electronics

MDPI

Article

Enhancing Cloud Communication Security: A Blockchain-Powered Framework with Attribute-Aware Encryption

Raghunandan K. R. [1], Bhavya Kallapu [2,*], Radhakrishna Dodmane [1], Krishnaraj Rao N. S. [3], Srinivasarao Thota [4] and Aditya Kumar Sahu [5,*]

[1] Department of Computer Science and Engineering, NMAM Institute of Technology—Affiliated to Nitte (Deemed to be University), Karnataka 574110, India; raghunandan@nitte.edu.in (R.K.R.); rkdodmane@gmail.com (R.D.)
[2] Department of Mathematics, NMAM Institute of Technology—Affiliated to Nitte (Deemed to be University), Karnataka 574110, India
[3] Department of Information Science and Engineering, NMAM Institute of Technology—Affiliated to Nitte (Deemed to be University), Karnataka 574110, India; krisndi@gmail.com
[4] Department of Mathematics, Amrita School of Physical Sciences, Amrita Vishwa Vidyapeetham, Amaravati 522503, Andra Pradesh, India; t_srinivasarao@av.amrita.edu
[5] Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amaravati 522503, Andra Pradesh, India
* Correspondence: bhavyak@nitte.edu.in (B.K.); adityasahu.cse@gmail.com (A.K.S.)

Abstract: The global production of information continuously increases in quantity and variety. However, the tools and technologies developed to handle such large volumes of data have not adequately met the security and privacy requirements. Existing cloud security systems, often managed by a trusted third party, are susceptible to various security risks. To address these challenges and ensure the protection of personal information, blockchain technology emerges as a crucial solution with substantial potential. This research uses the blockchain-powered attribute-aware encryption method to establish a real-time secure communication approach over the cloud. By employing attribute-based encryption technology, data owners can implement fine-grained search permissions for data users. The proposed solution incorporates accessible encryption technology to enable secure access to encrypted data and facilitate keyword searches on the blockchain. This study provides a functional comparison of recently developed attribute-based encryption algorithms. The access control strategy comprises two access tree types and a linear secret-sharing system, serving as the main components. The elliptic curve's base field was set to 512b, and the bilinear pairing parameter type used was Type-A. This approach involves storing keywords on a remote server and encrypting them using attribute-based encryption. Furthermore, the encrypted data blockchain and the corresponding ciphertext are stored in the blockchain. Numerical experiments were conducted to evaluate the system's key generation, trapdoor building, and keyword retrieval capabilities.

Keywords: attribute-aware encryption; blockchain; cloud; ciphertext; fine-grained data access; trapdoor security

Citation: R., R.K.; Kallapu, B.; Dodmane, R.; S., K.R.N.; Thota, S.; Sahu, A.K. Enhancing Cloud Communication Security: A Blockchain-Powered Framework with Attribute-Aware Encryption. *Electronics* **2023**, *12*, 3890. https://doi.org/10.3390/electronics12183890

Academic Editors: Prince Waqas Khan, Pyae Pyae Phyo and Khizar Abbas

Received: 17 August 2023
Revised: 7 September 2023
Accepted: 12 September 2023
Published: 14 September 2023

1. Introduction

Cloud computing has revolutionized the way data are managed and stored, offering a cost-effective solution that is flexible and scalable. However, as the volume of data being stored and exchanged on the cloud continues to grow exponentially, several security and privacy issues must be addressed to ensure the safety of user data. One of the most significant challenges facing cloud computing is securely exchanging data. The following paragraph provides an overview of the existing landscape in cloud computing, highlighting its revolutionary impact on data management and storage, as well as the challenges it faces

in terms of security and privacy. It also mentions traditional encryption methods and their limitations in addressing these challenges.

Traditional encryption methods are no longer suitable for cloud data exchange security as they produce multiple encrypted versions of the same data using different keys. Attribute-based encryption has been proposed as a possible solution, but it still relies on trusted third parties to protect users' privacy [1]. Another issue with cloud computing is the need for data files to be encrypted before they are stored on a cloud server. Most cloud servers are not entirely trustworthy and reliable, and data files may need to be downloaded locally before they can be decrypted, wasting network bandwidth and time. Searchable encryption (SE) has been put forth to address these issues. SE allows data to be encrypted while enabling users to search for and retrieve specific pieces of data without downloading and decrypting the entire file. This approach is more efficient and can save time and resources. Despite these advances, cloud computing still has security and privacy concerns, particularly around authentication processes, identity authentication, and storage security. Users must be vigilant about securing their data, and cloud service providers must implement robust security measures to protect user data from unauthorized access or attack. In conclusion, cloud computing has revolutionized managing and storing data, providing a cost-effective and flexible solution. However, security and privacy concerns must be addressed to ensure the safety of user data, and new approaches, such as searchable encryption, must be adapted to meet the evolving needs of cloud computing. This paper introduces attribute-based encryption as a potential solution and touches on the need for encrypted data files and the concept of searchable encryption to enhance security and efficiency.

The Internet of Things (IoT) has revolutionized our daily lives with the massive amount of data it generates. However, storing and managing these data is a significant challenge due to the limited resources of IoT devices. Cloud storage is an effective solution but raises security and privacy concerns, such as unauthorized data access and manipulation. Symmetric encryption is a standard technique that provides data confidentiality. However, it does not enable accurate data sharing or searching. Keyword-based searchable attribute-based encryption (KSABE) is a more comprehensive solution that offers data protection and granular access control. The ability to conduct keyword-based searches is especially critical for data users. However, the decryption process is computationally intensive, and managing large user keys is challenging when applying attribute-based encryption techniques to IoT. Blockchain-aided searchable attribute-based encryption (BC-SABE) is a promising solution that addresses these challenges effectively. This system employs a decentralized blockchain system to manage threshold parameter construction, key management, and user revocation. It enables efficient revocation and decryption without updating keys or re-encrypting ciphertext. This is because the blockchain holds all revocation procedures. Furthermore, the coalition blockchain allows users to create partial tokens, enhancing their privacy. BC-SABE is an innovative approach to IoT data management that addresses security and privacy concerns while enabling accurate data sharing and searching. It holds great potential for future IoT applications, especially when data protection, access control, and efficient revocation are critical.

Bitcoin is a type of digital currency and a pioneer of blockchain technology. It works as a distributed, open database that stores transactions in blocks. Each block is connected to the previous block using a hash function and a peer-to-peer network. The database uses the Merkle tree structure to store specific transactions in the block. If a rogue user tries to modify a transaction, the block's hash will change, affecting the Merkle tree's root hash. However, the network's consensus mechanism makes it difficult for a rogue user to carry out such an attack. They would need to have 51% of the network's computational power to obstruct a transaction. Even if they had this computational power, they would not be able to obtain the recognition of other nodes in the network or gain access to the blockchain. This is because the consensus mechanism ensures that all nodes in the network agree on the validity of a transaction. If a rogue user tries to alter a transaction, other nodes

in the network will reject it, and the blockchain will not acknowledge it. The consensus mechanism also prevents double-spending, where a user spends the same bitcoin twice.

Blockchain technology is an open, decentralized, and independent system that does not require third-party management of transactions. Transactions are automatically distributed across the entire network and secured through workload-proof strategies and encryption technology, making it more secure than conventional data storage methods. The proposed blockchain-based distributed cloud storage infrastructure, such as the Block-Secure technology, uses a digital signature mechanism (DSA) to encrypt and sign user's files' blocks. To reduce the load on the peer-to-peer (P2P) network and quickly deliver users' files from the cloud, a random file replication placement strategy is employed. In addition, the Merkle hash tree is used as a validation tool to provide file integrity verification. The Block-Secure system uses a genetic algorithm to address the file block replica placement problem across multiple users and different data centers in a distributed cloud storage environment. Integrating blockchain technology in the cloud storage infrastructure provides a more secure and decentralized system that ensures data privacy and integrity. Using genetic algorithms and validation tools such as the Merkle hash tree enhances the system's efficiency and reliability in managing distributed data storage.

SE [2] enables users to search encrypted data files using keywords, significantly reducing user communication and computation. The majority of existing SEs are based on essential public SEs. The authors of [3] proposed a general basic SE application scheme in the mobile medical system and applied significant public SE to medical information management. The authors of [4] suggested a comparatively secure public key SE scheme using the random oracle model to address the security of offline keyword guessing attacks. But most SEs using public keys only support one-to-one encryption and decryption. The data owner cannot implement the outsourced encrypted data—effective access control, which lacks convenience and practicability in the actual application environment—because each encryption requires the recipient's identity information to be known, and the search authority of the search user is not considered. Data users can perform unlimited searches using any keyword to request encrypted content from the server that contains their desired keywords using any of the SE schemes mentioned above. Since data owners cannot impose adequate access controls to outsourced data information, researchers need attribute-based encryption technology to develop a SE scheme with keyword search authorization. The idea of attribute-based encryption was first put forth in [5], and it implements fine-grained access control of data through fuzzy identification. This particular cryptographic primitive is brand new. The authors of [6] then published an attribute-based encryption system that incorporates attributes into keys to accomplish fine-grained data access control. To balance user experience and security in data outsourcing, the authors of [7] then proposed a fine-grained data SE method. They mentioned the potential applications for it in a safe mobile cloud environment. In the attribute-based encryption approach, the key to ensuring good communication is embedded with attributes [8].

In contrast to other traditional encryption techniques, attribute-based encryption provides a collusion-resistance property. As a result, an adversary needs the user key to access encrypted files. Multiple collusion attacks can be stopped by using attribute-based encryption for login security. Therefore, the authorized user with the same qualities may access or decrypt the encrypted files. Distributed attribute-based encryption, multi-authority attribute-based encryption, attribute-based broadcast encryption, and ciphertext policy attribute-based encryption were just a few of the hybrids made simpler by the ABE technique. Data analysts classify various encryption techniques based on how crucial they are for data security. Attribute-based encryption finds and encrypts the user key using attribute sets. As a result, the client is permitted to distribute, manage, and maintain the PHR using identity sets. ABE, however, forbids the use of user revocation characteristics. Public-key encryption features cryptography, public keys, and private keys. The user can then decode communications using the associated private or public key. The protected file can be accessed by two users thanks to public-key encryption.

Consequently, the user must have both public and private keys. The public key can encrypt private data on a cloud server, while the private key is used to decrypt encrypted communication. However, the message is sent to the output server via the ciphertext. Hence, the public-key technique's key encryption device is its most crucial component. The user may then use the private key to protect and authenticate the integrity of the data. The public-key technique, however, imposes restrictions on the encryption procedure. Therefore, the user must run various algorithms to relay and receive encrypted messages. AES, or Advanced Encryption Standard, is a symmetric encryption technique that encrypts data blocks of 128 bits at a time. The keys used to encrypt these data blocks have lengths of 128, 192, and 256 bits. A 256-bit key requires 14 rounds of data encryption, a 192-bit key requires 12 rounds, and a 128-bit key requires 10 rounds. Each cycle has several steps for operations, such as plaintext mixing, substitution, and transposition. File encryption, secure sockets layer/transport layer security, mobile app encryption, and Wi-Fi security may all be accomplished with AES.

Data sharing and storage in cloud storage platforms are the focus of cloud data security (CSP). The DO is allowed to upload encrypted data. Others must obtain their own decryption key with the DO's permission to decrypt the data. In other words, clients who meet the requirements can decode the data using their private keys after they have only been encrypted once. The uploaded data are entirely within the DO's control and the DO is accountable for its posted data. Users' actions will all be preserved and unaffected in their entirety. Users cannot retract their behavior. This supports fine-grained access control, in which the DO encrypts the data. After meeting the attribute requirements, other users can collaborate with the DO to develop a unique decryption key. The administration of keys and data storage are independent of outside parties. A third party cannot affect the DO's key generation and data encryption. The integrity of the key, ciphertext, and plaintext is required. The user cannot obtain the correct decrypted data if one integrity has been lost. The DOs have control over the data they post. Since the cloud platform is public, anybody may view the uploaded data that have been encrypted. Users must, however, negotiate the key with the DO if they wish to be able to decrypt the data. It is important to note that ABE techniques have their own strengths and limitations, and the choice of which scheme to use depends on the application's specific requirements. For example, KP-ABE is more suitable for scenarios with a large number of users and a limited number of resources. In contrast, CP-ABE is more ideal for scenarios with complex access policies and more dynamic attribute updates. In addition, recent research has focused on enhancing ABE schemes to address various security and efficiency challenges. For instance, researchers have proposed hybrid ABE schemes that combine the advantages of different ABE schemes to achieve better performance and security. Other approaches include using multi-authority ABE to address scalability issues, and incorporating techniques such as proxy re-encryption and homomorphic encryption to improve the efficiency of ABE-based systems. Overall, ABE techniques have proven to be effective in ensuring fine-grained access control in cloud-based storage systems. As cloud computing and storage continue to evolve, ABE will likely continue to play a crucial role in ensuring the security and privacy of sensitive data in the cloud.

Hashes in the blockchain-based ciphertext-policy attribute-based encryption method (BCAS) ensure data integrity by preventing unauthorized access to the ciphertext, key, and starting data. To ensure that the data are valid and have not been tampered with, the DO publishes the ciphertext's hashes and starting data to the blockchain simultaneously with the encrypted data. The system verifies the data validation and uploads the data only if the validation is successful. The DO and DR must provide the key and submit its hashes to the blockchain for verification. Once the key is generated, the DR decrypts the data and provides their hash for comparison with the original data's hash to ensure their integrity. By incorporating data hashes into the blockchain, the BCAS method provides a secure and efficient way to manage data integrity in cloud-based storage systems.

It is important to note that while blockchain technology can address specific data security and privacy concerns, it is not a silver bullet solution. Blockchain-based systems can have their security vulnerabilities and must be carefully designed and implemented. Additionally, there may be trade-offs between security and efficiency, as blockchain-based systems can be computationally intensive and require significant resources. Nonetheless, using blockchain technology in conjunction with other cryptographic techniques, such as attribute-based encryption, can offer enhanced data security and privacy protection in specific applications. The attribute-based encryption method proposed by the authors of [9] is designed for semi-honest cloud storage environments and provides a more versatile and comprehensive access control technique. The process is based on attribute-based encryption (ABE), allowing for fine-grained access control using user attributes instead of fixed identities. This approach enables data owners to specify access policies based on the attributes of the users, such as their job titles, age, or location.

Another paper [10] suggests an attribute-based SE system where the cloud server handles sophisticated computing tasks to lessen the user's computational load and increase flexibility when the access policy is altered. Since most attribute-based SE schemes employ cloud storage, data security and privacy protection issues are becoming increasingly prevalent. Users may access convenient and ample data storage services from the cloud server. However, the complexity of its security situation significantly undermines customers' confidence in it, as unauthenticated individuals can access cloud servers at will, and data protection cannot be guaranteed. Due to blockchain technology's [11] ability to enable the access and sharing of data in a free and safe manner has opened up new avenues to address these issues. For the first time, the authors of [12] highlighted the importance of storing data in the public chain and proposed a new data deletion scheme based on blockchain technology. Regardless of how poorly the cloud server behaves, the data owner can still verify the deletion result, increasing the transparency of the deletion operation. The authors of [13] then suggested a blockchain-based SE scheme combining SE and blockchain technology to guarantee fairness and minimize computation for users to search encrypted data files illegally. The authors of [14] presented a trusted SE strategy for criminal users and cloud service providers based on cloud storage. Data sharing relies heavily on attribute-based encryption, particularly encryption incorporating attributes into ciphertext. Blockchain technology can ensure and access the integrity and immutability of policy-related information. However, access control systems in dispersed networks typically leak sensitive data information. The authors of [15] suggested a traceable, efficient, and privacy-preserving attribute-based searchable encryption technique [16] in the blockchain to address the effectiveness of attribute encryption, privacy leakage, and critical abuse.

The system uses blockchain technology to guarantee the immutability and integrity of data. Searchable symmetric encryption (SSE) is also utilized, which enables selective querying of encrypted data without the risk of data leakage. To ensure the security of records on the blockchain, each network participant has a private key for signing transactions with a digital signature. If a document is altered, the signature becomes invalid, and the network is immediately alerted. To determine if a file is encrypted, the file's entropy can be calculated using tools like bin-walk, and if there is a steady increase in entropy, the file is likely encrypted.

In the system, the hospital server stores the ciphertext of a patient's electronic medical record, while the alliance chain and medical cloud server record the keyword ciphertext and the patient's pseudo-identity as the specific index. The alliance chain contains the term ciphertext and receives the trapdoor when it is generated. The nodes on the alliance chain search for keywords when a patient needs a database for the incentive system. The node on the alliance chain retrieves the security index and matches the patient's pseudo-random identification when searching for the ciphertext of the related patient. The node on the alliance chain then locates the doctor's identification on the medical cloud server to determine the hash value of the keyword ciphertext, and the patient can decrypt the electronic medical record to reveal its plaintext. Traditional data encryption methods

protect data privacy but have limitations regarding easy sharing. Ciphertext processing methods are applied to enable third-party users to perform mathematical calculations on encrypted data while protecting user privacy. The ciphertext processing methods also allow statistical or machine learning tools for privacy-preserving data analysis.

Cloud computing has significantly transformed data management and storage, providing a cost-effective, flexible, and scalable solution. However, the rapid growth of data stored and exchanged in the cloud presents security and privacy challenges. Secure data exchange is a pressing concern, with traditional encryption methods becoming inadequate. Attribute-based encryption has been proposed but still relies on trusted third parties. Additionally, data must often be locally downloaded before decryption on untrustworthy cloud servers, leading to inefficiencies. Searchable encryption (SE) offers a solution by enabling data encryption while allowing specific data retrieval without full file decryption. Nonetheless, security concerns persist, particularly regarding authentication, identity verification, and storage security.

The existing landscape highlights several gaps in addressing cloud computing's security and privacy challenges. These include the need for improved data exchange security, efficient and trustworthy cloud storage, and enhanced authentication processes. Existing solutions, such as attribute-based encryption, require further development to minimize reliance on third parties. Limitations include the reliance on trusted third parties in attribute-based encryption, the need for local data downloads before decryption, and security concerns around authentication and identity verification.

The proposed approach, which combines blockchain technology with searchable attribute-based encryption (BC-SABE), is significant for addressing these limitations. BC-SABE offers an innovative solution to securely manage IoT data by employing a decentralized blockchain system. It streamlines parameter construction, key management, and user revocation while maintaining data privacy. This approach enhances data protection, access control, and efficient revocation, making it suitable for future IoT applications.

2. Materials and Methods

This section mainly introduces the scheme's system, formal definition, and security model.

2.1. System Model

The system implements fine-grained access control for encrypted data, enabling different data users to access data according to their authorized attributes. The cloud server stores data files and encrypted keywords, and the blockchain stores the encrypted keywords' storage addresses on the cloud server. The system model shown in Figure 1 consists of several components: data owners, various data users, a cloud server, a trusted attribute authorization center, and the blockchain. The system uses cloud-based blockchain technology to ensure the immutability and integrity of data.

Attribute Authorization Center: It is completely trustworthy to the data owners and users interacting with it in the system, and responsible for setting system parameters and registering users. The key and corresponding parameters are generated by the attribute authorization center and returned to the user.

Data owner: According to the established guidelines, the data owner extracts the keyword set from the data file, encrypts the keywords using his access policy, and then uploads the data file's ciphertext and the keyword ciphertext to the cloud server. After receiving it, the cloud server stores the ciphertext and gives the data owner the storage address. The data owner then creates a reverse index relationship between both the ciphertext of the data file and the ciphertext of the keyword in the storage address of the cloud server. To complete and publish the new block, the data owner uploads a built transaction to the blockchain, the keyword ciphertext, and its storage address. The blockchain's other data consumers are in charge of the new collaboration.

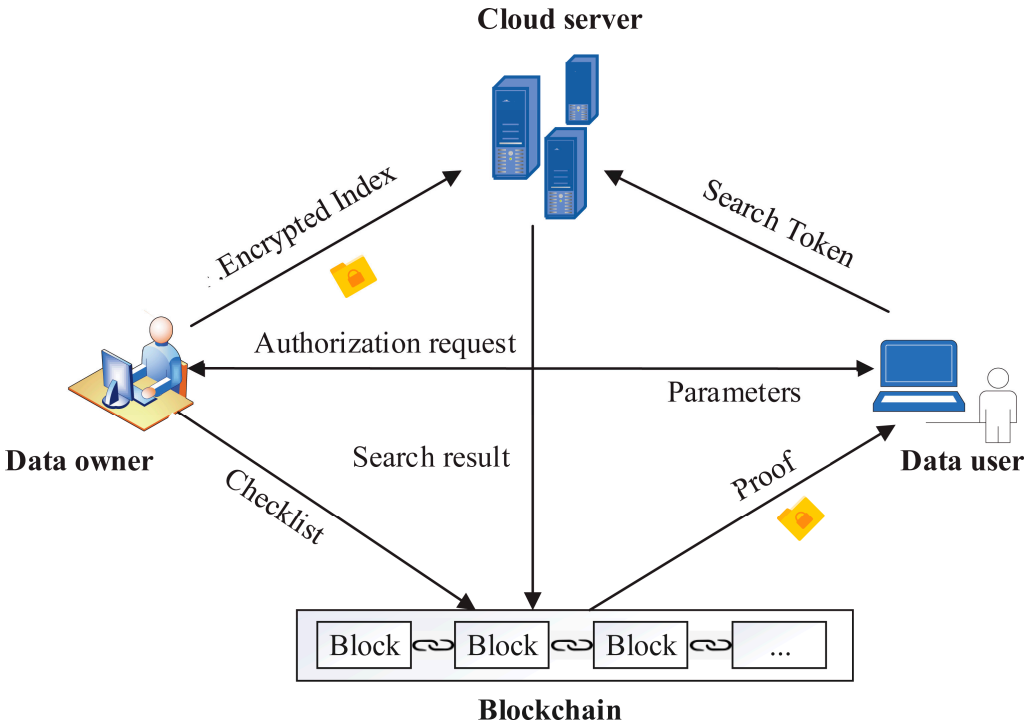

Figure 1. The proposed system models.

Cloud server: Cloud servers provide data storage services. The storage address is returned to the data owner after the cloud server stores the data file ciphertext and keyword ciphertext that the data owner provided. When the keyword search is successful, the data owner uses the address supplied by the blockchain to locally check the index connection between the data file's ciphertext and the keyword's ciphertext. After receiving a request, the cloud server will search using the ciphertext of the user's data file and respond with the results.

Blockchain: Blockchain nodes offer data search functions. The data owner creates a transaction and uploads it to the blockchain, the keyword ciphertext, and its address. When more blockchain data users receive the broadcasted block, the block is considered verified. The search algorithm is executed by a blockchain node that wishes to receive the reward when a user uploads the trapdoor as a transaction. The blockchain node gives the storage address of the keyword ciphertext to the data owner if the search is successful; otherwise, return failure is returned.

Data users: Users create search trapdoors using their private keys and desired keywords, upload the trapdoors to the blockchain as transactions, and the blockchain's nodes carry out searches using the transactions. The blockchain node provides the data owner with the keyword ciphertext storage address if the search is successful. The data owner then informs the cloud server of the ciphertext address for the data file using the index relationship. The cloud server finds the encrypted data file next, after which it offers the user access to the ciphertext of the data file.

2.2. Security Model

Keyword ciphertext indistinguishability security and trapdoor indistinguishability security of the scheme under chosen-plaintext attack are defined via the probabilistic polynomial time game between attacker A and challenger B.

i. Game 1: Keyword ciphertext indistinguishability.

In the initial phase, B runs the system to establish the algorithm output public parameters; A defines a challenge access tree T.

Stage 1: At this stage, A adaptively performs the following query of polynomial bounded degree.

Key extraction challenge: A adaptively asks B for the private key corresponding to the $R_1,\ R_2,\ \ldots,\ R_n$ attribute sets.

Trapdoor inquiry: The keyword ciphertext query adaptively asks B for the ciphertext corresponding to $l_1,\ l_2, \ldots,\ l_m$. During this process, none of the private keys that are asked for satisfies the access tree U.

Challenge: A submits two challenge keywords, x_0 and x_1, to B.

B randomly selects $\mu \in \{0,1\}$, encrypts x_μ to obtain the keyword ciphertext $J_{x\mu}$, and returns it to A.

Stage 2: A continues to initiate a series of queries corresponding to the attribute sets $R_{q+1},\ R_{q+2}$.. as in stage 1, and requires that none of the private keys obtained by the question satisfies the access tree T. Finally, A outputs $\mu' \in \{0,1\}$; if $\mu' = \mu$, then A wins game 1.

A's advantage in successfully winning this game is defined in Equation (1):

$$Adv_A^{DJQ}(\lambda) = |Qr[\mu' = \mu] - \frac{1}{2}| \tag{1}$$

If $Adv_A^{DJQ}(\lambda)$ is negligible for attacker A in probabilistic polynomial time, the scheme is said to satisfy the indistinguishability of the key-ciphertext security.

ii. Game 2: Trapdoor indistinguishability.

Let's assume that A is a polynomial-time attacker attempting to circumvent the indistinguishable trapdoor protection. Challenger B then develops a technique to overcome the DDH problem, allowing B to gain the instance. $F = \left(H_1,\ H_2,\ f, q,\ h,\ b, c, h^{bc}\right)$.

The initial phase: B runs the system to establish the algorithm to output the public parameters.

Stage 1: At this stage, A adaptively performs the following query of polynomial bounded degree.

Key extraction challenge: B runs the key generation algorithm to calculate RL_U and returns the essential RL_U to A.

Trapdoor inquiry: Given a keyword ω, the corresponding trapdoor T_ω is computed and returned to A.

Challenge: A submits two challenge keywords, ω_0 and ω_1, to B. B randomly selects $\mu \in \{0,1\}$ and uses ω_μ to get the trapdoor $U_{\omega\mu}$ and returns it to A.

Stage 2: A continues to initiate a series of queries as in stage 1, but cannot ask for information about the challenge keyword. Finally, A outputs $\mu' \in \{0,1\}$; if $\mu' = \mu$, then A wins game 2.

A's advantage in successfully winning this game is defined using Equation (2):

$$Adv_A^{USB}(\lambda) = |Qr[\mu' = \mu] - \frac{1}{2}| \tag{2}$$

If $Adv_A^{USB}(\lambda)$ (λ) is negligible for attacker A in probabilistic polynomial time, the scheme is said to satisfy trapdoor indistinguishability security.

3. Proposed Work

The blockchain's cloud-assisted attribute-based searchable encryption scheme is divided into three stages: system establishment, data encryption, and data search.

3.1. System Establishment

This stage is divided into two steps: system initialization and key generation.

i. **System initialization (*SetUp*).**

In this process, the attribute authorization center executes the algorithm to initialize the system. Input the security parameter λ, output the system's public parameter PP and the data owner's key SK.

- Generate a bilinear map, e. $H_1 \times H_1 \rightarrow H_2$, where H_1 and H_2 is cyclic multiplicative
- Hash functions I. $\{0,1\}^* \rightarrow A_a^*$, $I_1 : \{0,1\}^* \rightarrow I_1$.
- Define the Lagrange coefficient using Equation (3)

$$\Delta_{i,R}(x) = \prod_{j \in R, j \neq i} \frac{x-j}{i-j} \tag{3}$$

- where R represents a set, $i, j \in A_a^*$
- Randomly select $\alpha,\ \beta \in A_a^*$, and calculate $h^\alpha,\ h^\beta,\ e(h,\ h)^\alpha$.
- Return $QQ = \{H_1,\ H_2,\ e,\ h,\ I,\ I_1\},\ RL = \left\{e(h,\ h)^\alpha,\ h^\beta\right\}$.

ii. **Key generation (*KeyGen*).**

Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence.

During this procedure, the attribute authorization center runs the algorithm to produce the user's private key for its attribute set R_{uid}.

- Randomly select $s \in A_q^*$, and calculate $RL_{u1},\ RL_{u2}\ and\ RL_{u3}$ using Equation (4):

$$RL_{u1} = h^{\frac{\alpha+s}{\beta}},\ RL_{u2} = h^{\frac{1}{\beta}},\ RL_{u3} = h^s \tag{4}$$

- For $\forall att \in R_{uid}$, randomly select $s_a \in A_q^*$ and calculate RL_{ua} using Equation (5):

$$RL_{ua} = RL_{u3} \times I_1(att)^{s_a} = h^s \times I_1(att)^{s_a} \tag{5}$$

- Finally, the user's key $RL_U = \{RL_{u_1}, RL_{u_2}, RL_{u_3}, \forall att \in R_{uid} : RL_{ua}, RL'_{ua}\}$ is obtained, and RL_U is returned to the user.

3.2. Encryption

At this stage, the data owner invokes this algorithm to encrypt all keywords, each corresponding to the access tree defining the keyword search authority.

- Randomly select $r \in A_q^*$ as the secret value, and calculate using Equation (6):

$$D_x = e\left(h^{I(x)r}, h\right) e(h,h)^{\alpha r} and D'_x = g^{\beta r}. \tag{6}$$

- First, execute the secret sharing algorithm for each node x in the access tree U (including the leaf node t) from the root node T. To start, choose a polynomial p_x. The specific steps are:
- For each node in T, set the degree e_x of the polynomial p_x as the node's threshold value $l_x - 1$, that is, $e_x = l_x - 1$.
- Starting from the root node T, define $p_t(0) = r$, and then randomly select e_t points of the polynomial p_t to complete the definition of T_t. For other nodes x, define $p_x(0) = p_{parent(x)}(index(x))$, and randomly select e_x points to complete the definition of p_x.
- Let X be the set of leaf nodes in U. For the node $\forall x \in X$ in the set X, calculate $D_x = h_{px(0)}, D'_x = I_1(attr(x))^{px(0)}$.

Finally, the encrypted keyword is $J_w = \{D_w, D'_w, \forall x \in X :\ D_x,\ D'_x\}$. The data owner delivers the encrypted data file F and the encrypted keyword J_w to the DR, who then returns the storage address. J_w will be the data owner, and the address will be the storage address. Through marketing $U_x,\ J_w$ embeds the transaction U_x, signs it, and broadcasts it to the blockchain system, and miners record the confirmed transaction on Blockchain.

The structure of a blockchain consists of two main components: a block header and a trade. The block header has the following information: block identifier, block size, hash, and the date of the preceding block. Transactions include the following information: block producer (DO) identity ID_{DO}, block producer's signature DO and I_w, and address. The transaction U_x comprises $J_w = (J_w,\ Address.\ J_w)$.

3.3. Data Search

This stage includes trapdoor generation (Trapdoor) and keyword search (Search).

i. **Trapdoor generation**

In this process, the user uses his essential SKU and the keyword ω to be queried to generate the trapdoor U_ω.

- Randomly select $s1 \in A^*_{qa}$ and calculate U_1 using Equation (7):

$$U_1 = RL_{u_1} \times RL_{u2}^{I(\omega)+s_1} = h^{\frac{\alpha+s+(\omega)+s_1}{\beta}} \tag{7}$$

- For $\forall att \in R_{uid}$, calculate $U_a = RL_{ua} \times h^{r_1} = h^{r+r_1} \times I(att)^{ra} and\ U'_a = RL'_{ua}$. Therefore, the trapdoor generated by the keyword ω to be queried is $U_\omega = \{U_1, \forall att \in R_{uid} : U_a, U'_a\}$. Embed the trapdoor U_ω into the transaction, U_y, sign it, and broadcast it to the entire blockchain system in the form of the transaction U_y. The miners record the verified transaction $U_y = (U_\omega)$ on the blockchain.

ii. **Keyword search**

In the keyword search stage, according to the trapdoor information, U_ω submitted by the user, the node on the blockchain (also called the searcher P) executes the algorithm to search for the keyword ciphertext. During the whole search process, helpful information about data files and keywords to be searched will not be leaked to the blockchain and cloud servers. The user constructs a transaction U_y that contains his trapdoor information. The nodes on the blockchain calculate the central part of the transaction g according to the transaction U_y, embed the searched I_w into the transaction g, and sign it to the whole blockchain network. Then, they broadcast the transaction and get the reward in trade U_y simultaneously d. When the transaction g does not appear on the blockchain, the user can choose to construct a new transaction to recover the reward in the previous transaction, U_y.

The nodes on the blockchain verify whether the equation $A = D_w$ holds, where $A = \frac{e\left(\grave{D_w}, U_1\right)}{G_t}$. If the equation is established, the search is successful, indicating that the user's attribute set R_{uid} satisfies the access tree embedded in J_w and w and ω are consistent; at this time, the blockchain will store the address of $J_w Address$. This is returned to the data owner. If the equation does not hold, the search fails. There are two situations in which the search fails: the user's attribute set R_{uid} does not satisfy the access tree embedded in J_w, and the algorithm terminates; that is, the user does not have the search authority for the keyword w, or the user has the search authority for the keyword w, but the search found that w and ω are not the same. x means to visit the node in the tree U; the algorithm runs:

1. If node x is a leaf node, let $att = attr(x)$, that is, att represents the attribute associated with the leaf node x.
   - If $att \in R_{uid}$, then G_x is calculated using Equation (8):

$$\begin{aligned} G_x = \frac{e(U_a, D_x)}{e(\grave{U_a}, \grave{D_x})} &= \frac{e\left(h^{s+s_1} \times I_1(att)^{s_a}, h^{p_x(0)}\right)}{e\left(h^{s_a} \times I_1(att(x))^{p_x(0)}\right)} \\ &= \frac{e\left(h^{s+s_1}, h^{p_x(0)}\right) e\left(I_1(att)^{s_a}, h^{p_x(0)}\right)}{e\left(h^{s_a}, I_1(att(x))^{p_x(0)}\right)} \\ &= e(h,h)^{(s+s_1)p_x(0)} \end{aligned} \tag{8}$$

- If $att \notin R_{uid}$, define $G_x = \perp$.

1. If node x is a non-leaf node, for all child nodes z of node x, the result after executing the algorithm is denoted as G_z, and all values of $G_z \neq \perp$ are reserved in the set V_x.
   - If $|V_x| < kx$, it means that the attribute set of the child node of node x does not meet the threshold value of this node; then terminate and output $\perp$.
   - If $|V_x| > kx$, it means that the attribute set of the child node of node x satisfies the threshold value of this node; then randomly select l_x values of G_z from the set V_x, and calculate the G_x value in combination with the Lagrange coefficient according to Equation (9):

$$\begin{gathered} G_x = \prod_{z \in U_x} G_z^{\Delta_i, R_x(0)} = \prod_{z \in U_x} (e(h,h)^{(s+s_1)p_z(0)})^{\Delta_i, R_x(0)} = \\ \prod_{z \in U_x} (e(h,h)^{(s+s_1)} r_{Parent(z)}(index(z)))^{\Delta_i, R_z(0)} = \\ \prod_{z \in U_x} e(h,h)^{(s+s_1)_{qx}(i)\Delta_i, R_x(0)} = e(h,h)^{(s+s_1)_{qx}, R_x(0)} \end{gathered} \tag{9}$$

where $i = index(z), Rx = \{\forall z \in V_x : index(z)\}, \Delta_i, R_x$ represent the Lagrange coefficient.

2. If the user's attribute set R_{uid} satisfies the access tree U, the execution result of the algorithm is expressed as $G_t = e(h,h)^{(s+s_1)_{q_x}, R_x(0)} = e(h,h)^{(s+s_1)_r}$.

iii. **Proof of Corrections**

Calculate A using Equation (10) and B using Equation (11).

$$\begin{gathered} \boldsymbol{A} = \frac{e\left(\grave{D_w}, U_1\right)}{G_t}, \text{ verify whether } \boldsymbol{A} = \boldsymbol{D_w} \text{is established and if so, return 1.} \\ \boldsymbol{D_w} = \boldsymbol{e}\left(\boldsymbol{h}^{\boldsymbol{G}(w)\boldsymbol{r}}, \boldsymbol{h}\right) \boldsymbol{e}\left(\boldsymbol{h}, \boldsymbol{h}\right)^{\boldsymbol{\alpha r}} \end{gathered} \tag{10}$$

$$\begin{gathered} \boldsymbol{B} = \frac{e\left(\grave{D_w}, U_1\right)}{G_t} = \frac{e\left(h^{\beta r}, h^{\frac{\alpha + s + G(\omega) + r_1}{\beta}}\right)}{e\left(h,h\right)^{(s+s_1)r}} = \frac{e\left(h^r, h^{s+s_1}\right) e\left(h^r, h^{\alpha + s + G(\omega) + s_1}\right)}{e\left(h,h\right)^{(s+s_1)r}} \\ = e\left(h^r, h^{\alpha + G(\omega)}\right) = e(h,h)^{\alpha r} e\left(h^{eG(\omega)}, h\right) \end{gathered} \tag{11}$$

When the data owner obtains the storage address of $\boldsymbol{J_w}$***Address*** A, after $\boldsymbol{J_w}$, the data owner is based on address. $\boldsymbol{J_w}$ and address. The index relationship of $\boldsymbol{G}$ finds ***Address***.$\boldsymbol{G}$, and ***Address***. $\boldsymbol{G}$ returns to the cloud server. The cloud server according to ***Address***. $\boldsymbol{G}$ finds the corresponding encrypted data file and replaces the encrypted data file with the user.

4. Performance Analysis

4.1. Comparison of Functional Characteristics

This paper compares functionally with attribute-based encryption schemes [17–19] in recent years. The access control strategy mainly includes two kinds of access trees and a linear secret sharing scheme.

The comparison results are shown in Table 1. In addition, Table 1 shows that the proposed method has certain advantages in functional characteristics.

Table 1. Comparison of functional characteristics of proposed method with existing attribute-based encryption schemes [17,20,21].

Program	Access Control Policy	Searchable	Privacy Protection	Blockchain Technology
Ref. [17]	Access tree	×	×	×
Ref. [18]	Linear Secret Sharing	√	×	×
Ref. [18]	Access tree	√	√	×
Ref. [19]	Access tree	√	√	×
Proposed Scheme	Access tree	√	√	√

4.2. Comparison of Theoretical Characteristics

Running time is the time the algorithm takes during the running process [20]. The following two subsections present the complexity with respect to time and space.

i. **Comparison of the amount of calculation**

In Table 2, U_p represents the time of pairing, U_e represents the time of exponentiation, T_m represents the time of multiplication, and U_h indicates the time of hashing. T represents the time of multiplication and the inverse element action. Also, in Tables 2 and 3, $|T|$, $|U|$, and $|V|$ represent the attribute set of a user, the leaf node set of an access tree, and the minimum attribute set that satisfies the access tree, respectively.

Table 2. Comparison of the amount of calculation (|T| represents the attribute set of a user, |U| represents the leaf node set of an access tree, and |V| represents the minimum attribute set that satisfies the access tree).

Algorithm	Reference [17] Scheme	Proposed Scheme
SetUp	$3U_e$	$U_P + 3U_e$
KeyGen	$(3\|T\|+1)U_e + (\|T\|+2)U_m + \|T\|U_h + U_{inv}$	$(2\|T\|+1)U_e + (\|T\|+1)U_m + \|T\|U_h + U_{inv}$
Encrypt	$(2\|V\|+4)U_e + 2U_m + (\|V\|+1)U_h$	$U_p + (2\|V\|+3)U_e + 3U_m + (\|V\|+1)U_h$
Trapdoor	$(2\|T\|+4)U_e + U_m + U_h$	$(\|T\|+1)U_e + (\|T\|+1)U_m + U_h$

Table 3. Comparison of storage costs (|H1|, |H2|, and $|A_q^*|$ denote the lengths of elements in H1, H2, and A_q^*).

Algorithm	Reference [17] Scheme	Proposed Scheme
SetUp	$4\|H_1\|+3\|A\|$	$\|H_1\|+\|H_2\|+\|Z*\|$
KeyGen	$(2\|R\|+1)\,\|H_1\|$	$(2\|T\|+2)\,\|H_1\|$
Encrypt	$(2\|U\|+3)\,\|H_1\|$	$(2\|U\|+1)\,\|H_1\|+\|H_2\|$
Trapdoor	$(2\|R\|+3)\,\|H_1\|$	$(2\|T\|+1)\,\|H_1\|$

ii. **Comparison of storage capacity**

In Table 3, we use |H1|, |H2|, and $|A_q^*|$ to denote the lengths of elements in H1, H2, and A_q^*, respectively.

4.3. Comparison of Numerical Simulation

The numerical simulation is implemented in C using a 2.9GHz CPU and 4GB of RAM on a Linux computer using the bilinear pairing package (pairing-based cryptography library) [19]. Figure 2 and Table 4 display the experimental results.

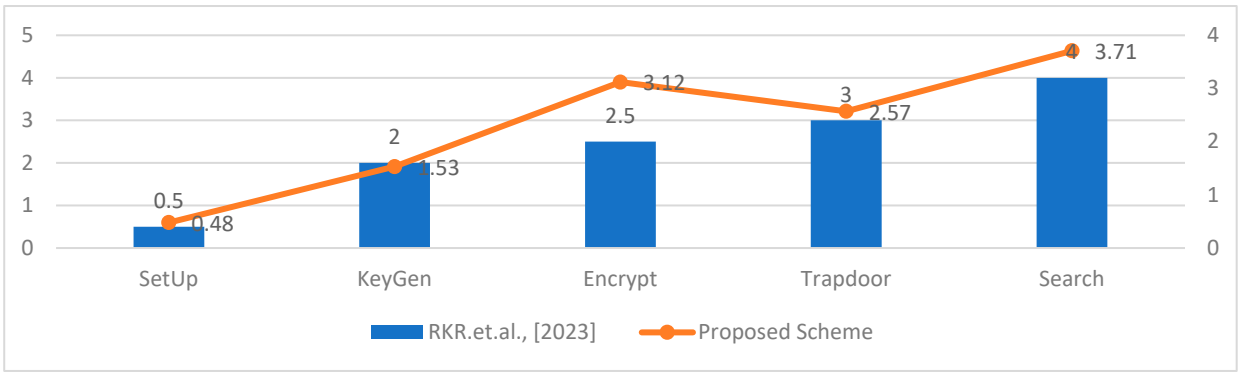

Figure 2. The running time comparison of the proposed algorithm with reference paper [17] (the number of keywords and attributes is 500 and 10, respectively).

Table 4. The running time comparison of the proposed algorithm with reference paper [17] (the number of keywords is 500 and the number of attributes is 10).

Algorithm	Reference [17] Scheme	Proposed Scheme
SetUp	0.5	0.48
KeyGen	2	1.53
Encrypt	2.5	3.12
Trapdoor	3	2.57
Search	4	3.71

Figure 2 demonstrates the proposed algorithm's time cost compared to the existing algorithm presented in the literature [17]. In the results of key generation, trapdoor generation, and search stages, the proposed system's efficiency is greater than the existing work's. Table 5 illustrates how the proposed technique performs more effectively than previous methods specified in paper [17] regarding the key generation and trapdoor generation stages.

Table 5. The running time comparison of the key generation stage and trapdoor generation stage of the proposed algorithm with reference paper [17] (the number of keywords is 500).

Algorithms	Reference [17] Scheme		Proposed Scheme	
Key Size	Key Generation Stage	Trapdoor Generation Stage	Key Generation Stage	Trapdoor Generation Stage
0	0.4	3.7	0.49	1.73
2	0.6	3.7	0.70	1.75
4	1	3.7	1.37	2.30
6	1.4	4	1.75	2.32
8	1.6	4	1.98	2.37

The investigations on the big data framework's performance that were carried out in a dynamic and heterogeneous computing environment are presented in this section. Numerous tests were conducted to determine how well the proposed methodology worked using various metrics. Three different blockchain platforms were used for the evaluations to compare them. As the most recent sample and parent source, PoW [21] was chosen. PoS [22] was also utilized as a benchmark for subsequent high-throughput research deployment on blockchain computing systems. Evaluations are typically carried out on one of the public test networks. One of the secure networks to use shared storage was the big data framework, which was introduced. A virtual evaluation was created in this work employing hardware resources, including an Intel core i5 CPU with 8 GB of RAM and 1 TB of storage to ensure a fair comparison. Multiple instances of the blocks were used to mimic the test network, and the default PoW and PoS setups were used, just like in the proposed system. Table 1 compares the performance of the suggested framework with the benchmark works. We compare the effectiveness of the proposed approach to that of current techniques like PoW and PoS to examine the data transaction rate. Due to its lengthy computation process, PoW has a relatively low transaction rate. Based on computer power, blocks are confirmed in a PoW system. Similarly, PoS require additional transaction time because stake procedures ensure the block. A highway protocol based on a flexible finality mechanism is presented to shorten transaction times.

5. Discussion

This research proposes a solution that uses blockchain- and attribute-based encryption to provide fine-grained search permission to data owners and users. The system also employs searchable encryption technology to allow the user secure access to encrypted data without transmitting critical information to the cloud server. The research includes detailed correctness and performance analyses as well as security proofs, and numerical experiments show that the proposed scheme is efficient compared to existing work.

As more data owners and users are added to the system, managing and distributing encryption keys for access control can become cumbersome and challenging to scale efficiently. The current framework may lack flexibility in data sharing. With traditional encryption methods, if a data owner wants to share encrypted data with a new user, they may need to re-encrypt the entire dataset with the new user's key, which can be resource-intensive and slow. Proxy re-encryption can address these limitations and enhance the overall effectiveness of the proposed framework. The integration of proxy re-encryption can enable more fine-grained access control and better align with the goals of the system, making it a valuable addition to the current solution.

6. Conclusions

This research introduces a blockchain- and attribute-based searchable encryption system that addresses data privacy challenges and user access control. The proposed article's unique contributions lie in its combination of attribute-based encryption, searchable encryption, and blockchain technology to create a novel system that not only addresses data privacy challenges but also offers fine-grained access control, improved search capabilities, and efficiency gains compared to existing approaches.

The experimental setup conducted in this study provides a comprehensive analysis of correctness, performance, and security. The numerical experiment results demonstrate the proposed scheme's significant efficiency when compared to existing work cited in paper [17]. These findings highlight the potential of the system to enhance the search capabilities and data privacy in blockchain-based environments.

This paper outlines a clear roadmap for future work, including the incorporation of proxy re-encryption technology. This addition will facilitate secure data sharing from electronic medical records with external users, further advancing data accessibility and privacy in healthcare settings. By exploring these advancements, this research aims to contribute to the ongoing development of secure and efficient methods for managing sensitive data in decentralized systems.

Author Contributions: Conceptualization and methodology: R.K.R.; validation and investigation: K.R.N.S. and B.K.; resources and data curation; R.D. and S.T.; writing—original draft preparation: R.K.R. and A.K.S.; writing—review and editing: S.T. and A.K.S.; supervision: K.R.N.S., A.K.S. and B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bao, Y.; Qiu, W.; Cheng, X. Secure and Lightweight Fine-Grained Searchable Data Sharing for IoT-Oriented and Cloud-Assisted Smart Healthcare System. *IEEE Internet Things J.* **2022**, *9*, 2513–2526. [CrossRef]
2. Bao, Y.; Qiu, W.; Tang, P.; Cheng, X. Efficient, Revocable, and Privacy-Preserving Fine-Grained Data Sharing with Keyword Search for the Cloud-Assisted Medical IoT System. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 2041–2051. [CrossRef] [PubMed]
3. Gao, J.; Yu, H.; Zhu, X.; Li, X. Blockchain-Based Digital Rights Management Scheme via Multi authority Ciphertext-Policy Attribute-Based Encryption and Proxy Re-Encryption. *IEEE Syst. J.* **2021**, *15*, 5233–5244. [CrossRef]
4. Zhang, H.; Yang, Z.; Yu, H. Lightweight and Privacy-preserving Search over Encryption Blockchain. In Proceedings of the 2021 7th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC), Beijing, China, 17–19 November 2021; pp. 423–427. [CrossRef]
5. Zhang, Z.; Zhang, J.; Yuan, Y.; Li, Z. An Expressive Fully Policy-Hidden Ciphertext Policy Attribute-Based Encryption Scheme with Credible Verification Based on Blockchain. *IEEE Internet Things J.* **2022**, *9*, 8681–8692. [CrossRef]
6. Chen, P.-C.; Kuo, T.-H.; Wu, J.-L. A Study of the Applicability of Ideal Lattice-Based Fully Homomorphic Encryption Scheme to Ethereum Blockchain. *IEEE Syst. J.* **2021**, *15*, 1528–1539. [CrossRef]
7. Mamta, B.; Gupta, B.; Li, K.-C.; Leung, V.C.M.; Psannis, K.E.; Yamaguchi, S. Blockchain-Assisted Secure Fine-Grained Searchable Encryption for a Cloud-Based Healthcare Cyber-Physical System. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1877–1890. [CrossRef]
8. Wang, Z.; Ma, W.; Gong, B. An Attack Scheme of RSA Encryption System with Protocol Failure. In Proceedings of the 2020 3rd International Conference on Smart Blockchain (SmartBlock), Zhengzhou, China, 23–25 October 2020; pp. 87–91. [CrossRef]

9. Yaji, S.; Bangera, K.; Neelima, B. Privacy Preserving in Blockchain Based on Partial Homomorphic Encryption System for Ai Applications. In Proceedings of the 2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW), Bengaluru, India, 17–20 December 2018; pp. 81–85. [CrossRef]
10. Yang, Y.; Hu, M.; Cheng, Y.; Liu, X.; Ma, W. Keyword Searchable Encryption Scheme based on Blockchain in Cloud Environment. In Proceedings of the 2020 3rd International Conference on Smart Blockchain (SmartBlock), Zhengzhou, China, 23–25 October 2020; pp. 1–4. [CrossRef]
11. Lin, G.; Wang, H.; Wan, J.; Zhang, L.; Huang, J. A blockchain-based fine-grained data sharing scheme for e-healthcare system. *J. Syst. Archit.* **2022**, *132*, 102731. [CrossRef]
12. Zhang, Y.; Xu, C.; Ni, J.; Li, H.; Shen, X.S. Blockchain-Assisted Public-Key Encryption with Keyword Search Against Keyword Guessing Attacks for Cloud Storage. *IEEE Trans. Cloud Comput.* **2021**, *9*, 1335–1348. [CrossRef]
13. Cui, H.; Deng, R.H.; Lai, J.; Yi, X.; Nepal, S. An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited. *Comput. Netw.* **2018**, *133*, 157–165. [CrossRef]
14. Liu, S.; Yu, J.; Xiao, Y.; Wan, Z.; Wang, S.; Yan, B. BC-SABE: Blockchain-Aided Searchable Attribute-Based Encryption for Cloud-IoT. *IEEE Internet Things J.* **2020**, *7*, 7851–7867. [CrossRef]
15. Cui, H.; Wan, Z.; Wei, X.; Nepal, S.; Yi, X. Pay as You Decrypt: Decryption Outsourcing for Functional Encryption Using Blockchain. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3227–3238. [CrossRef]
16. Ghorbel, A.; Ghorbel, M.; Jmaiel, M. Accountable privacy preserving attribute-based access control for cloud services enforced using blockchain. *Int. J. Inf. Secur.* **2022**, *21*, 489–508. [CrossRef]
17. Dodmane, R.; K. R., R.; N. S., K.R.; Kallapu, B.; Shetty, S.; Aslam, M.; Jilani, S.F. Blockchain-Based Automated Market Makers for a Decentralized Stock Exchange. *Information* **2023**, *14*, 280. [CrossRef]
18. Whaiduzzaman, M.; Mahi, M.J.N.; Barros, A.; Khalil, M.I.; Fidge, C.; Buyya, R. BFIM: Performance Measurement of a Blockchain-Based Hierarchical Tree Layered Fog-IoT Microservice Architecture. *IEEE Access* **2021**, *9*, 106655–106674. [CrossRef]
19. Awadallah, R.; Samsudin, A.; The, J.S.; Almazrooie, M. An Integrated Architecture for Maintaining Security in Cloud Computing Based on Blockchain. *IEEE Access* **2021**, *9*, 69513–69526. [CrossRef]
20. Sun, Y.; Li, X.; Lv, F.; Hu, B. Research on Logistics Information Blockchain Data Query Algorithm Based on Searchable Encryption. *IEEE Access* **2021**, *9*, 20968–20976. [CrossRef]
21. Raghunandan, K.R.; Dodmane, R.; Bhavya, K.; Rao, N.S.K.; Sahu, A.K. Chaotic-Map Based Encryption for 3D Point and 3D Mesh Fog Data in Edge Computing. *IEEE Access* **2023**, *11*, 3545–3554. [CrossRef]
22. N S, K.R.; K R, R.; Dodmane, R.; K, B.; Islam, S.M.N.; Shetty, S. Security Attacks and Key Challenges in Blockchain Technology: A survey. In Proceedings of the 2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 16–17 December 2022; pp. 1–6. [CrossRef]