IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 October 2023, accepted 20 December 2023, date of publication 25 December 2023, date of current version 25 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3346432

RESEARCH ARTICLE

Blockchain-Based Logging to Defeat Malicious Insiders: The Case of Remote Health Monitoring Systems

HAMZA JAVED[1], ZAINAB ABAID[1], SHAHID AKBAR[2], KIFAYAT ULLAH[3], ASHFAQ AHMAD[4], AAMIR SAEED[5], HASHIM ALI[2], YAZEED YASIN GHADI[6], TAHANI JASER ALAHMADI[7], HEND KHALID ALKAHTANI[7], (Member, IEEE), AND ALI RAZA[4]

[1]Department of Computer Science, National University of Computer and Emerging Science, Islamabad 44000, Pakistan
[2]Department of Computer Science, Abdul Wali Khan University Mardan, Mardan, Khyber Pakhtunkhwa 23200, Pakistan
[3]Department of Electrical Engineering, Sarhad University of Science and Information Technology, Peshawar, Khyber Pakhtunkhwa 25000, Pakistan
[4]Department of Computer Science, MY University, Islamabad 44000, Pakistan
[5]Department of Computer Science and IT, University of Engineering and Technology, Peshawar 25000, Pakistan
[6]Department of Computer Science, Al Ain University, Abu Dhabi, United Arab Emirates
[7]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia

Corresponding author: Hend Khalid Alkahtani (Hkalqahtani@pnu.edu.sa)

This work was supported by the Princess Nourah bint Abdulrahman University Researchers Supporting Project, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, under Grant PNURSP2023TR140.

ABSTRACT IoT-based remote health monitoring is a promising technology to support patients who are unable to travel to medical facilities. Due to the sensitivity of health data, it is important to secure it against all possible threats. While a great deal of work has been done to secure IoT device-cloud communication and health records on the cloud, insider attacks remain a significant challenge. Malicious insiders may tamper, steal or change patients' health data, which results in a loss of patient trust in these systems. Audit logs in the cloud, which may point to illegal data access, may also be erased or forged by malicious insiders as they tend to have technical knowledge and privileged access to the system. Thus, in this work, we propose a Cloud Access Security Broker (CASB) model that (a) logs every action performed on user data and (b) secures those logs by placing them in a private blockchain that is viewable by the data owners (i.e., patients). Patients can query the blockchain, track their data's movement, and be alerted if their data has been accessed by an administrator or moved outside the cloud storage. In this work, we practically implement a web application that receives health data from patients, a CASB that securely stores the records in the cloud, and integrate a private blockchain that immediately logs all actions happening in the backend of the web application and CASB. We evaluate the system's security and performance under varying numbers of patients and actions.

INDEX TERMS Insider attack, private blockchain, cloud access security broker, remote health.

I. INTRODUCTION

Busy lifestyles make regular medical checkups difficult for many people, especially for chronic conditions like diabetes and hypertension. Some patients may be less mobile for medical reasons, such as the weak and elderly or those with motion sickness, light sensitivity, or social anxiety. In the recent Covid-19 pandemic, concern about contracting the virus or other illnesses has increased. Remote health monitoring utilizing smart IoT devices could help people unwilling or unable to visit the doctor regularly. Health monitoring IoT devices connect to a mobile app via Bluetooth to share patients' health data with doctors and receive medical suggestions. Such a system, depicted in Figure 1, allows remote medical consultations. Due to the sensitivity of health data and high-security requirements in this domain, a remote health monitoring system must secure user health data at all stages. It is important to ensure (CIA) confidentiality,

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

integrity, and availability of patient data [1]. If patient data is mismanaged or leaked, the lack of privacy will damage the system's reputation, reduce patient trust and hence leave it with few users [2]. All possible threats to patient data must be secured by a successful remote health monitoring system.

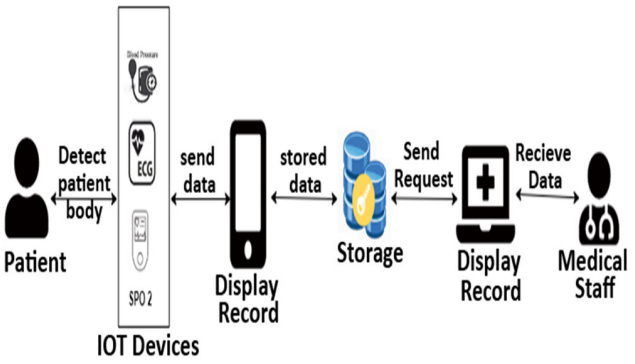


FIGURE 1. IoT-based health monitoring system.

A large amount of work has been done to secure various aspects of remote monitoring, such as authentication, access control, and secure storage. Notably, Cloud Access Security Broker (CASB) is a complete solution for securing cloud data, monitoring its movement and managing access policies. Several CASB products are available commercially, such as Bitglass CASB [3], Lookout CASB [4], CISCO cloudlock [5] and Microsoft Cloud App Security [6]. A CASB provides many security services, including malware detection, cloud configuration, single sign-on for authentication and identity management, user behavior analytics, encryption, key management, and access control [7], [8]. However, even with CASB deployment, insider attacks remain a key challenge. Insider attacks are known to cause significant data breaches. According to the report of *ObserveIT* in 2020, 60% of data breaches were caused by insider attacks [9]. According to a survey by Colombia University researchers, 50% of organizations suffered operational disruption because of insider attacks, 48% reported the loss of critical data and intellectual property, and 37% experienced damage to their brands [10]. These attacks may be perpetrated by a malicious administrator (e.g., disgruntled employees, spies, opportunists looking to expose/sell data for money) who has privileged system access and is familiar with the system policies. As a classical example, a medical device packaging business let go of an employee, Christopher Dobbins, in March 2020. After March, when receiving his last payment, he hacked the company's computer network, gained administrator access, and destroyed 120,000 documents, causing delays in medical equipment delivery [11]. Typical insider attacks in the eHealth domain are tampering, selling, or publishing patients' health data, such as a breach discovered by the Florida hospital where two hospital staff procured patient data sheets, including personal data such as phone numbers, names, and addresses. Two years of data were compromised and possibly used for false insurance claims [12].

The solution to the detection of insider attacks is continuously auditing system activities. For auditing, log data is used to store user actions with timestamps. However, tampering with log data itself is an issue. Malicious administrators with log access can modify log data to cover their tracks after illegally accessing patients' health data. To solve this problem, an immutable logging system is needed. Blockchains present a natural solution for immutability. The blockchain is an immutable, decentralized, and distributed ledger. It consists of several blocks which contain data representing transactions with timestamps. Each block contains the previous block's hash stored in the block's header. The first block in any blockchain is the Genesis block, which does not have any hash of the previous blockchain. Once data is stored in the blockchain, it can neither be updated nor removed. Because each block contains the previous block's hash, and if any record is updated or removed from the first block, the next connected block hash cannot be matched. If somehow, we match the hash of the first two blocks, and then we need to connect the second with the third block and also the third with the fourth and up to the last block of the blockchain by using the previous block hash matching scheme and this process is very difficult or impossible. Therefore, it is known as an immutable ledger [13], [14], [15]. In this paper, we present a blockchain-based logging system that will be integrated with our proposed CASB architecture to benefit from the security properties of CASB while ensuring visibility into attempted insider attacks. We will deploy a web application that receives patients' health data (from their mobile devices or medical staff, e.g., doctors or nurses) and passes it to the CASB to securely store in a public cloud service. Likewise, the web application also receives data retrieval requests and passes them to the CASB to process according to its access control policies, which in turn passes the data back to the web application. Each action of the web application backend and the CASB, whether it is related to data storage or retrieval, is logged immediately into a private blockchain. To be effective at detecting insider attacks, logging systems usually need to be monitored, i.e., someone needs to continuously read the logs and identify when illegal access has occurred. In our case, we would define illegal access as an action performed on user data that is not directly or indirectly initiated by the user. For example, if a doctor updates a patient's record and the patient approves it for upload into the system, the data will be encrypted, indexed, and stored in the cloud. All of these actions stem from the patient initiation of approving their data for upload. However, if the patient has not interacted with the server at all but their data is being downloaded, decrypted, or shared, then this defines a breach that is potentially an insider attack. Practically, manual analysis of logs is impossible given their large volume, and it is also difficult and potentially ineffective to define rules covering all possible legal and illegal access scenarios for automated analysis [16]. Therefore, we go with a simpler approach and simply give the data owners (i.e., patients) visibility into the private blockchain. Every data item gets assigned a tracking

ID, and the patient can simply use a blockchain explorer tool to view the logs that include the given ID. That way, the patient can see if their data has been accessed by someone not authorized by themselves.

While this approach does suffer from the drawback that users are not always tech-savvy or responsible enough to closely monitor their data, we argue that just the fact that logs are viewable by many users will serve to discourage insider attacks, as blockchain records are permanent. If an insider does succeed in a data breach, their actions can be immediately traced back by the data owner using the data item's tracking ID, or an external observer such as a regulatory body. As part of our future work, we will strengthen this approach by coupling the system with an automated intrusion detection system that inspects logs and reports anomalies to users, but for now, we believe the system is sufficiently effective to deter insider attacks. The major contributions of this paper are as follows:

- We propose a blockchain-based secure, transparent, and auditable logging mechanism for recording actions performed on sensitive data (such as health records) that can give immediate visibility into breaches caused by malicious insiders.
- We practically implement and analyze the feasibility (in terms of latency) of integrating a permission-based blockchain for continuous logging of data access events with our cloud access security broker.
- We present an analysis of the security properties of the proposed blockchain.

The remaining paper is organized as follows. Section II discusses the related literature to detect insider attacks. Section III outlines the foundational theories and technologies that underpin our proposed solution. Section IV describes our suggested methodology and architecture while Section V shows the implementation details/technologies for all modules of the system. Section VI presents the proposed system's results, including the security proof and the system performance relative to the amount of data and number of users. Section VII describes the complete security analysis of the proposed model. Section VIII discusses the comparison of the proposed methodology with the existing scheme. Section IX concludes the paper and Section X discusses open future directions.

II. LITERATURE REVIEW

A remote health monitoring system is an important step towards supporting reduced mobility patients' medical needs by enabling doctors to remotely monitor their vitals and provide advice. However, given the sensitivity of health data, the system needs high security to prevent tampering and unauthorized access due to insider attacks from potentially adversarial users and devices. Existing work has provided security schemes for various aspects of remote health monitoring or secure access to health data.

In 2008, Nakamoto, and Satoshi proposed Bitcoin as the first blockchain-based application which contains a peer-to-peer distributed ledger system for financial transactions [17]. The blockchain is one of the emerging technologies because of its immutability attribute, it is employed to secure the integrity of data [18]. Once data is stored in the blockchain, it can neither be removed nor updated. Due to the decentralized mechanism of blockchain, the data is stored in all the connected nodes; as a result, these nodes can protect the data from being lost. In addition, many consensus algorithms (proof of work, proof of stake, proof of idea, proof of authority, and many others) establish trust among all of the participant nodes when it comes to storing data and sharing it [19]. There is a wide variety of existing methods that can be used to identify an insider attack based on log data by applying blockchain technology. Kumar et al. proposed a blockchain-based healthcare system for patients to monitor and grant access to data. They recommend Blockchain-Based Privacy-Preserving and Robust Healthcare Data, which employs a public key to store encrypted data in the cloud and a private key to update it, to assure authorized access to patients' medical data. Meanwhile, the hash of data will be recorded in the permission blockchain [20]. Sahai et al. proposed the Verity framework, which is extensible with any SQL database. All insert, delete, and update operations will be performed in the database, and a fixed size of the salted cryptographic hash of each tuple of the database, known as a tuple, will be kept in the blockchain [21]. Cueva-Sanchez et al. [22] proposed keeping the record of the wood supply chain using cloud blockchain to ensure the transparency and integrity of forestry data and also reduce illegal logging and prevent internal or external tempering of registration, trees, and authorized certificate records. Zieglmeier et al. [23] proposed a pseudonym provisioning system also named P3 for the secure usage of log data while the metadata will be stored in blockchain and pseudonym data with related identifiers will be stored separately. Moreover, the anomaly data are modified data and impossible to re-identify. Adlam et al. [24] proposed a Hyperledger fabric based on zero-knowledge proof blockchain infrastructure to store the audit log of Electronic Health Records. The purpose of this solution is too safe the audit log data from tampering by different criminal attacks. Ma et al. [25] proposed two methods which are baseline and enhanced methods for efficiently storing log data and overcoming the indexing problem of an immutable Blockchain system. All [22], [23], [24], [26] techniques are good for reducing the storage cost of log data and also increasing the system speed but on the other hand due to the usages of off-chain in [22] and [23] and multichain in [24] and [25] they compromise the immutability, Decentralization, Data availability and also drop the transparency of the system. By using the off-chain, the data will be stored in any local database or other cloud-based centralized system but the hash of the given data will be stored in the block chain [27]. Similarly, in the multichain concept, there is no smart contract will be deployed and each node communicate with the other by using a digital signature while the smart contract is one of the important assets in

the blockchain to achieve the transparency and autonomy of data [28].

Now to overcome the immutability, Data availability, and transparency problems of the system. Ivan Chistiakov et al. [29] proposed a Directed Acyclic Hypergraph (DAG) which is based on a token therefore automatically timestamped and secures multiple-component data using logging. Rakib et al. [30] proposed a blockchain-based framework to store log data and also query and audit system the given data. They also ensure that the given system is transparent, has data accountability, and data confidentiality. Zhao et al. [31] proposed a method for drastically addressing the throughput of blockchain using hierarchical processing and logging of large amounts of data. Hsu et al. [32] proposed a block chain-based autonomous log management model and also access control for securing IoT from cyber security. But all these [30], [31], [32] methods are based on Bitcoin which contains a proof of work consensus algorithm. Within the proof-of-work consensus mechanism, the miner will be selected by successfully solving a problem, and once chosen, it will be able to carry out transactions and store data within the blockchain. The miner transaction can be validated by any other connected node in the network. However, selecting each node as a miner requires a significant level of computation power to solve the puzzle, which also takes an excessively long amount of time [33].

In a similar vein, to reconcile high computational power and rapid data storage many researchers proposed state-of-the-art solutions, Ahmed et al. [34] proposed a blockchain-based logs management to ensure log audit security named BCALS. The main purpose of this framework is to develop a secure and immutable audit log security that the admin cannot modify it. The proposed system contains Auditability, Immutability, Decentralization, and Analysis Support But they can use a Proof-of-concept consensus algorithm which is mostly employed for testing or feasibility analysis of any software, so it is not applicable for security purposes [35]. Mendon et al. [36] proposed an Audit chain to assure log integrity using proof of existence (POE). On the other way, The Proof of existence can be obtained by taking the hash of a file or document, which can then be stored in the blockchain using a timestamp just like the off-chain [22] concept, while the actual data of every document or file is available in either a local database or a cloud-based system [37]. So, by using POE we can compromise the immutability, Transparency, and Data availability. Jadidi et al. [38] proposed a framework to detect the anomaly of industrial control systems. The given system contains two stages. The first step is the collection of logs in a secure and distributed manner using blockchain while the second step detect the anomaly of blockchain. But for storing data in blockchain they use byzantine fault tolerance (BFT) with proof of capacity consensus algorithm while BFT is vulnerable to Sybil attack and it also has scalability issues because each node requires communication at every step of the process. Moreover, due to the usage of the capacity algorithm, the proposed system will become more costly to produce higher-capacity hard drives, and higher-capacity malware may affect the system [39]. With the help of a private blockchain, Yenugunti et al. suggested a collaborative intrusion detection system. Using a trust score consensus technique with a trust score range of 1-100, this system is designed to identify malevolent insiders. Although this method is beneficial in that only one node performs transactions, it still requires all other nodes to verify them. After the transaction has been verified, the node will be given a trust score; if the score is high enough, the node will be considered trustworthy or wise transaction will be discarded [40]. However, issues arise when some trustworthy nodes carry out a successful transaction while the rest of the network has not been validated trustworthy node as a potential threat node. Klinkmuller et al. [41] proposed an Ethereum Logging Framework (ELF) with a cost-efficient smart contract and also extracting log data into a common format. Both [34], [41] store data in text file which is in JSON format while the file is available in the cloud therefore log data file tampering is possible by any insider malicious user. Moreover, the text files which just contain plain text, are often thought to be malicious due to the nature of their content. Tuan et al. [42] Proposed a blockchain logs management system for smart gird with ciphertext policy Attribute-based encryption to establish fine-grained access control. While introducing blockchain novel signature chain for efficient logs protection. Lu et al. [43] proposed a shadow chain for storing log data in a decentralized manner for auditing data sharing between different parties. The system can solve internal leaks, facilitate data sharing among distrusted institutions, and even change. To overcome the problem of data reading in blockchain they will use a homomorphic encryption scheme. Moreover, hybrid storage in introduced to store only the hash of log data in the blockchain of the shadow chain. Both [42] and [43] use encryption in log data, as we know that log data does not contain any personally identifiable information about users, but it does contain very little information that can be used to track malicious users, therefore, perform encryption log data and allows only authorized user to view log make system more expensive and time-consuming process.

The analysis of the existing work shows in Table 1, that myriad techniques and technologies have been proposed to address the insider attacks. According to the first section, various solutions were proposed to detect insider attackers by applying different use cases like supply chain, wood chain, general cloud system, and health record to continuously monitor the system activity and also store data into a different type of blockchain like Hyperledger or Ethereum but we see that many techniques [23], [24], [25] not solve the immutability problem completely instead of that they only store the hash of give data using off-chain approach but on another hand [29], [30], [31] to solve the problem of immutability but their proposed solution is based on bitcoin, therefore, they

TABLE 1. Comparative analysis of existing schemes that use blockchain.

Methods	Immutability	Transparency	Data Availability	Integrity	Autonomy
Cueva-Sanchez et al. [22]	×	×	×	×	×
Adlam et al. [24]	×	×	×	×	×
Rakib et al. [30]	✓	✓	✓	✓	✓
Lu et al. [43]	✓	✓	✓	×	✓
Mendonc et al. [36]	×	×	×	×	×
Madhwal et al. [29]	✓	✓	✓	✓	✓
Jadidi et al. [38]	✓	✓	✓	×	×
Chien-Lung et al. [32]	✓	✓	✓	✓	✓
Valentin et al. [23]	×	×	×	×	×

increase computational power, storage cost as well as may be malicious attacks like in [34], [36], [38], and [41] DDOS or 51% attacks are also possible. Moreover, researchers also provide the solution to solve the above problem but these have many drawbacks in the sense of complex infrastructure, storage costs including gas fees, poor read-write data. as well as the vulnerability of storing log data in text files. So, we will require a solution that will monitor all the activity held in the health monitor system and have no single point of failure, scalability, non-repudiation, integrity, data privacy and confidentiality, more reliable and immutable that nobody can tamper, steal, or manipulate the log data. The required solution also can protect data from different cyber-security attacks like DDOS attacks and 51% attacks, as discussed in the security analysis section.

III. BACKGROUND

In this section, we describe the necessary background technologies and related concepts that form the building blocks of our proposed solution. We cover two technologies here, cloud access security broker and block chain.

A. BLOCKCHAIN

Blockchain is a decentralized and distributed ledger that may record any transaction or data in its blocks. The most well-known application of blockchain technology is Bitcoin, which was proposed by Satoshi Nakamoto in 2008 [17]. The application of a cryptographic hash function ensures that each block in a blockchain is related to the block that came before it. It uses the cryptographic technique of hash chaining to ensure the integrity and authenticity of data. It involves taking a block, hashing it with a cryptographic hash function, and then appending the resulting hash value to the next block. In the same manner, if we wish to add a new block, it will connect at the end of the current blockchain rather than the center or the beginning of the chain. The sp: genesis block is the first block in a blockchain and it does not include a hash value from the previous block. However, the header of each subsequent block does include a large amount of information, including the block number, the hash of the previous block, the nonce value, the total number of transactions, and the hash of all of the transactions contained in the block [44]. Blocks in the Bitcoin network are verified through a process called mining, which involves solving a complex mathematical problem and validating the block's transactions and header against the network's consensus rules. Once the data has been saved in the blockchain, it cannot be altered or deleted [15]. the fundamental blockchain structure of the proposed model is depicted in Figure 2.

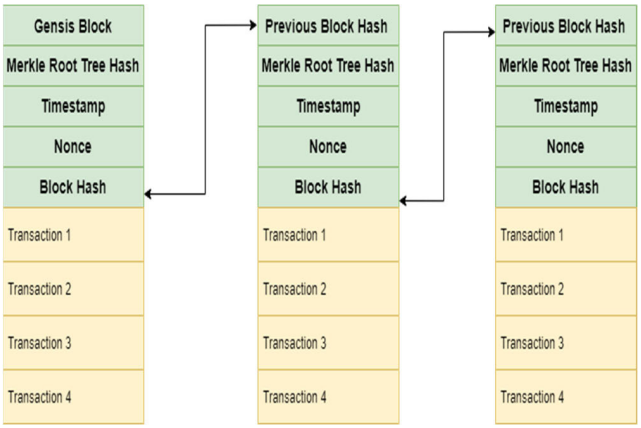


FIGURE 2. The basic structure of blockchain.

1) FEATURE OF BLOCKCHAIN

The blockchain contains many features that have contributed to its widespread adoption. The detail of some important features is given below:

- Immutability: Since the blockchain's transactions are based on a hash function, which is irreversible, any attempt to modify or delete data once it has been added to the ledger would necessitate updating the entire blockchain hash. Moreover, each node in a distributed network operates its blockchain that communicates with the networks of other nodes in the network through a decentralized protocol. Therefore, the record cannot be altered from the local blockchains of any other nodes [13].
- Decentralized Consensus: All transactions on a blockchain are validated by a network of users, rather than a single authority. This consensus mechanism helps to prevent fraud and ensures the integrity of the network [14].
- Transparency: Blockchain offers transparency by providing a public ledger of all transactions that have occurred on the network. This ledger is a decentralized

and distributed database that is maintained by a network of nodes, each of which has a copy of the ledger. Anyone can access the blockchain and view all transactions that have occurred, including the amount transferred, the time and date of the transaction, and the addresses of the sender and recipient. This transparency enables individuals and organizations to verify the authenticity and accuracy of transactions, and it also makes it more difficult for bad actors to manipulate or corrupt the data [15].

Availability: Blockchain technology ensures availability by being a decentralized and distributed system that is maintained by a network of nodes. Each node in the network has a copy of the blockchain ledger, and if one node goes offline or experiences a failure, the other nodes in the network can still maintain the availability of the ledger and continue to process transactions. In addition, blockchain technology uses a consensus mechanism to ensure that all nodes in the network have the same copy of the ledger. This ensures a single source of truth is available to all nodes on the network [44].

2) TYPE OF BLOCKCHAIN

The three most common forms of blockchains are Public, Private, and Consortium. The following characteristics of each blockchain are consistent across all of them. That is, in a P2P network, each node maintains its ledger and may access the blockchain data maintained by all other nodes. In [45] a public blockchain, every node has access to all of the data in the blockchain and can participate in transactions. It is immutable but anonymous where nobody can get the personal information of any user. Bitcoin is the best-known public blockchain. Public blockchains may incur high processing and storage overhead. It is not a secure place to save vital or private information. Private blockchains [46], often called permission-based blockchains, are only accessible to those who have been granted access. With this blockchain configuration, only authorized nodes can conduct transactions. Since private blockchains see a smaller volume of transactions, they often have a lower data storage requirement and a faster transaction time. Enterprises and other types of organizations make extensive use of private blockchains. In addition, it is useful for achieving information that is both private and vital [44].

3) CONSENSUS ALGORITHM

In a blockchain network, each participant uses a procedure known as a consensus algorithm to reach an agreement with everyone else in the network and is also able to ensure trust in unidentified nodes in the distributed computing system. The blockchain consensus procedure is made up of a few particular goals, such as reaching a consensus, working together, giving equal rights to every node, and requiring each node to take part in the consensus process. Consensus algorithms are a crucial component of blockchain technology, as they enable all participants in a distributed network to agree on the current state of the ledger. In simple terms, a consensus algorithm is a set of rules that determine how nodes in a network reach an agreement on the validity of a transaction and add it to the blockchain. Several consensus algorithms play a crucial role in blockchain technology as they ensure that all nodes in a network agree on the state of the blockchain. Some of the commonly used consensus algorithms in blockchain include Proof of Work (PoW), Proof of Stake (PoS), Proof of Authority (PoA), Delegated Proof of Stake (DPoS), and Practical Byzantine Fault Tolerance (PBFT), among others. Proof of Authority (PoA) is a consensus algorithm that relies on a group of approved validators to verify transactions and create new blocks on the blockchain. Unlike PoW and PoS, where participants compete to solve a cryptographic puzzle or stake their tokens to become validators, PoA requires validators to be approved by a central authority or a consortium of entities. PoA is known for its speed and energy efficiency, as it doesn't require large amounts of computational power or energy consumption to validate transactions [47].

4) SMART CONTRACTS

Smart contracts are typically written in a programming language that is specifically designed for the blockchain, such as Solidity for the Ethereum blockchain. The code of a smart contract is stored on the blockchain, which makes it transparent and immutable. This means that once a smart contract has been deployed, it cannot be changed or modified, and its execution is guaranteed to follow the rules that were defined in the code [41]. Smart contracts are executed automatically to meet certain conditions. These conditions are typically based on data that is stored on the blockchain, such as the balance of an account, the timestamp of a transaction, or the outcome of a previous contract execution. One of the key benefits of smart contracts is that they can automate complex processes and transactions, which can save time, reduce costs, and improve efficiency. For example, in the context of decentralized networks, smart contracts can be used to automate lending, borrowing, and trading, which eliminates the need for intermediaries and reduces the risk of fraud. Smart contracts can also help to increase transparency and accountability in contractual relationships, as all parties can see and verify the terms of the contract and its execution on the blockchain [48]. Overall, smart contracts have the potential to revolutionize the way that contracts are executed and enforced, offering many benefits in terms of efficiency, security, and transparency.

5) TYPE OF BLOCKCHAIN

The process of performing the transaction in the blockchain is storing any digital record and validating whether the store record is correct or not. To perform any transaction in the blockchain depends on the blockchain type or consensus algorithm as discussed above like in public blockchain or bitcoin any connected node can perform the transaction which is also known as a miner. The miner will be selected based

on the POW consensus algorithm by solving a puzzle. When the miner performs the transaction, it will be validated by all other nodes. If 51% of nodes validate the transaction then it is stored in the blockchain but if not validate the transaction, then it will be discarded and rewards will be given to the miner [17]. Moreover, in a private blockchain, there are a fixed number of validators selected by the specific organization that is responsible for storing data in the blockchain and also checking whether data is correct or not [49]. Overall, the transaction life cycle in the Bitcoin network involves the following steps: creation, broadcasting, verification, inclusion in a block, confirmation, and finality of transactions.

Creation A transaction is created when a user initiates a transfer of digital assets from one account to another. The user must provide details such as the recipient's public address, and the amount of assets to be transferred. Broadcasting: Once a transaction is created, it needs to be broadcast to the network so that it can be validated and added to the blockchain. Broadcasting is typically done using a peer-to-peer network, and once the transaction is broadcasted, it is available to all nodes in the network. ***Verification*** The next step is verification, where nodes on the network validate the transaction to ensure it is valid and doesn't violate any rules or double-spend the same coin. The validation process is carried out by specialized nodes called validators, who ensure that the sender has sufficient funds to complete the transaction and that the transaction hasn't already been spent. ***Inclusion in a block*** Once the transaction has been validated, it is added to a block of transactions. A block is a collection of validated transactions that are bundled together and added to the blockchain in sequential order. Each block contains a reference to the previous block, which creates a chain of blocks that make up the blockchain. ***Confirmation*** After a block containing the transaction has been added to the blockchain, it needs to be confirmed. This involves waiting for a certain number of subsequent blocks to be added to the blockchain, creating a level of security against fraud, double spending, and other attacks. The number of blocks needed to confirm a transaction varies depending on the specific blockchain, with some requiring just one confirmation, while others may require several. Finality: The final step in the transaction lifecycle is finality. Once a transaction has been confirmed by the required number of blocks, it becomes irreversible and is considered final. At this point, the recipient can safely assume that the transaction is complete, and the cryptocurrency has been transferred to their address.

B. CLOUD ACCESS SECURITY BROKER (CASB)

Cloud Access Security Brokers, also known as CASBs, are points of providing security that are located between customers and service providers. As the use of SaaS (Software as a Service) grows in businesses, cloud access security brokers become an increasingly important tool for information security experts. Additionally, they could have authority within the organization that who has access to what resources. Also, Encryption, logging, single sign-on, authentication, alerting, tokenization, anti-malware software, credential mapping, and intrusion prevention systems are all examples of security checks that may be provided by CASB [50]. The basic architecture of CASB is provided in Figure 3.

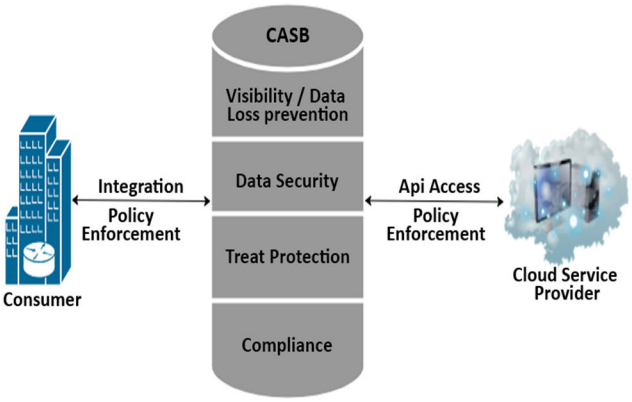


FIGURE 3. The basic structure of cloud access security broker.

1) CASB'S FOUR MAIN PILLARS

In CASB, there are several brokers and each broker contains these four main pillars that ensure CASB in one of the efficient security mechanisms. All these important pillars are discussed as given below:

- CASB for visibility: A CASB visibility strategy provides full transparency for the use of cloud apps, including those that have been sanctioned as well as those that have not been sanctioned, which are referred to as shadow IT. The cloud detection analysis delivers a risk evaluation for each cloud service that is being utilized. This offers business security experts the ability to decide whether to continue allowing the app to be ccessed or whether to prohibit it. This information is also helpful to build more grained controls, such as providing varied amounts of access to data based and application on an individual's technology/device, location, and field of activity. Another purpose for this information is that it is useful in assisting with finer controls [50], [51].
- Data Security/Loss Prevention Using CASB: A great number of businesses have already begun moving their information technology resources outside of their in-house data centers and onto numerous cloud environments, as well as the extensive catalog of online apps provided by SaaS (Software as a service) vendors. The organization staff already exchanging valuable information via platforms such as Amazon S3 [52], Microsoft 365 [53], salesforce [54], and others. Businesses must link their existing data loss prevention (DLP) technology with a cloud access security broker (CASB) solution to gain knowledge of critical information moving between on-premises and cloud environments. Because of this, companies can monitor the individuals who have access

to important data across their whole network without sacrificing security. Applying security protocols such as protecting information rights, access control, encryption, tokenization, and prevention of data loss, are some examples of measures that might be taken to reduce the amount of organizational data that is lost or stolen [50], [55], [56].

- Threat Protection with CASB: The Cloud Access Security Broker (CASB) helps businesses improve their knowledge of the data stored in the cloud by providing a variety of capabilities for detection, monitoring, and prevention. The CASB can apply machine-learning techniques to rapidly recognize suspicious behavior if a user tries to hack the system to get unauthorized access to information or makes an attempt to do so. To stop and prevent malware assaults, a broad variety of technologies and techniques are deployed. Some examples of these technologies and methods include dynamic and static malware analysis, adaptive access control, and threat intelligence. For instance, the CASB will trigger an alert if a developer attempts to access customer data within an application that is used for sales because only salespeople should be authorized to view that data. In the first scenario, the CASB does not restrict access to the client's information and does not notify an administration. However, in the second scenario, it does restrict access to the client's information and it alerts an administration [55], [56].
- Compliance using CASB: A CASB equipped with effective data privacy safeguards that are dispersed across various applications can help with this. By providing features for policy awareness and data classification, CASBs also help ensure compliance with data residency regulations like HIPAA, GDPR, and regulatory standards like ISO, PCI-DSS, and others [51], [55].

IV. PROPOSED SOLUTION DESIGN

A. THREAT MODEL AND DESIGN GOALS

1) THREAT MODEL

As we see in section I, there are a large number of IoT devices and users connected to a remote health monitoring system to send and receive vital data via the network. To protect sensitive patient information, a CASB is a good framework that is responsible for key management, access control, authentication, encryption/decryption, as well as monitoring user activity with a log management system. Administrators are key users of the remote health monitoring system who are most familiar with system policy as well as have the highest level of access. Hence, insider attacks can originate from malicious administrators who can tamper or illegally obtain or distribute patient records and delete the log generated against the action performed. Thus, in a log management system, one of the most important considerations is how audit logs will be protected from malicious actors. The protection of logs from modification after a system has been compromised is referred to as a log's "forward security". After a successful attack, a malicious administrator may engage in multiple types of attacks. An insertion attack occurs when fake log entries are generated, which can either overload or interrupt the log management system, resulting in a denial of service (DOS). Modification or reorder attacks may alter logs in such a way that their integrity is compromised. A withhold attack may launch a delayed attack to disrupt the log management system.

2) DESIGN GOALS

Malicious administrators (or generally, high privileged users) are the main threat considered in our proposed remote health monitoring system. Thus, our proposed model is designed to meet the following security goals.

- The immutability of the proposed system is a primary requirement, and it must ensure that once log data has been stored in cloud storage, it cannot be removed, updated, or altered in any way.
- Decentralization is essential in that audit log data is stored in multiple, independently accessible locations to provide a fail-safe against the possibility of a single point of failure.
- Our model must ensure non-repudiation by employing digital signatures that are based on asymmetric key encryption.
- The integrity of the log data should be ensured through the utilization of a blockchain hashing algorithm such as SHA-256.
- The audibility of data should be ensured. Each node should be able to perform an audit, i.e., a check on log data, before storing it on a private blockchain.
- Scalability is a critical factor given the large number of devices and users in a remote health monitoring system, especially when time-consuming blockchain operations are involved. As we only integrate a private blockchain that has a pre-defined fixed number of nodes, we do not envision that this will be an issue in the implementation.
- Easy tracking of data by end users is important as our security model relies on data owners themselves being able to determine if a breach has occurred. Any user should have visibility of their data, i.e., they should be able to view log data by employing a tracking-ID.

B. DESIGN AND ARCHITECTURE

In our proposed system, we assume that a patient uses IoT-based wearable devices that continuously measure various health metrics and send data to the cloud. As the health record is very sensitive, it is, therefore, crucial to secure access to the web-based application used by medical practitioners for viewing patient data. Our system implements a CASB to enforce strict access control. Moreover, a blockchain-based log management system is integrated with the CASB to continuously store audit log data, which contains

information about each action that was done on a patient's data, indexed by a tracking ID. Logged information includes the IP address and other identifying information of the user who accessed the data item and the time when the user acted. Our proposed architecture is shown in Figure 4. We now describe each of its components, users, and the overall workflow below:

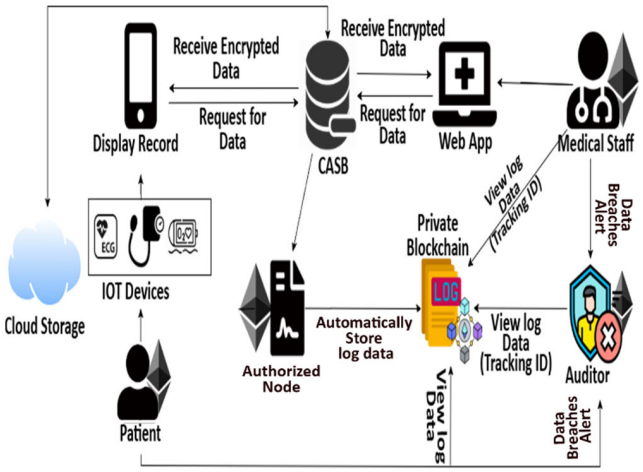


FIGURE 4. Proposed architecture.

C. END USERS

The end users of the system are either patients or medical professionals who view patient data. Both entities interact with the Cloud Access Security Broker (CASB) for storage and retrieval of data through end-user devices shown in the figure.

1) PATIENTS

The patient-CASB communication workflow is shown in Figure 4. Patients' end-user devices will be smartphones, installed with an application that receives data from wearable devices and aggregates it in the form of daily, weekly, or asynchronous (urgent) updates. The application also encrypts aggregated data with the patient's key (encryption scheme discussed later) before sharing it with the CASB. When the patient data is received by the CASB, CASB indexes it and sends it to cloud storage. When the patient wishes to view the data item, he can search for it through a unique and easy-to-search tracking ID composed of the date of upload, time of upload, and device that triggered the update (e.g., smartwatch_dailyupdate_31/08/22_21:59). A search query will be generated by the patient's mobile application and received by the CASB, which will in turn search for the record from cloud storage and return it in its encrypted form to the patient's device, where it will be decrypted.

2) MEDICAL STAFF

Medical staff will interact with the CASB through a secure desktop or web application deployed on their official devices (say, office computers). The workflow is similar to the patient workflow, except that the main function of medical staff is adding updates to patient health records (for example, the summary of a hospital visit) or retrieving medical records of their patients. Access control functionality is managed by the CASB and described in the following section.

D. CLOUD ACCESS SECURITY BROKER

This is the first of the two main security components we propose to use as part of our solution. The CASB works as an access control and managing module. We propose to use an identity-based encryption scheme such as CP-ABE [57] such that patients can specify exactly which medical staff are allowed to view their data. The scheme can be managed by CASB (i.e., CASB itself can be the key management authority) while the hospitals using the system, or a regulatory body, can be the attribute authority for schemes such as CPABE. We leave the specifics of the scheme to the deployment as the encryption is not our main contribution and many schemes have already been proposed and shown to work well in practice such as in our earlier work [58]. The main point is that CASB can ensure fine-grained and flexible access control through an appropriate encryption scheme.

E. LOGGING MODULE

Secure logging is the second main security component we propose in our solution. This module is specifically targeted at deterring insider attacks.

The intuition behind this module is that malicious insiders, such as CASB administrators who have admin-level privileges, can be deterred from misusing their privileges if there exists a permanent, public, and tamper-resistant record of all their actions. We, therefore, propose a rigorous logging module integrated with CASB. The key idea is to integrate the CASB with a secure and immutable log. When CASB performs any action on a user's request, such as retrieving patient data in response to a patient's request to view his data, the logging module is triggered automatically. A single record of the action will be generated containing the initiating user ID (in this case the patient's ID), the action requested, the unique tracking ID of the data item that is accessed, and the timestamp. This record will then automatically be added to the immutable log.

Specifically, in our design, we propose a private blockchain to serve as this immutable log. Thus, each action will trigger a transaction on this blockchain, where the content of the transaction will be the record specified above. As described in Section III-A, a private blockchain has several different kinds of users. We now describe what the role of each kind of user will be. The nodes can be any user or device that interacts with a private blockchain to view the saved record.

1) VALIDATOR

Every user of a private blockchain is not authorized to perform transactions and validation. Therefore, we introduce

some specialized nodes known as validators which are defined by the organization using the system. The validators are responsible for performing transactions on our private blockchain. Against every user action that generates a log record, a real-time transaction will be performed on the private blockchain by one of the defined validator nodes. Once all other nodes verify this transaction, as outlined in Section III-A, the transaction is immutably stored in a private blockchain. This validation is a fully automated process and no user, even administrators or insider users, will be involved in it.

2) AUDITOR

After performing a transaction and storing data in the private blockchain, the auditor can view each blockchain data using a decentralized web app (DAPP). If the auditor can detect any intrusion in the system, then they can report the relevant administrator or any other user. For example, if a malicious administrator removes patient data from the cloud storage managed by CASB, while the data deletion process can only be initiated by the patient according to CASB policy, the log stored in the private blockchain can be detected by the auditor and they can report those administrators.

3) VIEWERS (CASB USERS)

All users that are registered with the system and allowed to interact with the CASB, i.e., all patients and medical staff, but will have the status of only viewers for the private blockchain. They cannot add any transaction to the blockchain, but they are allowed to explore the blockchain and view the stored logs. the intuition behind this module is to (a) deter malicious insiders from the illegitimate use of their authority, as they will know that their actions are publicly viewable by patients as well as their colleagues and employers, (b) add transparency in the system for patients, who will feel at ease if they can see what is happening to their private data, and (c) to reduce the dependence on the auditor. With a large user base, a very large volume of actions is likely to be stored on the blockchain every day; hence, for an auditor to track illegal accesses from the logs is not very scalable. We argue that a patient who is concerned about the security of his data can simply use a blockchain explorer tool to track his data through its tracking ID and alert the administrator if any unauthorized access is discovered.

The overall workflow of the logging module is summarized in Figure 5. After the initialization of the system, the user can interact with CASB and perform any activity on it. The log data will be generated and stored in the private blockchain by using a validator node. Any user will be able to view the data through a blockchain explorer tool. Many such tools are available like Blockchain [59], and Etherscan [60] The user, who can be a doctor or a patient, looks for the data using the tracking ID. If he does not discover any breaches, then there is no cause for concern; but, if he discovers any undesirable behaviors, he must submit an alert to the auditor, who will

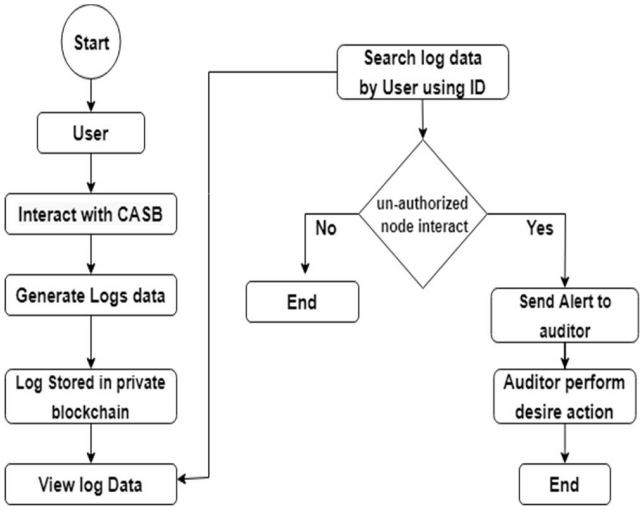


FIGURE 5. Logs data flow.

then take the desired action, which may include blocking the user because they are a malevolent administrator.

Data Verification for Insertion Into the Blockchain: To detect intrusion with a private blockchain, we implement a proof of authority algorithm with the Ethereum blockchain. As we discuss in sections IV-D CASB is only accessible to approved users. Moreover, Once the log data has been generated by the cloud, it will be automatically sent to a decentralized app. Based on a smart contract, the log data is initiated by one validator. According to the proof of authority, there are a fixed n number of authorized nodes that are already selected as shown in Figure 6. The private blockchain receives the request from the authorized node and generates the new block of transactions that needs to be inserted in the peer-to-peer blockchain network. After creating a new block, the validator node is considered a primary node and automatically arranges the transaction of log data into the block and verifies it. After the verification of log data by the primary node all other also verifies the transaction and the block has

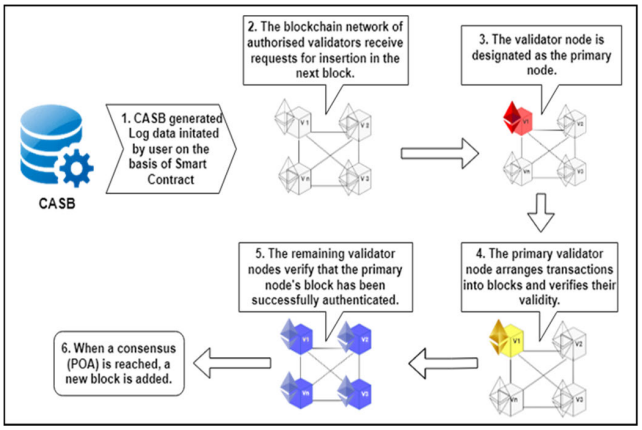


FIGURE 6. Implementing proof of authority.

TABLE 2. The main feature of our design goal.

Features	Description
Decentralization	The Log data will be stored in multiple places that can be accessed by diverse persons. This makes the system resistant to the effects of having a single point of failure
Immutability	The recorded logs data is permanent; it will never be modified or deleted.
Scalability	Using a private blockchain, only a predetermined number of nodes in the network can conduct transactions, making the resulting design model highly scalable.
Integrity	The proposed system ensures the authenticity of stored information by employing a cryptography hashing algorithm (SHA-256).
Non-repudiation	The digital signatures based on asymmetric keys eliminate the possibility of a node disputing a transaction
Auditability	Before committing information to a private blockchain, each node conducted an audit or verification of the log data
Data Tracking	Any user has been able to access their log data, by using a tracking-ID.

been authenticated. In the end, consensus based on proof of authority is achieved then a new block will be added to the private blockchain while any other user including the auditor, an Authorized user by CASB can be allowed to view this data. For intrusion detection, any user like an authorized user by CASB (patient or doctor) feels data breaches, they search for the log action by interacting decentralized app and sending the request to the auditor. Moreover, the Auditor also feels any kind of real health data also performs the desired action on the malicious user. In summary, we highlight the main features of our design against each of the design goals in Table 2, specified in Section IV-A.

V. IMPLEMENTATION

A. TECHNOLOGY STACK

We use the following technology stack for each of the main components:

- **Cloud Storage:** For storing patient data generated by IoT devices, we use Google Cloud storage.
- **Frontend:** The front web application consists of HTML, CSS, and JavaScript programming languages. For developing the Dapp (Decentralized Application) to display log data, we use React.js, a JavaScript library.
- **Backend:** The CASB will be developed using PHP as a backend programming language and smart contract of Ethereum blockchain written in Solidity.
- **Blockchain environment:** We use Ganache which offers multiple nodes with 100 Ether per node. We may build, test, and release your smart contracts and Dapp in a risk-free and deterministic setting when you use Ganache as your development kit. Ganache is a desktop application that is compatible with Ethereum and includes an easy-to-navigate user interface.
- **Wallet:** We use MetaMask wallet to perform blockchain transactions when log data has to be stored.

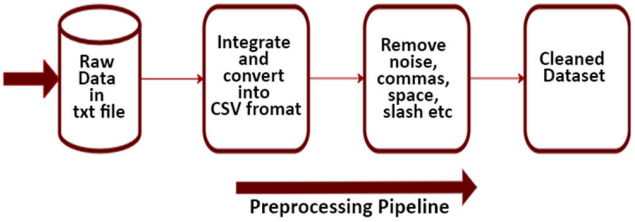


FIGURE 7. Data pre-processing steps.

B. SIMULATING IOT DEVICES UPDATES

We use a dataset generated from wearable IoT devices to simulate updates transmitted from IoT devices to CASB. We used a wearable IoT device dataset downloaded from the UCI machine learning repository [61] in the form of .txt files. This data consists of twelve different activities and 3 sensor devices including Heart monitoring, effects of exercise on the ECG, and blood pressure. We follow pre-processing steps for this data as shown in Figure 7. For the data cleaning process, we have used the Rapid Miner software [62].

C. PRACTICAL IMPLEMENTATION

We have implemented our system for empirical evaluation of its performance. We performed the experiments on an HP Core i5 laptop with 8GB RAM, a 2.6GHz processor, and a 2GB AMD graphic card. The machine ran the Windows 10 operating system and a Google Chrome web browser.

- **Cloud Access Security Broker (CASB):** We developed CASB using the PHP programming language while data was stored on Google Cloud. We used Google API for adding, viewing, updating, and deleting data from Google Cloud. We used an XAMPP local server to run our system. The front end of CASB was created using HTML and Bootstrap. When a user performs any action on the system, the log data is created at the backend of our application. The created logs appear in the Google Chrome Developer Console (for development purposes only) as shown in Figure 8.
- **Permissioned blockchain:** We created a Ganache-based local blockchain with several accounts for performing transactions. Similarly, we wrote blockchain smart contracts using the Solidity programming language and deployed them using Truffle. For the front-end design to display data, we used the JavaScript library React.js. We used Web3.js to monitor and display historical transactions, and MetaMask Wallet to interact with the blockchain from the front end and validate new transactions, as shown in Figure 9.

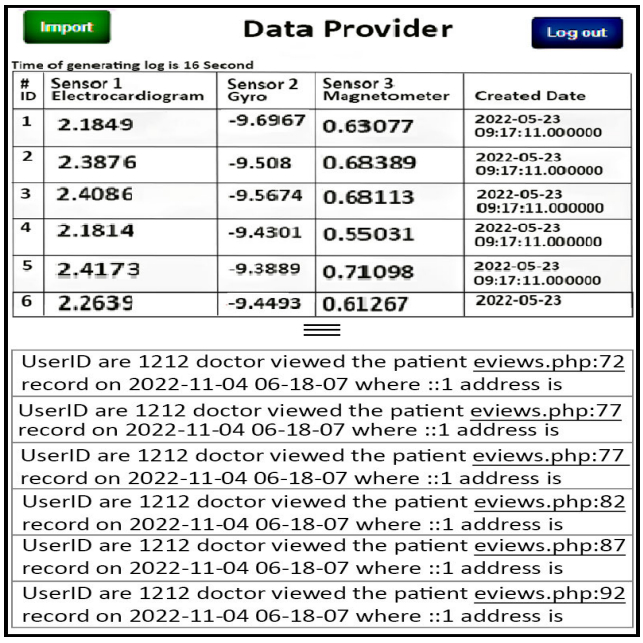


FIGURE 8. Logs data creation.

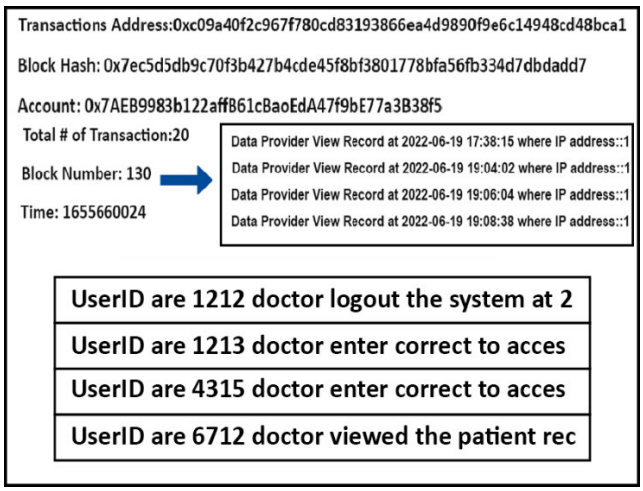


FIGURE 9. Blockchain transaction.

- **Searching the blockchain-based log data:** After storing the log data in the private blockchain, the next step is to retrieve the data from the blockchain so that the auditor can easily detect any malicious activity. As we discussed in Section IV-A, patients can also access stored log data from the blockchain. Due to the large amount of log data, it would be very difficult for patients or auditors to find unwanted activity. Therefore, we also develop a searching module as shown in Figure 10. Thus, any user can easily search for a specific record by entering the tracking ID or other keywords. If any data leak or unauthorized user access is observed, then patients can send an alert to the auditor to take action against the intruding user.

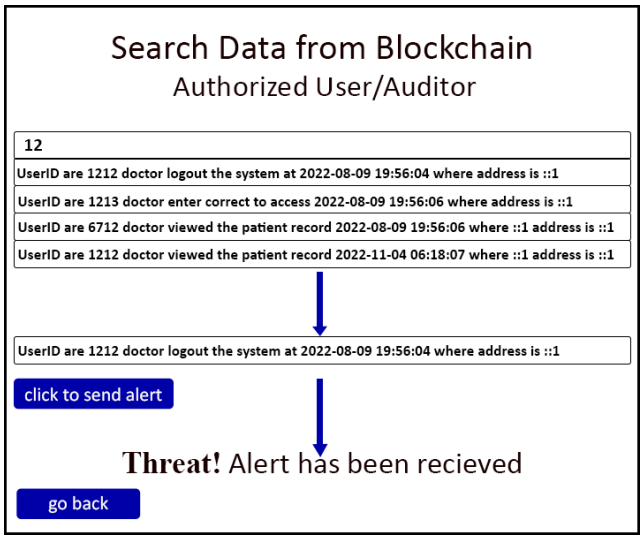


FIGURE 10. Searching logs data.

VI. EXPERIMENTAL EVALUATION

We evaluated our system using the time to transfer log data to the blockchain, relative to the size of the log. The result is shown in Table 3. At the backend of our own design Cloud Access Security Broker (CASB)-based prototype, we write script (Code) that calculates the time of log action and displays the result in terms of seconds on the front end of the screen. To run the system, we perform different actions like adding, deleting, or viewing patient data in web applications. The log data will be generated automatically at the backend of the system based on these actions. The log data is then automatically loaded in DAPP and a transaction is performed to store log data in a private blockchain. Each block contains the most recent logs information, such as *UserID 6712 doctors viewed the patient record on 2022-08-09 at 19:56:06 where an IP address is 192.172.1.1:30.* Also Other important information includes the hash of the previous block, the transaction hash, the block number, the transaction number, the account number that performs the transaction, and the timestamp. The log data will be retrieved from the blockchain and shown in a web application, either as a whole or against a specific search query. We also report the retrieval and search time in our evaluation results. Table 3, shows the results against varying log sizes and number of actions. The column headers are described below:

- Action: Represents the number of actions performed by different users at the same time (e.g., view, add, delete, update data).
- T1: Time required to create log data (at the backend) against the number of user actions.
- T2: Time to store data, i.e., into the private blockchain.
- T3: Time to retrieve data from the private blockchain for the auditor or patient and display it on the web app. During data retrieval, the web application shows all of

TABLE 3. Time in seconds for storing log data against different log sizes and number of actions.

No. of action	LOG DATA SIZE (KB)	T1 (S)	T2 (S)	T3 (s)	Total T (s)	Searching Time ST (s)
1	0.111	0.02	20.54	0.03	20.54	0.054
5	0.656	0.03	21.15	0.03	21.25	0.06
10	1.17	0.04	22.02	0.04	22.12	0.069
20	2.23	0.15	22.98	0.11	23.10	0.087
40	4.36	0.17	23.35	0.16	23.95	0.102
50	5.43	0.17	23.53	0.20	24.23	0.112
100	10.7	0.29	23.61	0.50	25.11	0.208
150	16	0.40	23.79	0.87	25.99	0.227
200	21.3	0.50	24.11	0.93	27.11	0.295
250	26.7	0.52	24.61	0.20	28.51	0.311
300	32	0.71	24.94	1.51	29.74	0.358
350	37.3	0.82	25.31	1.58	31.21	0.398
400	42.6	0.90	25.80	1.63	32.70	0.437
450	48	1.01	26.66	1.74	34.16	0.481
500	53.3	1.3	26.77	2.1	37.57	0.514

the log's data stored in the blockchain, from the first block to the most recent block.

- Total T: Represent overall time from the user being acted storing log data into the private blockchain, i.e., $T1 + T2$.
- ST: The searching time of log data from the blockchain using a tracking ID. The search interface only returns log information in response to a tracking ID query.

A. LOG SIZE

WE start our analysis by varying the number of actions performed by users at the same time, as the log data size increases with the number of actions. As expected, the time to store data increases with the log size. However, the overall size of the log against 500 actions performed by users is only 53.3 KB. Extrapolating from this, the system will require approximately 10,000 actions to be performed simultaneously for a log size of only 1 MB. Thus, log storage is not expected to be a problem in the real implementation of this system.

B. LOG DATA STORAGE IN BLOCKCHAIN

In our second parameter, we analyzed the total time from creating log data in the CASB to displaying it in the web app for an auditor. For this purpose, we calculate the time at different stages of the log creation workflow. We start with time to create log data as shown in Figure 11. For the maximum number of actions, we tested against, i.e., 500 actions where 53.3 KB of log data will be generated, the creation of the log takes only 1.3 seconds. Similarly, the given result shows that when the number of actions increases, the time of creating a log also increases. We extrapolate that when 23,000 user actions are performed at the same time, only then the time to create logs will exceed 1 minute, which

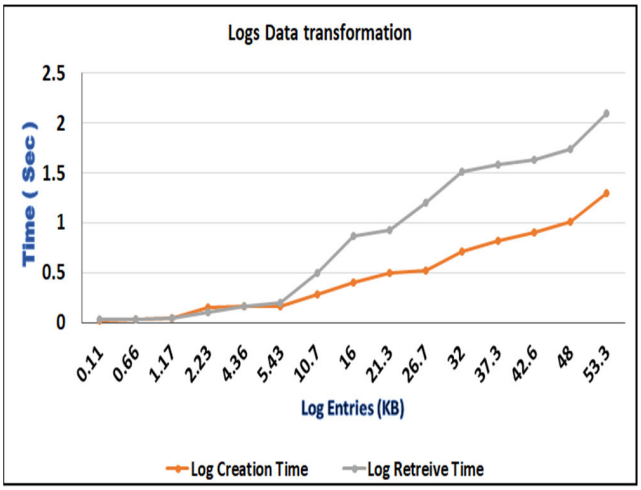


FIGURE 11. Creation and retrieving of logs data.

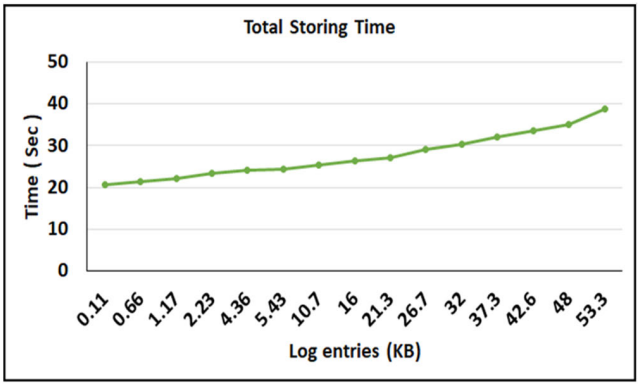


FIGURE 12. Logs data storing time.

we find satisfactory, as in practice, such a large number of actions is not anticipated to occur in parallel. In the next step, we calculate the time to store the logs in the private blockchain. As we see from Figure 12, only storing one action from the log data takes up to 20 seconds, although the size of this data is only 111 Bytes. However, this is standard as storing data in a blockchain requires several steps, and we can see that the time increases very little with a manifold increase in data size. Thus, the blockchain storage time seems to have very little dependence on data size or an increased number of actions. The last step is to fetch the log data from the private blockchain and display it in the web app from where the auditor can read the data. The time to read data from the blockchain is very small, approximately 2 seconds when up to 500 actions are performed by users at the same time. Moreover, the increase in retrieval time when data size increases is also very low. This is good for our system's security because if data retrieval from the blockchain became very slow with a large amount of data being retrieved, it would allow a malicious user to carry out a denial-of-service attack by adding a lot of data or tying up the system in a large retrieval request.

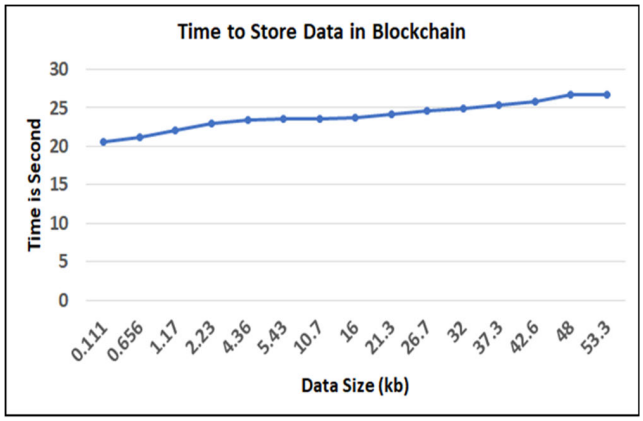


FIGURE 13. Total storing time of logs data from CASB to blockchain.

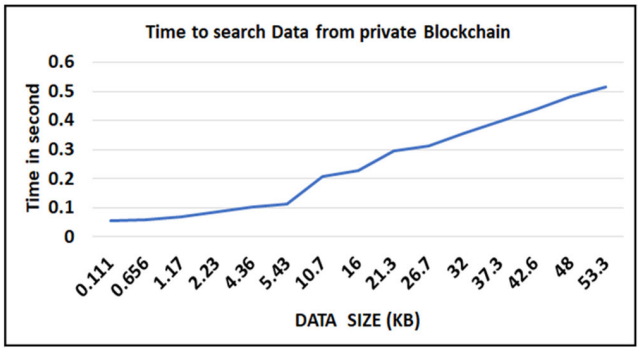


FIGURE 14. Logs searching time.

C. OVERALL PERFORMANCE

The result of the overall flow of log data from creating action to storing into the private blockchain can be calculated by the sum of time to create and time to store data in the private blockchain as shown in Figure 13. We conclude that the main delay in our system is in storing any piece of data in the blockchain (as discussed above, even one action takes approximately 20 seconds to store, which is very high). Although this time does not seem to depend on increasing data very much, it is high, to begin with. The reason is that when a new block is added to the blockchain, all blocks are traversed, and hashes are calculated which is a time-consuming process. Thus, when even a single user action is performed, the storage time in the blockchain is relatively high. Finally, we also evaluate our searching module because a large number of users may interact with the searching module at the same time, causing possible delays due to a large amount of concurrent request traffic. In our evaluation of this module, when we enter the tracking ID for a patient's data, it only takes 0.054 sec when the store's data size is 0.111 kb as shown in Figure 14. When data size increases up to 10.7 KB, our retrieval time is still only 0.208. At 53 KB, the module takes 0.514 sec. Thus, even for a relatively large log size (53 KB is generated against 500 user actions), data searching is quite efficient and peaks at only half a second.

D. SECURITY ANALYSIS

Having evaluated the performance, we now evaluate the security of the proposed system with the list of certain desired properties:

- **Decentralized Distributed Ledger:** We use blockchain technology which is one of the best platforms for achieving the distribution of data like a ledger. In the blockchain, several nodes connect via a peer-to-peer network. When the user performs any transaction in the blockchain then all other connected nodes can access the data by using any of the blockchain data accessing tools. And because the blockchain is decentralized where every user has whole blockchain data. If one of the nodes transfers into the dead state for some time, then all other nodes will not be affected in scenes of data accessing problems.
- **Immutability:** Immutability is a primary requirement of the proposed system. The logs stored and accessed offer many other cloud storages but once data storage it can neither remove nor change updated only offered by blockchain. According to, to proposed private blockchain system, we apply proof of authority that several already selected users can allow storing data in the blockchain while after storing data every transaction data has its hash value and each block header contains all transaction hash. Moreover, the hash of the first block is also stored in the second block header. Similarly, if any malicious user tries to steal or record from the first block transaction then the hash of the first block will not match with the next block and the blockchain connection will be the brake. While blockchain is decentralized in nature and everyone other node has its copy therefore data breaches or immutability problems will be solved by matching the copy user's nodes and recently updated nodes.
- **Non-repudiation:** In Non-repudiation, node A claims that they provide service to B while B says that A does not provide and service. Similarly, in our case, the log data was generated by CASB but validator 1 did not perform any transaction and claimed that he performed the transaction. On the other side, only 2 validators claim that A does not perform transaction while all other validator turns mute. Then there is a digital signature based on the asymmetric key that helps to verify whether the transaction was performed by validator 1 or not.
- **Integrity:** In our proposed system, we use blockchain technology which is a distributed ledger while data integrity will be ensured by a hashing algorithm, which is a built-in characteristic of the blockchain ledger.
- **Scalability:** In general, blockchain is a decentralized network and the number of nodes may be increased over time while each node has equal right to perform transactions. Increasing the node to improve the scalability of the blockchain network is a very big challenge. In our

proposed system, we employed a private blockchain with a proof of authority (POA) algorithm. The POA contains a fixed number of validators that allow to perform the transaction and also validate the transaction performed by another validator. So, increasing network node issues will be resolved and the proposed system will be scalable.

- **Privacy and data confidentiality:** As we discuss in section III, our proposed system consists of two parts. The CASB provides security to real health data from outsider attacks and also integrates a private blockchain log management system that continuously detects insider or administrator activity. Moreover, the log data is generated based on each user action. It contains very minimal information about the user like *user designation, ID, Date, time or IP* while does not include any blatant displays of personal data. The log data is only helpful for auditors. Therefore, there's no chance to steal or remove real health data by using log information.
- **DDOS attacks (resist):** The DDOS (Distributed denial of service) attack occurs in the blockchain network when a large number of transactions are performed by some malicious nodes to target normal traffic. In our proposed solution, this attack is not possible because we use the POA algorithm which contains pre-authenticated nodes also known as a validator.
- **51% attacks (resist):** Similarly, a 51% attack is also possible when the maximum node validates the wrong transaction for example out of 100, the 51 nodes say validate the wrong transaction and store data in the blockchain. This attack is much harder in our proposed private blockchain network. Because only a pre-authenticated and fixed number of nodes are allowed to perform the transaction.

In the end, our private blockchain system has many features that make it highly compatible with securely detecting insider attacks.

VII. COMPARISON OF REPRESENTATIVE SOLUTION WITH PROPOSED SCHEMES

In this section, we compare the essential qualities of the most prominent solutions with our work as shown in Table 4. In our proposed system, we make a significant new contribution i.e., integrate the blockchain-based log mechanism directly with CASB (Cloud Access Security Broker) and permanently store log data without the involvement of any type of log file.

The proposed method relies on the inherent hashing property of blockchain therefore it's immutable. In contrast, several existing methods, such as the one described by Cueva-Sanchez et al. [22], employ off-chain methodologies, which offer challenges in maintaining system immutability. The proposed system also ensures non-repudiation by applying a digital signature that is based on an asymmetric key. This digital signature aids in the verification of whether the transaction was executed by an authorized user or not. Existing

TABLE 4. Comparative analysis of existing schemes that use blockchain.

Schemes	Security Feature						
	Immutability	Non-repudiation	Scalability	Transparency	Privacy	Accessibility	Data Integrity
Cueva-Sanchez et al. [22]	○	●	◐	○	◐	◐	○
Rakib et al. [30]	●	●	○	●	●	○	●
Lu et al. [43]	●	○	○	●	●	○	○
Madhwal et al. [29]	●	●	○	●	●	○	●
Jadidi et al. [38]	●	◐	○	○	●	○	○
Chien-Lung et al. [32]	●	●	○	●	●	○	●
Proposed Scheme	●	●	●	●	●	●	●

related systems like Madhwal et al. [29], Rakib et al. [30], and Chien-Lung et al. [32] implement proof of work [33] in the consensus algorithm. However, when the number of nodes increases over time. it's difficult to maintain the scalability system. But in the proposed system, we used proof of authority that contains pre-defined authenticated users who possess the necessary permissions to execute. transactions.

Another key advantage over the prior proposed system is in terms of security: Our main objective is to maintain the privacy of patients' real health; therefore, we use CASB to secure records from outside attackers. Additionally, we incorporate a private blockchain to identify and mitigate potential risks posed by internal actors with malicious intent. Further, to its advantage, our proposed system provides accessibility by allowing every user to view any log data by our search module as well as any authenticated nodes are permitted to execute transactions. The proposed scheme also provides data integrity by the utilization of cryptographic hashing techniques, which guarantee the veracity of stored information.

VIII. CONCLUSION

We have presented a private blockchain-based remote health monitoring system to protect against insider attacks. The proposed system offers immutability, distribution, and partial decentralization. The two components of our system are the Cloud Access Security Broker (CASB) for managing real health data and a private blockchain to continuously monitor each user's behaviors for detecting insider attacks. CASB would provide end-to-end security, which includes Authentication, Access Control, and Storage, while all user actions are logged and stored in the blockchain. However, due to blockchain's immutability, tampering or theft of log data is not possible. In addition, any user of the system including the auditors, patients, or doctors can search their log data

with ID from the blockchain and detect the administrator's malicious behaviors. Moreover, we practically implemented our system using the Ethereum blockchain and evaluated the performance of the system.

IX. FUTURE DIRECTION

In the future, the proposed approach will be extended to handle big log data. In the current scenario, we practically implement and test the performance with a small amount of data i.e., KB or MB but with time a large amount of data has been created which may be in GB or TB. Although blockchain has no option to remove data. Moreover, the basic requirement of our proposed system is that nobody can update or delete the log data. Therefore, due to increasing the size of log data in the blockchain, we will require any mechanism to compress this data. but the blockchain also has no option to compress this data. Therefore, the compression process will be possible on the cloud side that compresses every action of the user and stores it in the blockchain also compression does not affect real health data processing. Furthermore, for strong tamper-evidence & audibility, in the future, we may apply ledgerDB type stat-of-the art techniques that are capable of facilitating verifiable data removals, a feature that is highly sought after in various practical applications. This functionality allows for the elimination of outdated records to optimize storage space and the concealment of some records to comply with regulatory requirements, all while maintaining the system's capacity to be verified. Similarly, in our proposed system we will integrate private blockchain with CASB (Cloud Access Security Broker) and make like bridge structure but there is a little bit of chance that attackers may be trying to attack this bridge. Therefore, in the future, we will try to apply a hardware-based TPM (Trusted Platform Module) type solution to prevent disabling this logging module entirely.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] S. Sengupta, "A secured biometric-based authentication scheme in IoT-based patient monitoring system," in *Emerging Technology in Modelling and Graphics*, 2020, pp. 501–518.

[2] J. Sun, X. Yao, S. Wang, and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020.

[3] (2022). *Bitglass CASB*. [Online]. Available: https://www.bitglass.com/casb-cloud-access-security-broker

[4] (2022). *Lookout CASB*. [Online]. Available: https://www.lookout.com/products/casb-cloud-access-security-broker

[5] *Cisco Cloudlock*. https://www.cisco.com/c/en/us/products/security/cloudlock/index.html

[6] *Microsoft Cloud App Security*. https://www.microsoft.com/en-us/security/business/siem-and-xdr/microsoft-defender-cloud-apps

[7] *Cloud-Access-Security-Broker-CASB*. [Online]. Available: https://www.techtarget.com/searchcloudcomputing/definition/cloud-access-security-broker-CASB

[8] *Casb*. [Online]. Available: https://www.proofpoint.com/us/threat-reference/casb/

[9] *ObserverIT Cost of Insider Threats Global Report 2020*. [Online]. Available: https://www.proofpoint.com/us/products/information-protection/insider-threat-management

[10] *The Colombia University Researchers Perform Survey in 2019*. [Online]. Available: https://delinea.com/blog/insider-threats-in-cyber-security

[11] *Real world Insider Attack Example*. [Online]. Available: https://www.tessian.com/blog/insider-threats-types-and-real-world-examples/

[12] *Insider Threats at Hospitals*. https://resources.infosecinstitute.com/topic/insider-threats-at-hospitals/

[13] H. Halpin and M. Piekarska, "Introduction to security and privacy on the blockchain," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Apr. 2017, pp. 1–3.

[14] T. Yu, Z. Lin, and Q. Tang, "Blockchain: The introduction and its application in financial accounting," *J. Corporate Accounting Finance*, vol. 29, no. 4, pp. 37–47, Oct. 2018.

[15] P. Gomber, *Hinz-O. Nofer M. Schiereck D.,'Blockchain*, vol. 59. Cham, Switzerland: Springer, 2017, pp. 183–187.

[16] M. Cinque, D. Cotroneo, and A. Pecchia, "Event logs for the analysis of software failures: A rule-based approach," *IEEE Trans. Softw. Eng.*, vol. 39, no. 6, pp. 806–821, Jun. 2013.

[17] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," in *Decentralized Business Review*, 2008.

[18] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics Informat.*, vol. 36, pp. 55–81, Mar. 2019.

[19] T.-V.-L. T.-V. Le and C.-L.-H. T.-V. Le, "A systematic literature review of blockchain technology: Security properties, applications and challenges," *J. Internet Technol.*, vol. 22, no. 4, pp. 789–801, Jul. 2021.

[20] M. S. Kumar and V. Nagalakshmi, "Secure transfer of robust healthcare data using blockchain-based privacy," *Cluster Comput.*, pp. 1–17, May 2023.

[21] S. Sahai, M. Atre, S. Sharma, R. Gupta, and S. K. Shukla, "Verity: Blockchain based framework to detect insider attacks in DBMS," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Nov. 2020, pp. 26–35.

[22] J. J. Cueva-Sánchez, A. J. Coyco-Ordemar, and W. Ugarte, "A blockchain-based technological solution to ensure data transparency of the wood supply chain," in *Proc. IEEE ANDESCON*, Oct. 2020, pp. 1–6.

[23] V. Zieglmeier and G. L. Daiqui, "GDPR-compliant use of blockchain for secure usage logs," in *Evaluation and Assessment in Software Engineering*, 2021, pp. 313–320.

[24] R. Adlam and B. Haskins, "A permissioned blockchain approach to electronic health record audit logs," in *Proc. 2nd Int. Conf. Intell. Innov. Comput. Appl.*, Sep. 2020, pp. p1–7.

[25] S. Ma, Y. Cao, and L. Xiong, "Efficient logging and querying for blockchain-based cross-site genomic dataset access audit," *BMC Med. Genomics*, vol. 13, no. S7, pp. 1–13, Jul. 2020.

[26] S. Akbar, S. Khan, F. Ali, M. Hayat, M. Qasim, and S. Gul, "IHBP-DeepPSSM: Identifying hormone binding proteins using PsePSSM based evolutionary features and deep learning approach," *Chemometric Intell. Lab. Syst.*, vol. 204, Sep. 2020, Art. no. 104103.

[27] J. Eberhardt and J. Heiss, "Off-chaining models and approaches to off-chain computations," in *Proc. 2nd Workshop Scalable Resilient Infrastructures Distrib. Ledgers*, Dec. 2018, pp. p7–12.

[28] A. Ismailisufi, T. Popovic, N. Gligoric, S. Radonjic, and S. Šandi, "A private blockchain implementation using multichain open source platform," in *Proc. 24th Int. Conf. Inf. Technol. (IT)*, Feb. 2020, pp. 1–4.

[29] Y. Madhwal, I. Chistiakov, and Y. Yanovich, "Logging multi-component supply chain production in blockchain," in *Proc. 4th Int. Conf. Comput. Manage. Bus.*, Jan. 2021, pp. 83–88.

[30] M. H. Rakib, S. Hossain, M. Jahan, and U. Kabir, "Towards blockchain-driven network log management system," in *Proc. IEEE 8th Int. Conf. Smart City Informatization (iSCI)*, Dec. 2020, pp. 73–80.

[31] W. Zhao, S. Yang, and X. Luo, "Secure hierarchical processing and logging of sensing data and IoT events with blockchain," in *Proc. The 2nd Int. Conf. Blockchain Technol.*, Mar. 2020, pp. p52–56.

[32] C.-L. Hsu, W.-X. Chen, and T.-V. Le, "An autonomous log storage management protocol with blockchain mechanism and access control for the Internet of Things," *Sensors*, vol. 20, no. 22, p. 6471, Nov. 2020.

[33] N. Shi, "A new proof-of-work mechanism for bitcoin," *Financial Innov.*, vol. 2, no. 1, pp. 1–8, Dec. 2016.

[34] A. Ali, A. Khan, M. Ahmed, and G. Jeon, "BCALS: Blockchain-based secure log management system for cloud computing," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, Apr. 2022, Art. no. e4272.
[35] D. M. Maslove, J. Klein, K. Brohman, and P. Martin, "Using blockchain technology to manage clinical trials data: A proof-of-concept study," *JMIR Med. Informat.*, vol. 6, no. 4, Dec. 2018, Art. no. e11949.
[36] B. d. A. Mendonça and P. Matias, "Auditchain: A mechanism for ensuring logs integrity based on proof of existence in a public blockchain," in *Proc. 11th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Apr. 2021, pp. 1–5.
[37] Y. Zhang, S. Wu, B. Jin, and J. Du, "A blockchain-based process provenance for cloud forensics," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 2470–2473.
[38] Z. Jadidi, A. Dorri, R. Jurdak, and C. Fidge, "Securing manufacturing using blockchain," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 1920–1925.
[39] R. Upton, S. Clulow, M. J. Mahony, and J. Clulow, "Generation of a sexually mature individual of the eastern dwarf tree frog, *Litoria fallax*, from cryopreserved testicular macerates: Proof of capacity of cryopreserved sperm derived offspring to complete development," *Conservation Physiol.*, vol. 6, no. 1, Jan. 2018, Art. no. coy043.
[40] C. Yenugunti and S. S. Yau, "A blockchain approach to identifying compromised nodes in collaborative intrusion detection systems," in *Proc. IEEE Int. Conf Depend., Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Aug. 2020, pp. 87–93.
[41] C. Klinkmüller, I. Weber, A. Ponomarev, A. B. Tran, and W. van der Aalst, "Efficient logging for blockchain applications," 2020, *arXiv:2001.10281*.
[42] T.-V. Le, C.-L. Hsu, and W.-X. Chen, "A hybrid blockchain-based log management scheme with nonrepudiation for smart grids," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 5771–5782, Sep. 2022.
[43] F. Lu, W. Li, H. Jin, L. Gan, and A. Y. Zomaya, "Shadow-chain: A decentralized storage system for log data," *IEEE Netw.*, vol. 34, no. 4, pp. 68–74, Jul. 2020.
[44] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, 4th Quart., 2018.
[45] C. Jaikaran, *Blockchain: Background and Policy Issues*. Washington, DC, USA: Congressional Research Service, 2018.
[46] C. V. Helliar, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani, "Permissionless and permissioned blockchain diffusion," *Int. J. Inf. Manage.*, vol. 54, Oct. 2020, Art. no. 102136.
[47] E. Hofmann, U. M. Strewe, N. Bosia, E. Hofmann, U. M. Strewe, and N. Bosia, "Background III—What is blockchain technology?" *Supply Chain Finance Blockchain Technology*, 2018, pp. 35–49.
[48] N. D. Pattengale and C. M. Hudson, "Decentralized genomics audit logging via permissioned blockchain ledgering," *BMC Med. Genomics*, vol. 13, no. S7, pp. 1–9, Jul. 2020.
[49] S. Saxena, B. Bhushan, and M. A. Ahad, "Blockchain based solutions to secure IoT: Background, integration trends and a way forward," *J. Netw. Comput. Appl.*, vol. 181, May 2021, Art. no. 103050.
[50] E. B. Fernandez, N. Yoshioka, and H. Washizaki, "Cloud access security broker (CASB): A pattern for secure access to cloud services," in *Proc. 4th Asian Conf. Pattern Lang. Programs, Asian PLoP*, 2015.
[51] S. Ahmad, S. Mehfuz, F. Mebarek-Oudina, and J. Beg, "RSM analysis based cloud access security broker: A systematic literature review," *Cluster Comput.*, vol. 25, no. 5, pp. 3733–3763, Oct. 2022.
[52] *Amazon S3 Bucket*. [Online]. Available: https://aws.amazon.com/s3/
[53] *Microsoft365*. [Online]. Available: https://www.microsoft.com/en/microsoft-365/
[54] *Saleforce*. [Online]. Available: https://www.salesforce.com/in/?ir=1/
[55] E. B. Fernandez, N. Yoshioka, and H. Washizaki, "Patterns for security and privacy in cloud ecosystems," in *Proc. IEEE 2nd Workshop Evolving Secur. Privacy Requirements Eng. (ESPRE)*, Aug. 2015, pp. 13–18.
[56] S. Eftimie, L. Dumitru, and V. Opriş, "Cloud access security brokers," in *Education and Creativity for a Knowledge-Based Society*, 2016.
[57] R. Xu and B. Lang, "A CP-ABE scheme with hidden policy and its application in cloud computing," *Int. J. Cloud Comput.*, vol. 4, no. 4, p. 279, 2015.
[58] Z. Abaid, A. Shaghaghi, R. Gunawardena, S. Seneviratne, A. Seneviratne, and S. Jha, "Health access broker: Secure, patient-controlled management of personal health records in the cloud," in *Proc. 13th Int. Conf. Comput. Intell. Secur. Inf. Syst. (CISIS)*, 2021, pp. 111–121.
[59] *Blockchair*. [Online]. Available: https://blockchair.com/ethereum/testnet
[60] *Etherscan*. [Online]. Available: https://ropsten.etherscan.io/
[61] *Mobile Health Dataset Named as Mhealt*. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/mhealth+dataset
[62] *Rapidminer*. [Online]. Available: https://rapidminer.com/

HAMZA JAVED received the B.S. degree in computer science from Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi, Pakistan, in 2017, and the M.S. degree in computer science from the FAST-National University of Computer and Emerging Science (NUCES), Islamabad, Pakistan. He is currently a Lecturer with Muslim Youth University, Islamabad. He was a Teacher Assistant with FAST-NUCES, from August 2022 to February 2023. His research work is carried out in the field of blockchain, cryptography and health security.

ZAINAB ABAID received the Ph.D. degree from The University of New South Wales, Australia. She is currently an Assistant Professor with the FAST-National University of Computer and Emerging Science, Islamabad, Pakistan. Her research interests include malware detection and mitigation, adversarial machine learning, secure e-health application, and the application of speech recognition to Arabic learning tasks.

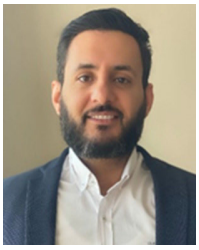
SHAHID AKBAR received the bachelor's degree in computer science and information technology from the Islamic University of Technology, Bangladesh, in 2011, and the M.S. and Ph.D. degrees in computer science from Abdul Wali Khan University (AWKU), Pakistan, in 2015 and 2021, respectively. His research interests include bioinformatics, digital image processing, biomedical engineering, machine learning, and deep learning.

KIFAYAT ULLAH received the bachelor's and M.S. degrees in electrical engineering from the Sarhad University of Science and Information Technology, Pakistan, in 2018 and 2021, respectively. His areas of interests include automation, cyber security, machine learning, and the IoT.

ASHFAQ AHMAD received the M.S. and Ph.D. degrees in computer science from Abdul Wali Khan University (AWKU), Pakistan, in 2016 and 2023, respectively. Currently, he is an Assistant Professor with the Department of Computer Science, Muslim Youth University, Islamabad. His areas of interests include machine learning, deep learning, and bioinformatics.

AAMIR SAEED received the Ph.D. degree in wireless communication from Aalborg University, Denmark. He is currently an Assistant Professor with the Department of Computer Science and IT, University of Engineering and Technology. His research interests include big data structures (LSM and Bloom filters), micro-services architecture, and IoT with security in focus.

HASHIM ALI received the Ph.D. degree in computer science from Abdul Wali Khan University Mardan, Pakistan. He is currently an Assistant Professor with the Department of Computer Science, Abdul Wali Khan University Mardan. His research interests include cloud computing, software testing, agile processes, energy-efficient systems, and enterprise systems. He is proficient in computer systems, both theoretically and practically.



YAZEED YASIN GHADI received the Ph.D. degree in electrical and computer engineering from Queensland University. He is currently an Assistant Professor of software engineering with Al Ain University. He was a Postdoctoral Researcher with Queensland University before joining Al Ain. He has published more than 80 peer-reviewed journal and conference papers and holds three pending patents. His current research is on developing novel electro-acousto-optic neural interfaces for large-scale high-resolution electrophysiology and distributed optogenetic stimulation. He was a recipient of several awards. His dissertation on developing novel hybrid plasmonic photonic on chip biochemical sensors received the Sigma Xi Best Ph.D. Thesis Award.

TAHANI JASER ALAHMADI received the B.S. degree in computer science and the M.S. degree in information technology (data management), and the Ph.D. degree from the Faculty of Information Technology, Griffith University, Australia, in 2019. She is currently an Assistant Professor with the Faculty of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Saudi Arabia. Her research interests include innovative research methods in data analysis and mining, the development of data analysis algorithms as a digital accessibility assessment tool, and artificial intelligence implementation for enhancing digital accessibility. She is a member of the Golden Key Society and a Media Access Australia. She received multiple awards, such as the Google Doctoral Consortium Award, Perth, in 2017; and the Institute for Integrated and Intelligent Systems (IIIS) Award for Quality and Impact Research, Brisbane, in 2016.

HEND KHALID ALKAHTANI (Member, IEEE) received the B.Sc. degree in computer science from the School of Engineering and Applied Science, The George Washington University, in 1992, the M.Sc. degree in information management from the Department of Engineering Management, The George Washington University, in 1993, and the Ph.D. degree in information security from the Department of Computer Science, Loughborough University, in 2018. She is an Assistant Professor with the Information Systems Department, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University. She has 23 years of work experience as a lecturer and worked as a computer center president and a statistic center president with faculty collages. She received the award from SIDF Academy: Leading Creative Transformation in Critical Time Program, Center for Professional Development, Stanford University.

ALI RAZA received the bachelor's degree in computer science from the University of Peshawar, Pakistan, in 2013, and the M.S. degree in computer science from City University Peshawar (CUSIT), Pakistan, in 2018. He is currently pursuing the Ph.D. degree with Qurtuba University, Peshawar, Pakistan. He is also a Lecturer with the Department of Computer Science, MY University, Islamabad. His research interests include bioinformatics, machine learning, and deep learning.