanism and likely cross chain retrieval techniques to improve the search efficiency of numerous associated blocks or data. Since most search works provide a time performance metric and a security analysis, it is clear that blockchain only needs to have a minimal amount of storage on hand in order to guarantee search performance. The linked paper takes into account the gas utilization and fairness in permissionless blockchain in further detail. Permissioned blockchains are able to query unlimited amounts of data since there is no gas cap.

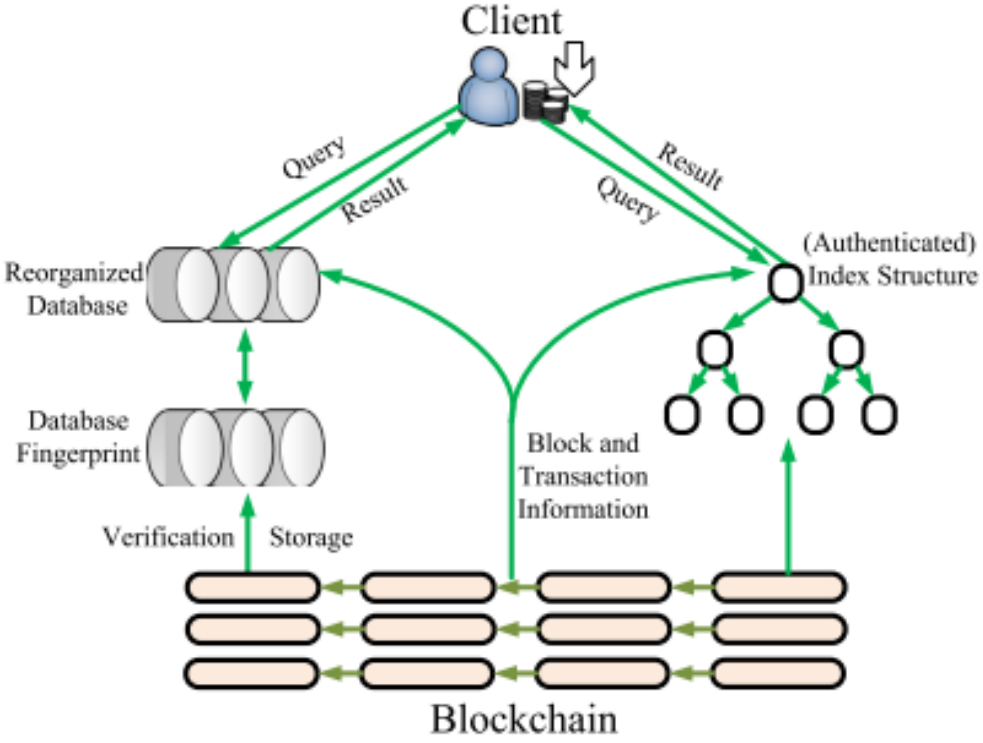

Fig. 6. System structure II for search in blockchain. [1]

5.2. Searching based on reorganized database structure

With the support of privacy protection and verification, search based on a rearranged database structure may efficiently offer a wide range of query services. The blockchain technology creates a public ledger by recording each user's transactions in chronological blocks. The blockchain's blocks and transactions are then retrieved by the middleware layer, which efficiently re-organizes the data in the database in order to present the client with a number of query options. To that end, Fig. 6 depicts the organizational framework of a search system.

A search that doesn't rely on keywords... Verifiable search services for blockchain-powered systems are proposed by Peng et al. [125], who also provide an efficient query layer. To facilitate various query services, such as block query and transaction inquiry, the middleware layer extracts the transactions from the underlying blockchain system and arranges them into databases.

eISSN 1303-5150

Each built database is given a cryptographic hash value to prevent middleware from saving the spoofed information. The miners first authenticate the database fingerprint, i.e., the hash value and some database attributes, and then record them in the blockchain. If you want to improve the speed of your searches without having to go through the trouble of querying each and every block in the blockchain, you may do so by building a new three-layer architecture for the blockchain's storage.

The drawback is that it creates update and calculation cost in the key database as new transactions and blocks will be continually created, and the databases require to be updated to handle various search services in a more effective and timely manner. Xie et al. [126] adopt a two-step private information retrieval to ensure that the necessary transactions from trust less full nodes may be obtained by lightweight clients without information leaking. Meanwhile, it considerably minimizes the communication overhead between the complete node and the lightweight client. To facilitate the lightweight client's access to the transaction and related simplified payment verification evidence, the authors reorganize the blockchain data into two types of relevant databases, namely, an index database and a database of transaction files. However, with the arrival of the new block, two databases need to be updated, resulting to greater update cost for merely searching recent transactions.

There is also a lack of thorough study and comparison of the suggested system's performance. In order to provide several search parameters, Pratama et al. [127] plan blockchain data retrieval's architecture. Data searching is simplified when numerous criteria are used. In addition, various analytical functions are provided to enhance the overall capabilities of the query layer system, which is used to conduct the actual queries. The three primary query functionalities presented in this paper are I retrieval query, which allows users to locate blockchain data according to multiple search parameters, (ii) aggregate query, which provides some elementary statistical analysis for a collection of blockchain data, and (iii) ranking query, which ranks blockchain data according to its blockchain component. However, the query layer technology involved in extracting and transforming blockchain data into database objects is just briefly presented and discussed, with no specifics provided on how to actually implement it. Additionally, there is a lack of comprehensive performance comparison and evaluation of the suggested system. A distributed database is characterized by its lack of a centralized data repository. It has fast query processing and a well-formed data type, but it can't guarantee the accuracy of the information.

Muzammal et al. [128] present ChainSQL, a new blockchain-based database solution that integrates the blockchain with distributed databases, to enhance safety. As a middleware, three use cases are implemented between the user application and the database: I a tamper-resistant multi-active database, (ii) a data-level disaster recovery backup, (iii) an audit middleware. The resulting search method combines the trustworthiness of blockchain technology with the speed of distributed database query processing. However, there hasn't been enough work put into testing the proposed system to draw any firm conclusions about how much it would cost to execute a smart contract or how much it will cost to verify the results of

Linoy et al. [136] suggest a search strategy that makes use of Hadoop, a scalable distributed processing and storage system, to deal with the growing volume of blockchain data. Hadoop's MapReduce processes download and synchronize data from the chain, and then store it in HDFS (Hadoop

eISSN 1303-5150

Distributed File System) using an in-memory B +Tree-based index. For obtaining blockchain blocks and transaction details, the search mechanism allows for queries written in SQL (Structured Query Language). In addition, it uses a secret information retrieval method to protect user privacy during the search. Even though it achieves excellent query efficiency in experiments, it is not compared to other state-of-the-art models to gauge its true performance. The benefits and drawbacks of the most recent search methods, which are based on the rearranged database structure, are outlined. Metadata descriptions of items and smart devices are predicted to rise at a more modest rate than the size of IoT networks, while the data created by smart devices is expected to increase exponentially.

Due to the potential high cost of both building and verifying such a blockchain, it is still difficult to conduct such a search efficiently. However, due to the time-sensitive nature of IoT data, ensuring data consistency between built databases and the underlying blockchain system can be challenging due to the need for frequent database re-construction, storage, and verification. In a scenario where smart devices are deployed on a massive scale, it would be impractical to constantly extract, reconstruct, and verify the real-time data being produced. In addition, the complexity of re-constructing and verifying databases is heightened by the many types of IoT data and the varying needs of individual users. The effectiveness of blockchain search systems may be improved by combining IoT data management and retrieval techniques with a restructured database strategy. In addition, we provide a summary of the relevant blockchain features, a summary of the measurement performance, a summary of the considered search requirements of existing surveys and the related solutions, and a summary of the information concerned by existing surveys. As we can see, most of the relevant research works are concerned with measuring time and throughput, or ensuring the authenticity, reliability, and consistency of a system. Since most search is turned into rearranged databases, blockchain's performance is not evaluated in the aforementioned polls. It is required to provide a more in-depth study of the costs associated with blockchain verification and other processes. Related search infrastructure opts for Ethereum, Bitcoin, and Ripple as the underlying blockchain because to the need for high throughput and fast confirmation of results.

5.3. Searching based on authenticated index structure

A blockchain-maintained authenticated index structure may be used to efficiently enable authenticated queries and guarantee the completeness and soundness of search results; this structure is used to organize the data in blockchain. During a query, a client first makes a request to a full node, which is responsible for validating all network transactions and maintains both the block header and block body. The complete node not only conducts the query using the authorized index, but also appends verification information to verification objects before returning both the results and the verification objects to the client. To that end, Fig. 6 depicts the organizational framework of a search system. What follows is a summary of how non-keyword search is currently being treated in studies.

A search that doesn't rely on keywords. The method provided by Zhu et al. [129] provides relational meaning to information stored in history blocks. This means that all blockchain data exists as multi-attribute relations among blocks. To speed up data access, three types of index mechanisms (i.e., Block-level B +-tree, Table-level Bitmap Index, and Layered index) are designed to answer the corresponding range query, such as I obtaining a block according to block id, transaction id, or timestamp, (ii) obtaining

eISSN 1303-5150

tuples that are with the same transaction type, and (iii) obtaining transactions given some conditions. In addition, to boost the effectiveness of both execution and verification, the authors build an authenticated index structure according to layered index and return verification objects based on it. The suggested search strategy benefits from using a relational semantics and stacked index to improve search performance.

The execution time and computation cost may grow exponentially with the size of the verification object if the request data span a large number of blocks, as the verification object incorporates the search results and the corresponding sibling hash values in the query path of each block that contains the search results into its total size. Currently, the vast majority of relevant studies are dedicated solely to providing basic query limitations (e.g., a single phrase search) by relying on symmetric encryption techniques that allow the data source to independently verify the authenticity of query results. Though several searchable symmetric encryption techniques already exist, it is not apparent how to make them verifiable such that they may handle complex data structures and expressive searches like Boolean queries and structured data without imposing an undue financial burden on the data source.

Boolean range inquiries and subscription questions are often used in blockchain applications, and Xu et al. [130] focus on these types of queries to address this issue. This work is the first of its kind that uses a built-in authenticated data structure and a returned verification object to verify the integrity of a blockchain database's queries. The suggested solution, however, does not address the gas-efficient issue of the authenticated data structure. To solve this problem, Zhang et al. [131] offer a gas-efficient Merkle Merge Tree, called a GEM2-tree, that can be effectively managed on the blockchain. With its efficient handling of authorized range requests, the smart contract's storage and calculation costs are drastically cut down. To verify results, we compute the root hash on the fly by suppressing internal nodes of the GEM2 -tree. Moreover, an enhanced GEM2 -tree is planned to lessen the burden of upkeep while keeping the query performance the same.

The multi-user setup, in which all authorized users can search the data provider's shared files, is not considered in the proposed method. Additional research on other authenticated inquiries, such as keyword and aggregate searches, and the development of gas-efficient data structures for these is crucial. Block directed acyclic networks are proposed by Qu et al. [96] as an alternative to sequential blockchain access for storing spatial-temporal data. A block space index is used throughout the query system to guarantee the uncorrupted body of the block at all times. Meanwhile, a Merkle Patricia Tree variation is used to track the worldwide position update. In particular, by traversing block headers, the presented approach may perform spatial-temporal queries directly on chains of directed acyclic networks.

The search system is able to handle basic spatial-temporal searches, but it is unable to carry out more complicated inquiries that involve several constraint requirements, such as a spatial-temporal query in addition to a keywords requirement. FalconDB is presented by Peng et al. [140], where a distributed blockchain network is used to keep servers and clients in sync. The client just saves the block headers, whereas the server stores the whole database with all of the authorized data structures. FalconDB stores the digests produced from authorized data structures and uses these for query/update authentication via the database servers, which clients can utilize as verification interfaces. Smart contracts are used to enforce the incentive model and offer penalties for dishonest activities, ensuring the

eISSN 1303-5150

service supplied by the server node. A smart contract's execution cost, however, is not quantified. FalconDB improves user collaboration with respect to efficiency, storage cost, and security, but it comes at a significant price in terms of verification and consensus cost. An improved and more secure blockchain search approach for IoT-related application domains is possible through the use of an efficient authenticated index to search IoT data transactions.

Due to the potential for significant verification costs and a sizable verification object, managing such a structure on a large-scale IoT network is no simple undertaking. Furthermore, the ever-expanding scale and dynamic nature of the IoT network, where the data is created at high volume, velocity, and kinds, makes it difficult to readily meet the complicated needs from diverse users. For the many varieties of real-time IoT data transaction, a new sort of storage structure and authentication contract is required. The goal is to reduce verification and communication costs while simultaneously increasing the accuracy and completeness of search results for many users.

5.4. Other researches

The search of payment systems in blockchain, the searching of data in semantic blockchain, the search utilizing sharding mechanism, and the search in other distributed systems will all be discussed, as will the search of electronic medical records paired with blockchain. Granular Access Authorization is a proposed architecture by Zhang et al. [138] for blockchain-powered electronic medical records. It can handle versatile queries that provide an authorization scheme and access model. Different degrees of permission granularity are possible within the access model. On the other side, it can keep working with the blockchain and even improve compatibility with it. However, there hasn't been enough time spent on performance testing the suggested design. It's important to remember that the blockchain infrastructure already contains many user-generated smart contracts. Up light of this, Tran et al. [137] introduce a new kind of search engine that is tied in with a smart contract on the blockchain. Users can utilize examples of comparable smart contracts that blockchain generates and saves to help them verify their own work.

There hasn't been much time spent on gauging the planned system's effectiveness, though. Integrating Semantic Web capabilities into blockchain architecture is the heart of the semantic blockchain concept. In order to address the current difficulties in cross-chain interoperability and resource sharing, the semantic blockchain proposes a semantic processing layer on top of the conventional blockchain, which use standard ontology to represent data (including blockchain metadata and content) [143]. At the moment, BLONDiE (blockchain ontology with dynamic extensibility) [144] and EthOn [145] are the most widely used blockchain ontologies (ethereum Ontology). Block header metadata and transaction data may be expressed as RDF (Resource Description Framework) triples using such ontologies, and SPARQL (SPARQL protocol and RDF query language) [146] can be used to access data stored in RDF format using a query language. A specialized SPARQL engine may be used to search for relevant data if the blockchain employs ontology and RDF to represent the block information and content. The block content is often stored in off-chain repositories like RDF databases (like RDF4J) or other NoSQL (Not Only SQL) databases, allowing semantic blockchains to lower the cost of on-chain storage by keeping only the metadata of block headers. This also makes it easier to use SPARQL and other native NoSQL query languages to search semantic blockchain data.

The sharding approach has the obvious benefit of increasing both the

eISSN 1303-5150

blockchain's scalability and the speed at which transactions may be processed. Participants in the food supply chain, validators, and a blockchain query manager make up the three tiers of the network proposed by Malik et al. [141]. Distribution network members categorize products by area. Transactions are processed in parallel by validators in each shard, which greatly enhances scalability. When a consumer submits a query request, the query manager quickly queries the system and verifies the results. The framework of the domain-based static sharding system developed by Yoo et al. [142] separates the shard into two parts: the local shard and the global shard. Through the use of smart contracts, node verification assures the atomicity and isolation of shard-level transactions and enables quick querying of both local and cross-shard transactions. To achieve simultaneous processing of transactions and effectively improve scalability, Li et al. [139] offer a network sharding strategy that I split the edge nodes into several shards to create multiple local blockchain networks, and (ii) creates a global blockchain network in the cloud. Customers' requests are first checked against the local blockchain and then then sent to the global blockchain if necessary. This has the potential to boost query efficiency while simultaneously decreasing the efficiency with which resources may be accessed.

Filecoin and IOTA are only two examples of other distributed ledgers that support search (Internet of Things Application). As an incentive layer for the Interplanetary File System (IPFS) ,Filecoin [134] can ensure the smooth execution of IPFS transactions at the institutional level. Users initiate a retrieval in Filecoin by locating retrieval-capable miners and requesting retrieval quotations; the system then sets up a payment channel between users and retrieval miners; and lastly, the miner transfers the retrieved data to the user. Distributed hash tables are the foundation technology for data retrieval by address by content (DHT). This not only guarantees that data integrity is verified, but it also eliminates the risk of a single point of failure in central file storage. Filecoin, on the other hand, might compromise the security of sensitive user information. The user will submit the original data to the storage miner immediately following a successful order matching.

If the user's data is not encrypted, the storage miner can access the whole database without raising suspicion on the network or among users. At the same time, bad miners may easily steal the user's information from this site. For cases involving a large number of simultaneous transactions involving devices connected to the Internet of Things, IOTA [135] is the distributed ledger of choice. In addition, IOTA does not charge a transaction management fee, which drastically lowers the overall transaction price for IoT micropayments and tiny purchases. Using the Application Programming Interface (API) supplied by the IRI software running on nodes, users may conduct queries and do other actions on the IOTA network (IOTA reference implementation). However, as there is no miner in IOTA to ensure the integrity of the ledger, there is very no safety in checking whether or not ledger transactions can be undone. Furthermore, anyone may transmit a large number of transactions at any moment with a transaction charge of $0, reducing the overall network's efficiency.

6. CHALLENGES AND FUTURE PROSPECT IN IOT NETWORKS

There are now various difficulties that reduce the query performance of blockchain search. To improve its usability and adaptability across various blockchain-based Internet of Things applications, it must overcome these obstacles and conform to specific standards. In this article, we'll go over the difficulties of blockchain-based search in

eISSN 1303-5150

the Internet of Things (IoT) space, then talk about where the field of study may go from here.

Searching Difficulties: Data sources, such as smart gadgets, provide a flood of diverse Internet of Things data at breakneck speeds. All of these obstacles make it harder for the blockchain search system to improve query speed. In light of this, it is imperative that it makes every effort to adapt to the difficulties and provide effective search solutions. The following are some of these difficulties: To improve search efficiency, a search engine should be able to retrieve diverse types of IoT data from different types of IoT-related applications.

For the sake of data storage and administration, a search engine should be adaptable enough to probe a scalable, distributed data model designed for IoT. For the encrypted index to be partitioned into many blocks and submitted to the smart contract with sufficient transactions, a trustworthy, resilient, and load-balanced storage structure should be provided. • Optimizing inquiries: In IoT domains, where smart devices and their services are constantly evolving, a search system's responses to a vast collection of queries should be timely, accurate, and multifaceted. Latency in obtaining dynamic IoT data may be minimized by minimizing the search system's computational, communication, verification, and execution overhead, all of which can add up to significant costs. • Data sharing: Based on the consensus mechanism, a blockchain network might be used as a shared database. Due to the dynamic nature of IoT data, it is essential that all involved parties maintain a consistent copy of the database and agree to every update before any observable results can be expected.

Possible Futures: In the preceding paragraphs, we discussed the major flaws of blockchain search. In this article, we provide a glimpse into the future of research and development in IoT domains, with the goal of discovering an effective blockchain search engine to fulfil the various search needs of IoT domains. • One promising avenue for further study is the combination of blockchain search with cutting-edge network and computer methods. Search speed, scalability, and security are all areas that might benefit from this combination.

In particular, there is promise for the integration of edge computing with blockchain systems to provide low-latency search and storage services; machine learning can achieve more accurate results in large and dynamic IoT datasets, and so may be able to help blockchain search optimize performance and realize artificial intelligence. However, for compute limited and resource constrained devices, a number of difficulties, including lengthy training times and high computation costs, need to be addressed and handled. • The second line of inquiry might focus on the need for keywords in IoT transaction retrieval in conjunction with more complicated query limitations like time and geography. Because of the exponential expansion of data in IoT-based systems, users have varying needs for transaction data for their various applications. Now is the time to build a blockchain search engine capable of handling a wide range of sophisticated query conditions.

Traditional search methods are not acceptable solutions due to blockchain's constraints in areas such as throughput, storage capacity, latency, size, and bandwidth. A deeper and more clear understanding of the search transaction in terms of the IoT properties, spatial-temporal property, relevance, access permission, et cetera, and the initiation of a scientific foundation for the search technique, result verification, and payment fairness are all areas that need to be explored in future research. • Maximizing usefulness across many user requests and developing a just payment system would be the third possible line of inquiry. Since IoT

eISSN 1303-5150

data is time-sensitive and no one wants to obtain the data with excessive delay, it is necessary to take into account all requests concurrently and optimize the advantages for all users.

Due to the fact that users may not be able to be trusted, it is important to guarantee I the data provider is compensated if users search the transactions, and (ii) users receive accurate search results that deliver their expected value if they pay the required fee. • The functions required by IoT applications may span many disciplines. Such a scenario calls for the co-creation of a search system across several domains. To quickly achieve linked information across different blockchains and to assure the soundness and integrity of the returned findings, future research may need to examine cross-chain retrieval mechanisms across numerous IoT domains and create associated verification algorithms.

• Information created in today's age of the Internet of Things (IoT) is both abundant and fast-moving, posing a problem for methods of boosting transaction throughput. To further improve throughput, future studies may also need to build a very efficient technique for retrieving blockchain data. The intra-shard and inter-shard consensus performance must also be optimized. • A SPARQL engine that operates on the blockchain network is required for the semantic blockchain in order to index the on-chain RDF data. An alternative is to construct the on-chain SPARQL engine using smart contracts. Smart contract data structures need to be built up to be more granular and complicated before this can be achieved.

7. CONCLUSIONS

Exploring blockchain search engines for recovering encrypted data in a fair and safe pattern is crucial as there is a growing need to do so in light of the development of potent upcoming blockchain approaches. We begin this research by introducing the search works pertinent to the IoT network. Then, the history of the blockchain and the basics of searching are covered. Here, we provide a problem analysis in IoT domains and an overview of the research conducted on blockchain search. In the meanwhile, we give a comparison of existing works that examines blockchain properties and search efficiency. We also discuss the obstacles and potential research paths for blockchain search in IoT sectors. These issues will be fixed, and the query optimization technique for the central blockchain search problem in an IoT network will be investigated in the future.

REFERENCES

[1]. Tang, J., Lu, X., Xiang, Y., Shi, C., & Gu, J. (2022). Blockchain search engine: Its current research status and future prospect in Internet of Things network. Future Generation Computer Systems. 138 (1) (2023) 120–141.

[2]. S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, https: //bitcoin.org/bitcoin.pdf.

[3]. M. El-Hindi, C. Binnig, A. Arasu, D. Kossmann, R. Ramamurthy, Blockchaindb: a shared database on blockchains, Proc. Very Large Data Bases Endow. 12 (11) (2019) 1597–1609.

[4]. Tara Salman, MaedeZolanvari, Aiman Erbad, Raj Jain, Mohammed Samaka, Security services using blockchains: A state of the art survey, IEEE Commun. Surv. Tutor. 21 (1) (2019) 858–880.

[5]. R. Yang, F.R. Yu, P. Si, Z. Yang, Y. Zhang, Integrated blockchain and edge computing systems: A survey, some research issues and challenges, IEEE Commun. Surv. Tutor. 21 (2) (2019) 1508–1532.

[6]. W. Yang, E. Aghasian, S. Garg, D. Herbert, L. Disiuta, B. Kang, A survey on blockchain-based internet service

eISSN 1303-5150

architecture: Requirements, challenges, trends and future, IEEE Access 7 (2019) 75845–75872.

[7]. M. Conoscenti, A. Vetro, J.C. De Martin, Blockchain for the Internet of Things: A systematic literature review, in: IEEE/ACS 13th International Conference of Computer Systems and Applications, 2016, pp. 1–6.

[8]. A. Ghasempour, J. Lou, Advanced metering infrastructure in smart grid: Requirements challenges architectures technologies and optimizations, in: Smart Grids: Emerging Technologies, Challenges and Future Directions, Nova Science Publishers, 2017, pp. 77–127.

[9]. Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, in: IEEE International Congress on Big Data (BigData Congress), 2017, pp. 557–564.

[10]. W. Wang, D. Niyato, P. Wang, A. Leshem, Decentralized caching for content delivery based on blockchain: A game theoretic perspective, in: IEEE International Conference on Communications, 2018, pp. 1–6.

[11]. T. Salman, M. Zolanvari, A. Erbad, R. Jain, M. Samaka, Security services using blockchains: A state of the art survey, IEEE Commun. Surv. Tutor. 21 (1) (2019) 858–880.

[12]. A. Ghasempour, Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges, Inventions 4 (1) (2019) 1–12.

[13]. R. Campos, G. Dias, A.M. Jorge, A. Jatowt, Survey of temporal information retrieval and related applications, ACM Comput. Surv. 47 (2) (2015) 1–41.

[14]. K. Abdulahhad, C. Berrut, J.-P. Chevallet, G. Pasi, Modeling information retrieval by formal logic, ACM Comput. Surv. 52 (1) (2019) 1–37.

[15]. S. Wu, D. Jiang, B.C. Ooi, K.-L. Wu, Efficient b-tree based indexing for cloud data processing, Proc. VLDB Endow. 3 (1–2) (2010) 1207–1218.

[16]. C.S. Jensen, D. Lin, B.C. Ooi, Query and update efficient B +-Tree based indexing of moving objects, in: Proceedings of the Thirtieth International Conference on Very Large Data Bases, 2004, pp. 768–779.

[17]. A. Guttman, R-trees: a dynamic index structure for spatial searching, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1984, pp. 47–57.

[18]. R.P. Mahapatra, P.S. Chakraborty, Comparative analysis of nearest neighbor query processing techniques, Procedia Comput. Sci. 57 (2015) 1289–1298.

[19]. R. Uddin, C. Ravishankar, V. Tsotras, Indexing moving object trajectories with hilbert curves, in: Proceedings of the 26th ACM International Conference on Advances in Geographic Information Systems, 2018, pp. 416–419.

[20]. B. Moon, H. Jagadish, C. Faloutsos, J.H. Saltz, Analysis of the clustering properties of the hilbert space-filling curve, IEEE Trans. Knowl. Data Eng. 13 (1) (2001) 124–141.

[21]. Z. Li, K.C. Lee, B. Zheng, W.-C. Lee, D. Lee, X. Wang, IR-tree: An efficient index for geographic document search, IEEE Trans. Knowl. Data Eng. 23 (4) (2011) 585–599.

[22]. F.M. Choudhury, J.S. Culpepper, Z. Bao, Batch processing of top-k spatial-textual queries, ACM Trans. Spatial

Algorithms Syst. 3 (4) (2018) 13:1–13:40.

[23]. M. Romero, S. Park, Decaying inverted quadtree: Index structure for supporting spatio-temporal-keyword query processing of microblog data, in: ACM Symposium on Applied Computing, 2017, pp. 951–956.

[24]. D. Zhang, K.-L. Tan, A.K.H. Tung, Scalable top-k spatial keyword search, in: International Conference on Extending Database Technology, 2013, pp. 359–370.

[25]. A.Khodaei, C. Shahabi, A. Khodaei, Temporal-textual retrieval: Time and keyword search in web documents, Int. J. Next-Gener. Comput. 3 (3) (2012) 288–312.

[26]. S. Nepomnyachiy, B. Gelley, W. Jiang, T. Minkus, What, where, and when: keyword search with spatio-temporal ranges, in: The Workshop on Geographic Information Retrieval, 2014, pp. 1–8.

[27]. P. Mehta, D. Skoutas, A. Voisard, Spatio-temporal keyword queries for moving objects, in: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2015, pp. 1–4.

[28]. T.-A. Hoang-Vu, H.T. Vo, J. Freire, A unified index for spatio-temporal keyword queries, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 135–144.

[29]. J. Zhao, Y. Gao, G. Chen, R. Chen, Towards efficient framework for timeaware spatial keywordqueries on road networks, ACM Trans. Inf. Syst. 36 (3) (2017) 24:1–24:48.

[30]. K.C. Lee, W. Lee, B. Zheng, Y. Tian, Road: A new spatial object search framework for road networks, IEEE Trans. Knowl. Data Eng. 24 (3) (2012) 547–560.

[31]. M. Qiao, L. Qin, H. Cheng, J.X. Yu, W. Tian, Top-k nearest keyword search on large graphs, Proc. VLDB Endow. 6 (10) (2013) 901–912.

[32]. R. Zhong, G. Li, K.L. Tan, L. Zhou, Z. Gong, G-tree: An efficient and scalable index for spatial search on road networks, IEEE Trans. Knowl. Data Eng. 27 (8) (2015) 2175–2189.

[33]. Z. Li, L. Chen, Y. Wang, G*-tree: An efficient spatial index on road networks, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), 2019, pp. 268–279.

[34]. Y. Gao, X. Qin, B. Zheng, G. Chen, Efficient reverse top-k boolean spatial keyword queries on road networks, IEEE Trans. Knowl. Data Eng. 27 (5) (2015) 1205–1218.

[35]. Y. Gao, J. Zhao, B. Zheng, G. Chen, Efficient collective spatial keyword query processing on road networks, IEEE Trans. Intell. Transp. Syst. 17 (2) (2016) 469–480.

[36]. Z. Chen, B. Yao, Z.-J. Wang, X. Gao, S. Shang, S. Ma, M. Guo, Flexible aggregate nearest neighbor queries and its keyword-aware variant on road networks, IEEE Trans. Knowl. Data Eng. 33 (12) (2020) 3701–3715.

[37]. F. Guo, Y. Yuan, G. Wang, L. Chen, Z. Wang, Cohesive group nearest neighbor queries over road-social networks, in: 2019 IEEE 35th International Conference on Data Engineering, 2019, pp. 434–445.

[38]. F. Guo, Y. Yuan, G. Wang, L. Chen, X. Lian, Z. Wang, Cohesive group nearest neighbor queries on road-social networks under multi-criteria, IEEE Trans. Knowl. Data Eng. 33 (11) (2020) 3520–3536.

[39]. Q. Li, Y. Zhu, J.X. Yu, Skyline cohesive group queries in large roadsocial networks, in: 2020 IEEE 36th International Conference on Data Engineering (ICDE), 2020, pp. 397–408.

[40]. J. Zhao, Y. Gao, G. Chen, R. Chen, Why-not questions on top-k geo-social keyword queries in road networks, in: IEEE International Conference on Data Engineering, 2018, pp. 965–976.

[41]. [41] S. Li, L.D. Xu, S. Zhao, The internet of things: a survey, Inf. Syst. Front. 17 (2) (2015) 243–259.

[42]. J. Pan, R. Jain, S. Paul, T. Vu, A. Saifullah, M. Sha, An Internet of Things framework for smart energy in buildings: Designs, prototype, and experiments, IEEE Internet Things J. 2 (6) (2015) 527–537.

[43]. S. Pattar, R. Buyya, K.R. Venugopal, S.S. Iyengar, L.M. Patnaik, Searching for the IoT resources: Fundamentals, requirements, comprehensive review and future directions, IEEE Commun. Surv. Tutor. 20 (3) (2018) 2101–2132.

[44]. E. Park, Y. Cho, J. Han, J.K. Sang, Comprehensive approaches to user acceptance of Internet of Things in a smart home environment, IEEE Internet Things J. 4 (6) (2017) 2342–2350.

[45]. F. Liang, C. Qian, W.G. Hatcher, W. Yu, Search engine for the Internet of Things: Lessons from web search, vision, and opportunities, IEEE Access 7 (2019) 104673–104691.

[46]. J. Gubbi, R. Buyya, S. Marusic, M. Palaniswamia, Internet of things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.

[47]. F. Zhang, M. Liu, Z. Zhou, W. Shen, An IoT-based online monitoring system for continuous steel casting, IEEE Internet Things J. 3 (6) (2016) 1355–1363.

[48]. H.S. Mohamed Amine Ferrag, Zineddine. Kouahla, M. Kurulay, Big IoT data indexing: Architecture, techniques and open research challenges, in: International Conference on Networking and Advanced Systems, 2019, pp. 1–6.

[49]. Y. Fathy, P. Barnaghi, R. Tafazolli, Large-scale indexing, discovery, and ranking for the Internet of Things (IoT), ACM Comput. Surv. 51 (2018) 39:1–39:67.

[50]. C.C. Tan, B. Sheng, H. Wang, Q. Li, Microsearch: A search engine for embedded devices used in pervasive computing, ACM Trans. Embedded Comput. Syst. 9 (4) (2010) 1–43.

[51]. M. Shah, A. Sardana, Searching in Internet of Things using VCS, in: International Conference on Security of Internet of Things, 2012, pp. 63–67.

[52]. H. Ma, W. Liu, Progressive search paradigm for Internet of Things, IEEE Multimedia 25 (1) (2018) 76–86.

[53]. Y. Zhou, S. De, W. Wei, K. Moessner, Search techniques for the web of things: A taxonomy and survey, Sensors 16 (5) (2016) 1–29.

[54]. K. Aberer, M. Hauswirth, A. Salehi, Infrastructure for data processing in large-scale interconnected sensor networks, in: International Conference on Mobile Data Management, 2007, pp. 198–205.

[55]. H. Wang, C.C. Tan, Q. Li, Snoogle: A search engine for pervasive environments, IEEE Trans. Parallel Distrib. Syst. 21 (8) (2010) 1188–1202.

[56]. K.-K. Yap, V. Srinivasan, M. Motani, MAX: human-centric search of the physical world, in: Proceedings of the 3rd international conference on

Embedded networked sensor systems, 2005, pp. 166–179.

[57]. W.I. Grosky, A. Kansal, S. Nath, J. Liu, F. Zhao, Senseweb: An infrastructure for shared sensing, IEEE Multimedia 14 (4) (2007) 8–13.

[58]. B. Ostermaier, K. Römer, F. Mattern, M. Fahrmair, W. Kellerer, A real-time search engine for the web of things, in: Internet of Things (IOT), 2010, pp. 1–8.

[59]. Z. Ding, Z. Chen, Q. Yang, IoT-svksearch: a real-time multimodal search engine mechanism for the internet of things, Int. J. Commun. Syst. 27 (6) (2014) 871–897.

[60]. J. Tang, Z. Zhou, L. Shu, G. Hancke, SMPKR: Search engine for Internet of Things, IEEE Access 7 (2019) 163615–163625.

[61]. D.C.M. Segura, R.D.S. Stabile, S.M. Bruschi, P.S.L.D. Souza, Providing computing services through mobile devices in a collaborative way - a fog computing case study, in: ACM International Conference on Modelling, 2017, pp. 117–121.

[62]. B. Alturki, S. Reiff-Marganiec, C. Perera, A hybrid approach for data analytics for internet of things, in: International Conference on the Internet of Things, 2017, pp. 1–8.

[63]. S. Fong, Big data mining algorithms for fog computing, in: International Conference on the Internet of Things, 2017, pp. 57–61.

[64]. E. Elmroth, P. Leitner, S. Schulte, S. Venugopal, Connecting fog and cloud computing, IEEE Cloud Comput. 4 (2) (2017) 22–25.

[65]. X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, G.-J. Ren, Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems, IEEE Wirel. Commun. 23 (5) 120–128.

[66]. A. Karamoozian, A. Senhaji Hafid, E.M. Aboulhamid, On the fog-cloud cooperation: How fog computing can address latency concerns of iot applications, in: Fourth International Conference on Fog and Mobile Edge Computing, 2019, pp. 166–172.

[67]. J. Tang, Z. Zhou, L. Wang, Answeringmultiattribute top-k queries in fog-supported wireless sensor networks leveraging priority assignment technology, IEEE Trans. Ind. Inf. 14 (10) (2018) 843–859.

[68]. J. Tang, Z. Zhou, X. Xue, G. Wang, Using collaborative edge-cloud cache for search in internet of things, IEEE Internet Things J. 7 (2) (2020) 922–936.

[69]. P. Zhang, Y. Liu, F. Wu, S. Liu, B. Tang, Low-overhead and high-precision prediction model for content-based sensor search in the Internet of Things, IEEE Commun. Lett. 20 (4) (2016) 720–723.

[70]. A.Shemshadi, Q.Z. Sheng, Y. Qin, ThingSeek: A crawler and search engine for the internet of things, in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, pp. 1149–1152.

[71]. Y. Hua, Y. Hua, K. Kyriakopoulos, Semi-edge: From edge caching to hierarchical caching in network fog, in: Theory and Practice for Integrated Cloud, Fog and Edge Computing Paradigms Workshop, 2018, pp. 3–8.

[72]. Y. Tao, P. Xu, H. Jin, Secure data sharing and search for cloud–edgecollaborative storage, IEEE Access 8 (2020) 15963–15972.

[73]. Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, H. Li, Lightweight fine-grained search over encrypted data in fog computing, IEEE Trans. Serv. Comput. 12 (5) 772–785.

[74]. J.-S. Fu, L. Yun, H.C. Chao, B.K. Bhargava, Z.-J. Zhang, Secure data storage and searching for industrial iot by integrating fog computing and cloud computing, IEEE Trans. Ind. Inform. 14 (10) 4519–4528.

[75]. W. Wang, P. Xu, D. Liu, L.T. Yang, Z. Yan, Lightweighted secure searching over public-key ciphertexts for edge-cloud-assisted industrial iot devices, IEEE Trans. Ind. Inf. 16 (6) (2019) 4221–4230.

[76]. M. Ali, S.U. Khan, A.V. Vasilakos, Security in cloud computing: Opportunities and challenges, Inform. Sci. 305 (2015) 357–383.

[77]. W. Sun, S. Yu, W. Lou, Y. Hou, H. Li, Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud, IEEE Parallel Distrib. Technol., Syst. Appl. 27 (4) (2016) 1187–1198.

[78]. R. Zhang, R. Xue, L. Liu, Searchable encryption for healthcare clouds: A survey, IEEE Trans. Serv. Comput. 11 (6) (2018) 978–996.

[79]. Z. Xiao, Y. Xiao, Security and privacy in cloud computing, IEEE Commun. Surv. Tutor. 15 (2) (2013) 843–859.

[80]. C. Bosch, P. Hartel, W. Jonker, A. Peter, A survey of provably secure searchable encryption, ACM Comput. Surv. 47 (2) (2014) 1–51.

[81]. L. Jin, W. Qian, W. Cong, C. Ning, K. Ren, Fuzzy keyword search over encrypted data in cloud computing, in: International Conference on Computer Communications, 2010, pp. 441–445.

[82]. J. Li, X. Lin, Y. Zhang, J. Han, KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage, IEEE Trans. Serv. Comput. 10 (5) (2017) 715–725.

[83]. Z. Wan, R.H. Deng, VPSearch: Achieving verifiability for privacypreserving multi-keyword search over encrypted cloud data, Trans. Dependable Secur. Comput. 15 (6) (2016) 1083–1095.

[84]. C. Cai, X. Yuan, W. Cong, Hardening distributed and encrypted keyword search via blockchain, in: IEEE Symposium on Privacy-Aware Computing, 2017, pp. 119–128.

[85]. W. Wang, P. Xu, D. Liu, L.T. Yang, Z. Yan, Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage, IEEE Trans. Ind. Inf. 16 (6) (2019) 4221–4230.

[86]. Y. Zhang, R.H. Deng, J. Shu, K. Yang, D. Zheng, TKSE: Trustworthy keyword search over encrypted data with two-side verifiability via blockchain, IEEE Access 6 (2018) 31077–31087.

[87]. Y. Yang, H. Lin, X. Liu, W. Guo, X. Zheng, Z. Liu, Blockchain based verifiable multi-keyword ranked search on encrypted cloud with fair payment, IEEE Access 7 (2019) 140818–140832.

[88]. L. Chen, W.K. Lee, C.-H. Chang, K.-K. Choo, N. Zhang, Blockchain based searchable encryption for electronic health record sharing, Future Gener. Comput. Syst. 95 (2019) 420–429.

[89]. J. Niu, X. Li, J. Gao, Y. Han, Blockchain-based anti-key-leakage key aggregation searchable encryption for IoT, IEEE Internet Things J. 7 (2) (2020) 1502–1518.

[90]. L.Z.X.D. Meng Shen, Yawen Deng, N. Guizani, Privacy-preserving image retrieval for medical IoT systems: A blockchain-based approach, IEEE Netw. 33 (5) (2019) 27–33.

[91]. A.-S. Kleinaki, P. Mytis-Gkometh, G. Drosatos, P.S. Efraimidis, E. Kaldoudi, A blockchain-based notarization

service for biomedical knowledge retrieval, Comput. Struct. Biotechnol. J. 16 (2018) 288–297.

[92]. A. Rahman, G. Loukas, S. Abdullah, A. Abdu, S. Rahman, E. Hassanain, Y. Arafa, Blockchain and IoT-based secure multimedia retrieval system for a massive crowd: Sharing economy perspective, in: ACM International Conference on Multimedia Retrieval, 2019, pp. 404–407.

[93]. S.Y. Nikouei, R. Xu, D. Nagothu, Y. Chen, A. Aved, E. Blasch, Real-time index authentication for event-oriented surveillance video query using blockchain, in: The 1st International Workshop on BLockchain Enabled Sustainable Smart Cities, 2018, pp. 1–8.

[94]. G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum Proj. Yellow Pap., 2014, pp. 1–32.

[95]. F. Li, M. Hadjieleftheriou, G. Kollios, L. Reyzin, Dynamic authenticated index structures for outsourced databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 2006, pp. 121–132.

[96]. Q. Qu, I. Nurgaliev, M. Muzammal, C.S. Jensen, J. Fan, On spatio-temporal blockchain query processing, Future Gener. Comput. Syst. 98 (2019) 208–218.

[97]. Eyal, A.E. Gencer, E.G. Sirer, R.V. Renesse, Bitcoin-ng: A scalable blockchain protocol, in: 13th USENIX Symposium on Networked Systems Design and Implementation, 2016, pp. 45–59.

[98]. G. Vizier, V. Gramoli, Comchain: Bridging the gap between public and consortium blockchains, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018, pp. 1469–1474.

[99]. R. Pass, E. Shi, Fruitchains: A fair blockchain, in: Proceedings of the ACM symposium on principles of distributed computing, 2017, pp. 315–324.

[100]. E.K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, B. Ford, Enhancing bitcoin security and performance with strong consistency via collective signing, in: 25th USENIX Security Symposium (USENIX Security 16), 2016, pp. 279–296.

[101]. J. Yu, D. Kozhaya, J. Decouchant, P. Esteves-Verissimo, Repucoin: Your reputation is your power, IEEE Trans. Comput. 68 (8) (2019) 1225–1237.

[102]. T. Hanke, M. Movahedi, D. Williams, Dfinity technology overview series, consensus system, 2018, arXiv:abs/1805.04548.

[103]. C. Dwork, Pricing via processing or combatting junk mail, in: Annual International Cryptology Conference, 1993, pp. 139–147.

[104]. E.B. Sasson, A. Chiesa, C. Garman, M. Green, M. Virza, Zerocash: Decentralized anonymous payments from bitcoin, in: 2014 IEEE Symposium on Security and Privacy (SP), 2014, pp. 459–474.

[105]. Y. Sompolinsky, A. Zohar, Secure high-rate transaction processing in bitcoin, in: International Conference on Financial Cryptography and Data Security, 2015, pp. 507–527.

[106]. S. King, S. Nadal, Ppcoin: Peer-to-peer crypto-currency with proof-ofstake, self-published paper, August 19 (1).

[107]. F. Schuh, D. Larimer, Bitshares 2.0: general overview, accessed June2017.[Online]. Available:

http://docs.bitshares.org/downloads/bitsharesgeneral.pdf.

[108]. M. Castro, B. Liskov, Practical byzantine fault tolerance and proactive recovery, ACM Trans. Comput. Syst. 20 (4) (2002) 398–461.

[109]. L. Lamport, Fast paxos, Distrib. Comput. 19 (2) (2006) 79–103.

[110]. D. Huang, X. Ma, S. Zhang, Performance analysis of the raft consensus algorithm for private blockchains, IEEE Trans. Syst. Man Cybern. 50 (1) (2019) 172–181.

[111]. C. Natoli, J. Yu, V. Gramoli, P.E. Veríssimo, Deconstructing blockchains: A comprehensive survey on consensus, membership and structure, 2019, arXiv:abs/1908.08316.

[112]. N. Szabo, Formalizing and securing relationships on public networks, First Monday 2 (9) (1997) 1–21.

[113]. K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things, IEEE Access 4 (2016) 2292–2303.

[114]. G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper, 2014, pp. 1–32.

[115]. C. Xu, J. Xu, H. Hu, M.H. Au, When query authentication meets finegrained access control: A zero-knowledge approach, in: ACM Conference on Management of Data, 2018, pp. 147–162.

[116]. H.H. Pang, K.L. Tan, Authenticating query results in edge computing, in: IEEE International Conference on Data Engineering, 2004, pp. 560–571.

[117]. R.C. Merkle, A certified digital signature, in: Advances in CryptologyCRYPTO, 1989, pp. 218–238. [118] X. Jian, L. Wei, Z. Yu, A. Wang, C.Z. Gao, Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures, J. Netw. Comput. Appl. 107 (2018) 113–124.

[118]. H.G. Do, W.K. Ng, Blockchain-based system for secure data storage with private keyword search, in: IEEE 13th World Congress on Services, 2017, pp. 90–93.

[119]. S. Tahir, M. Rajarajan, Privacy-preserving searchable encryption framework for permissioned blockchain networks, in: 2018 IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics, 2018, pp. 1628–1633.

[120]. S. Hu, C. Cai, Q. Wang, C. Wang, K. Ren, Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization, in: IEEE Conference on Computer Communications, 2018, pp. 792–800.

[121]. S. Jiang, Y. Duan, J. Wu, A client-biased cooperative search scheme in blockchain-based data markets, in: International Conference on Computer Communications and Networks, 2019, pp. 1–9.

[122]. S. Hu, C. Cai, Q. Wang, C. Wang, Z. Wang, D. Ye, Augmenting encrypted search: A decentralized service realization with enforced execution, IEEE Trans. Dependable Secure Comput. 18 (6) (2019) 2569–2581.

[123]. S. Morishima, H. Matsutani, Accelerating blockchain search of full nodes using GPUs, in: 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, 2018, pp. 244–248.

[124]. Z. Peng, H. Wu, B. Xiao, S. Guo, VQL: Providing query efficiency and data authenticity in blockchain systems, in:

IEEE 35th International Conference on Data Engineering Workshops, 2019, pp. 1–6.

[125]. [126] Y. Xie, C. Zhang, L. Wei, Y. Niu, F. Wang, Private transaction retrieval for lightweight bitcoin client, in: IEEE International Conference on Blockchain and Cryptocurrency, 2019, pp. 440–446.

[126]. F. Pratama, K. Mutijarsa, Query support for data processing and analysis on ethereum blockchain, in: International Symposium on Electronics and Smart Devices, 2018, pp. 1–5.

[127]. M. Muzammal, Q. Qu, B. Nasrulin, Renovating blockchain with distributed databases: An open source system, Future Gener. Comput. Syst. 90 (2019) 105–117.

[128]. Y. Zhu, Z. Zhang, C. Jin, A. Zhou, Y. Yan, SEBDB: Semantics empowered blockchain database, in: IEEE 35th International Conference on Data Engineering, 2019, pp. 1820–1831.

[129]. C. Xu, C. Zhang, J. Xu, vChain: Enabling verifiable boolean range queries over blockchain databases, in: ACM Conference on Management of Data, 2019, pp. 141–158.

[130]. C. Zhang, C. Xu, J. Xu, Y. Tang, B. Choi, GEM2 -Tree: A gas-efficient structure for authenticated range queries in blockchain, in: IEEE International Conference on Data Engineering, 2019, pp. 843–853.

[131]. C. Zhang, C. Xu, H. Wang, J. Xu, B. Choi, Authenticated keyword search in scalable hybrid-storage blockchains, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021, pp. 996–1007.

[132]. C. Cai, J. Weng, X. Yuan, C. Wang, Enabling reliable keyword search in encrypted decentralized storage with fairness, IEEE Trans. Dependable Secure Comput. 18 (1) (2018) 131–144.

[133]. Y. Psaras, D. Dias, The interplanetary file system and the filecoin network, in: 2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), 2020, p. 80.

[134]. W.F. Silvano, R. Marcelino, Iota tangle: A cryptocurrency to communicate internet-of-things data, Future Gener. Comput. Syst. 112 (2020) 307–319

[135]. S. Linoy, H. Mahdikhani, S. Ray, R. Lu, N. Stakhanova, A. Ghorbani, Scalable privacy-preserving query processing over ethereum blockchain, in: 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 398–404.

[136]. H. Tran, T. Menouer, P. Darmon, A. Doucoure, F. Binder, Smart contracts search engine in blockchain, in: International Conference on Future Networks and Distributed Systems, 2019, pp. 1–5.

[137]. X. Zhang, S. Poslad, Blockchain support for flexible queries with granular access control to electronic medical records (EMR), in: IEEE International Conference on Communications, 2018, pp. 1–6.

[138]. M. Li, Y. Qin, Scaling the blockchain-based access control framework for iot via sharding, in: ICC 2021-IEEE International Conference on Communications, 2021, pp. 1–6.

[139]. Y. Peng, M. Du, F. Li, R. Cheng, D. Song, Falcondb: Blockchain-based collaborative database, in: In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2020, pp. 637–652.

[140]. S. Malik, S.S. Kanhere, R. Jurdak, Productchain: Scalable blockchain framework to support provenance in supply chains, in: 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), 2018, pp. 1–10.

[141]. H. Yoo, J. Yim, S. Kim, The blockchain for domain based static sharding, in: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), 2018, pp. 1689–1692.

[142]. J. Cano-Benito, A. Cimmino, R. García-Castro, Towards blockchain and semantic web, in: International Conference on Business Information Systems, 2019, pp. 220–231.

[143]. U.-R. Hector, C.-L. Boris, Blondie: Blockchain ontology with dynamic extensibility, 2020, arXiv:abs/2008.09518.

[144]. J. Pfeffer, A. Beregszazi, S. Li, EthOn-an Ethereum ontology, 2021, https: //ethon.consensys.net/.

[145]. W3C, SPARQL query language for RDF, 2008, http://www.w3.org/TR/2008/ REC-rdf-sparql-query-20080115/.

[146]. J. Benet, IPFS - content addressed, versioned, P2P file system, 2014, arXiv:abs/1407.3561.