Ingénierie des Systèmes d'Information
Vol. 29, No. 1, February, 2024, pp. 1-8
Journal homepage: http://iieta.org/journals/isi

BAB-SDMM: Blockchain Attribute Based Secure Data Management Model

Battula Venkata Satish Babu[1*], Kare Suresh Babu[2], Durga Prasad Kare[3]

[1] Computer Science and Engineering Department, Jawaharlal Nehru Technological University, Hyderabad, Prasad Vara Potluri Siddhartha Institute of Technology, Vijayawada 520007, India
[2] Computer Science and Engineering Department, Jawaharlal Nehru Technological University, Hyderabad 500085, India
[3] Project Delivery Lead, Deloitte Consulting Limited Liability Partnership, Buffalo Grove 60089, United States

Corresponding Author Email: vsatish.phd@gmail.com

https://doi.org/10.18280/isi.290101

Received: 9 November 2023
Revised: 18 January 2024
Accepted: 23 January 2024
Available online: 27 February 2024

Keywords:
block chain, attribute, revocation, auxiliary tree, smart contract, encryption, decryption, policies

ABSTRACT

The secure access and reliable access revocation methods of modern digital systems are based on access control mechanisms. Access policies, which are used in access control mechanisms, are very important in safeguarding security and ensuring data protection. It is evident that the protection and tamper-proofing of such policies are very important. In addition, efficient access revocation schemes are required to promptly remove access privileges when users are no longer needed or authorized. The shortcomings of existing systems in ensuring efficient, streamlined access revocation and tamper-proof protection of access control policies underscore the need for innovative solutions. In this paper, we have introduced the novel Blockchain Attribute-Based Secure Data Management Model (BAB-SDMM). Our model is the first to integrate attribute-based encryption (ABE), Attribute-Based Access Control (ABAC), and blockchain to achieve multiple security features as well as provide partial and complete revocation at the same time. The experimental results and analysis, performed using the Ethereum blockchain network, demonstrated the enhanced performance of the proposed BAB-SDMM compared to existing research works.

1. INTRODUCTION

Ensuring the secure access control of sensitive information and maintaining privacy is of utmost importance for secure data sharing. This practice effectively counters unauthorized access and provides defense against potential breaches. Furthermore, the requirement for effective revocation mechanisms to promptly revoke access privileges is apparent. This is essential for preventing unauthorized usage and access delegation of shared data, consequently encompassing the comprehensive security of the data.

Access control mechanisms are critical for ensuring secure data access within such systems. They are essential to provide privacy, enforce non-repudiation, hinder unauthorized activities, ensure auditability and accountability and offer improved revocation mechanisms. There are multiple categories of access control mechanisms available, including Role-Based, Attribute-Based, Attribute encryption based, Capability-Based, and Blockchain/Smart Contract Based Access Control.

Most of the access control mechanisms mentioned above have limitations. Some of these limitations include limited granularity, complexity in access policy management and protection, difficulties involved in access revocation, and lower operational efficiency. Attribute based access control(ABAC) is preferred over the other access control methods due to its capability to provide finer granularity in access control using multiple attributes and to resolve most the limitations mentioned above [1].

Access policies used in access control mechanisms ensure security, data protection, and compliance within organizations. It is evident that such policies can be protected and hidden using encryption and decryption mechanisms. Attribute-based encryption and decryption can be used to protect and hide access policies from malicious users. To make it tamper-proof and secure, we can use blockchain technology.

Ethereum is a decentralized blockchain platform facilitating the creation of smart contracts and decentralized applications. Ethereum primarily operates on a decentralized network known as a blockchain. Blockchain is a decentralized and digital peer-to-peer ledger that works by incorporating cryptographic principles [2], a decentralized structure, and distributed consensus, where transactions are organized into blocks and linked using cryptographic hashes, resulting in an immutable data chain [3]. Considering the problems and research gaps identified in the literature review, we have proposed a novel Blockchain Attribute-Based Secure Data Management Model (BAB-SDMM).

The BAB-SDMM is a state-of-the-art model that uniquely incorporates Attribute-Based Encryption (ABE), Attribute-Based Access Control (ABAC), and Blockchain technology to enhance security, including the hiding and securing of access policies, implementation of reliable revocation methods, and assurance of forward and backward security.

The remainder of the paper is organized as follows: Section 2 explores findings and research gaps from the literature

review, Section 3 introduces the BAB-SDMM methodology, Section 4 presents results and offers a comprehensive discussion, Section 5 discusses future scope, and finally, Section 6 provides concluding remarks.

2. LITERATURE REVIEW

The literature review section provides a comprehensive analysis of research approaches that have emerged in the combination of ABAC with blockchain technology. Additionally, it presents valuable insights derived from significant findings and research gaps within this domain.

2.1 Access policy hiding

The TrustAccess [4] method involves sending ciphertext policies to a blockchain via transactions, simultaneously storing ciphertext locally and transmitting the ciphertext address to achieve access policy hiding. These operations come at the cost of reduced operational efficiency and a lack of insight into policy updates.

Ghaffaripour and Miri [5] used smart contracts to enforce access policy encryption with ABE, ensuring policy hiding, but lacked a mechanism for policy updates and management in the blockchain.

In the research conducted by Fan et al. [6], policies are stored inside the blockchain ledger, providing user self-certification and ensuring non-repudiation; however, achieving policy hiding using ABE encryption requires the execution of four complicated operations.

In their work Wang et al. [7], applied ABE encryption to protect access policies through hiding, allowing keyword search over the encrypted data. Their methodology for policy hiding involves an additional two-time encryption process.

Ying et al. [8] proposed a policy hiding scheme that supports both partial and full hiding. However, their approach does not involve the use of blockchain and is not suitable for blockchain applications.

2.2 Revocation

Zong et al. [9] used CP-ABE along with an auxiliary binary tree for user revocation. Access policies and revocation lists are stored as ciphertext to enable both forward and backward security. However, this method introduces overhead due to updates in ciphertext and new key generation for each non-revoked user, and it lacks support for dynamic revocation and specific action revocation.

Han et al. [10] used CP-ABE scheme for hiding policy and encrypting revocation lists. In this scheme, ciphertext consists of two components: the first part is related to access policy, and the second part is associated with revocation. During a revocation request, only the second part of the ciphertext is updated. However, this operation results in additional overhead due to updated key generation for each revoked user (re-encryption), and the scheme also lacks support for specific action revocation and dynamic revocation.

Jiang et al. [11] maintained a trace list to enable direct revocation of malicious users. They applied full revocation in the event of revocation request, with the intention of revoking access for users are responsible for data leakage. However, their system lacks the capability for partial revocation and allows only direct revocation.

Yang et al. [12] proposed data sharing method that supports attribute revocation with the help of attribute authority for user attribute revocation. However, their system completely depends upon attribute authority for revocation, and their system also does not support partial revocation or indirect revocation.

In Hoang et al. [13] work, revocation is carried out through update operations. However, this scheme also requires updates to the non-revoked proof and decryption keys for existing users who possess the revoked attribute. It creates additional overhead in overall data access management.

The current literature on access policy hiding and access revocation reveals several limitations. Some approaches lack effective strategies for policy updates within the blockchain environment. Additionally, the use of Attribute-Based Encryption (ABE) for achieving policy hiding introduces complexity, necessitating multiple intricate operations, including a two-time encryption process. Apart from this, to guarantee forward secrecy after user revocation, most of the existing methods attempt to update key parameters of non-revoked users, which results in overhead and impacts operational efficiency. When it comes to revocation, most of the existing methods in the literature support only full revocation. No separate strategy has been proposed to streamline both partial and complete revocation as a single operation. Some of these schemes depend solely on attribute authorities for revocation, potentially introducing centralization issues.

3. METHODOLOGY

The primary objective was to design innovative and robust access control systems using blockchain technology, ensuring the protection of access policy rules from tampering and the establishment of effective access revocation procedures.

The BAB-SDMM model is such a model, and it's the first model to fuse ABAC, ABE, and Blockchain technology to provide a reliable platform for secure, fine-grained, and flexible data access/revocation control mechanisms.

BAB-SDMM applies the eXtensible Access Control Markup Language (XACML) framework model for designing and implementing attribute-based access control (ABAC) [14]. Figure 1 illustrates the arrangement for XACML-based ABAC.

The Policy Information Point (PIP) is implemented in blockchain technology by Object Attribute Management Smart Contract (OAMC) and Subject Attribute Management Smart Contract (SAMC) to retrieve attribute values from subjects, objects, and the environment for the Policy Decision Point (PDP). Smart contracts comprise self-executing programs that automatically execute and implement the terms of a contract whenever predetermined criteria are satisfied, without the need for intermediaries.

The OAMC is used to manage object attributes, whereas the SAMC is responsible to administrate subject attributes. Both of these smart contracts have important functions in the management of attribute information linked to objects and subjects within the blockchain system.

The Policy Administrative Point (PAP) is responsible for the management and issuing of attribute values (access policies) linked to subjects, resources, or other entities. PMC (Policy Management Smart Contract) code is employed to implement PAP, enabling the retrieval of access policies stored within the blockchain.

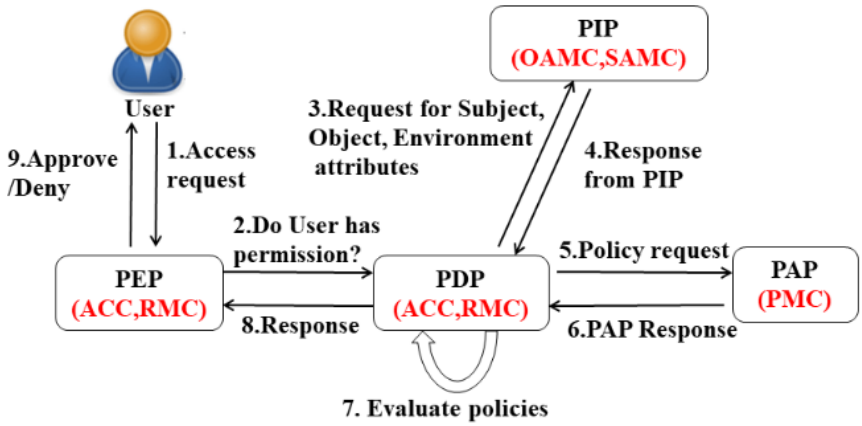

Figure 1. XACML-based ABAC

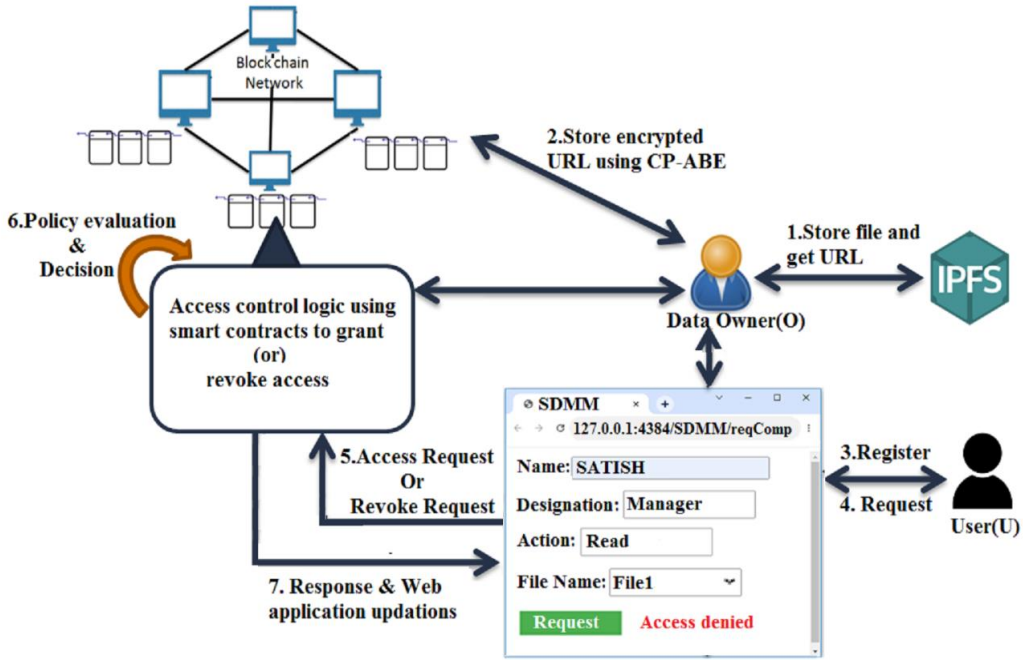

Figure 2. Blockchain attribute-based secure data management model

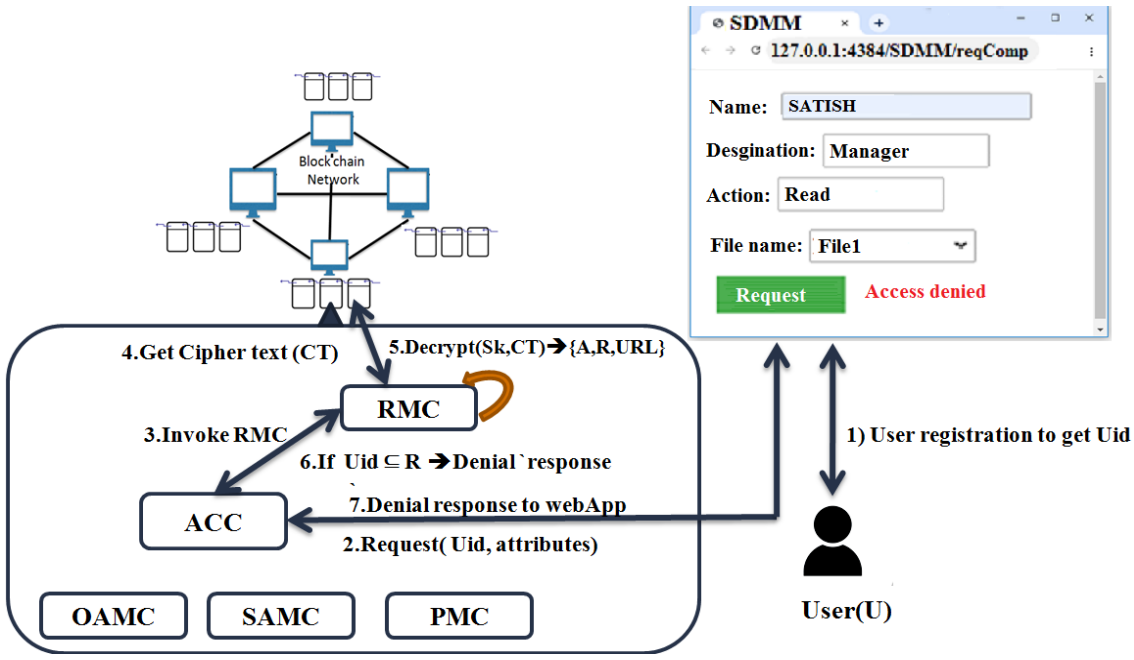

Figure 3. BAB-SDMM access denial

The Policy Decision Point (PDP) evaluates access requests by applying access control policies and attributes of the subject, resource, action, and environment, while the Policy Enforcement Point (PEP) serves as an intermediary between the subject and the resource, ensuring the enforcement of access control decisions made by PDP.

ACC (Access Control Contract) and RMC (Revocation Management Contract) are used to implement the PEP and PDP. These smart contract codes enable the evaluation of access policies and the determination of final access control decisions. Figure 2 illustrates the overall architecture of the proposed BAB-SDM model.

3.1 Initial setup

The data owner stores the file in IPFS and obtains a URL.

Subsequently, the data owner encrypts the URL utilizing CP-ABE.

a) Invokes the "setup ()" function to generate the Public Key (Pk) and Master Secret Key (Msk).

b) The data owner proceeds to define access policies in the specified format.

A={Subject1: Object1::Actions1;; Subject2: Object2: Actions2;;……}

Example: A= {Krishna, Admin: File1: write;; Ravi, Manager: File2, File3: read, write;;}

c) Create a tree (T) and empty revocation list denoted as R = { }.

d) Initiates the encryption operation.

encrypt (Pk, R, A, T, URL) → CT (Cipher text)

The data owner then saves the ciphertext (CT) in the Ethereum blockchain network using the "URLStore" contract.

Similarly, the data owner deploys the smart contracts OAMC, SAMC, ACC, RMC, and PMC within the Ethereum blockchain network. Smart contract ABI of these smart contracts are given in the Table 1:

Table 1. Smart contract ABI

Smart Contract	ABI Functions
Object Attribute Management Contract (OAMC)	❖ addAttribute(..) ❖ getAttribute(..) ❖ removeAttribute(..)
Subject Attribute Management Contract (SAMC)	❖ addAttribute(..) ❖ getAttribute(..) ❖ removeAttribute(..)
Access Control Contract (ACC)	❖ invokeRMC(..) ❖ checkAccess(..) ❖ enforce(..) ❖ updateComponent(..) ❖ requestHandle(..) ❖ compareTime(..) ❖ compareLocation(..)
Policy Management Contract (PMC)	❖ updatepolicy(..) ❖ addPolicy(..) ❖ removePolicy(..)
Revocation Management Contract (RMC)	❖ encrypt(..) ❖ decrypt(..) ❖ store(..) ❖ initTree(..) ❖ updateTree(..) ❖ check(..)

3.2 Access request denial

Initially, the user must complete the registration process on the WebApp, resulting in the assignment of a unique ID (UID).

Subsequently, the user submits a file access request via the WebApp. Example:

Request = {UID, satish, Manager: File1: Read}

Access Control Contract (ACC) receives "Request" and forwards it to RMC.

The Revocation Management Contract (RMC) then retrieves the stored ciphertext (CT) from the blockchain and carries out the decryption operation using the secret key (Sk).

a) At first it generates the secret key:

keygen (MSk, Attributes)→Secret Key (Sk)

b) Perform decryption operation:

decrypt (Sk, CT) →{R, A, T, URL}

Now RMC checks if Uid ⊆ R, then RMC sends a "Denial response" to the ACC.

The ACC transmit the response of denial to the WebApp.

The WebApp displays a message "Access denied" within the request component. Figure 3 illustrates this entire process

3.3 Access request allow

While conducting its verification process, if RMC finds that Uid ⊄ R, then it sends the access control policies (A) to the Access Control Contract (ACC), Figure 4.

Now Access Control contract (ACC) performs following operations

a) It requests "object" attributes OAMC.

b) It requests "subject" attributes SAMC.

c) Subsequently, the ACC starts Access Control Policy (ACP) evaluation and decision-making by comparing the user request against the Access Control Policies (A) and the attributes obtained from OAMC and SAMC.

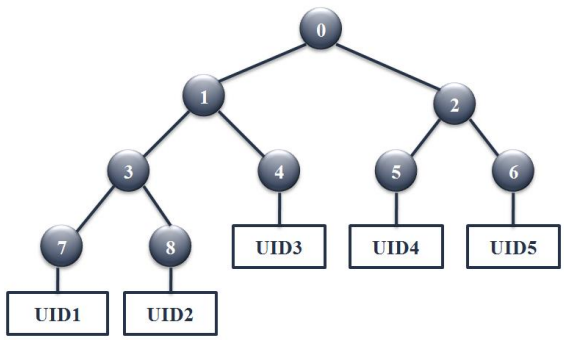

Figure 4. Auxillary Tree(T) with 5 users

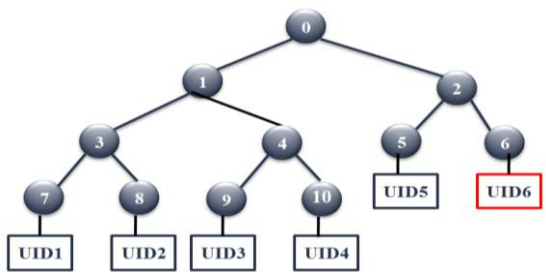

Figure 5. Auxillary Tree(T) updation

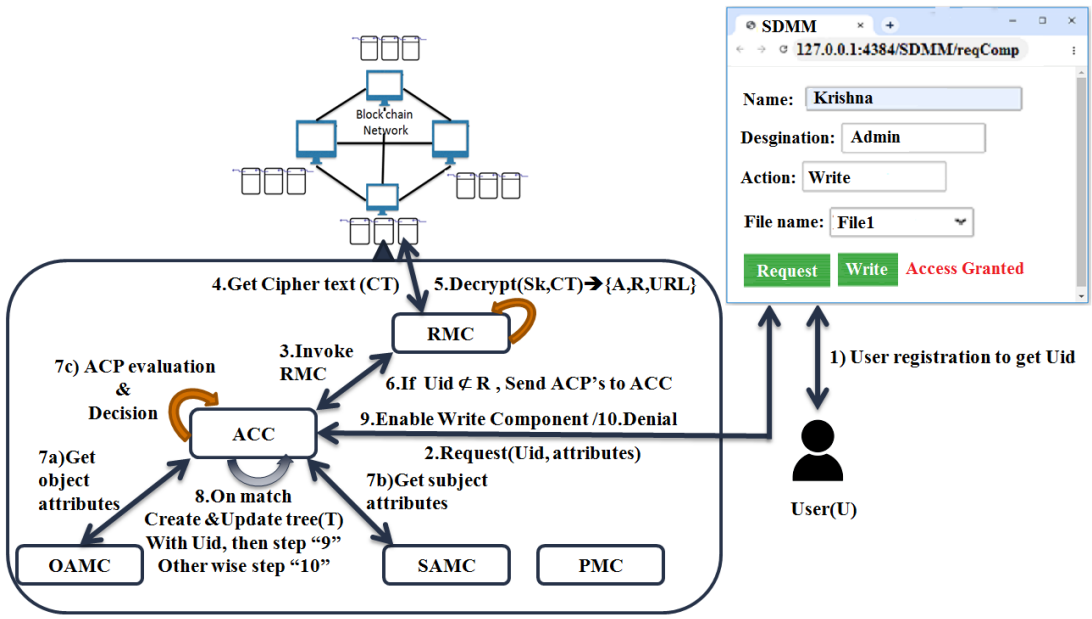

Figure 6. BAB-SDMM access grant

If the Access policies match with the attributes provided by the user, then the following steps are applied. Otherwise, a "Denial" response is the response to web application.

a) An auxiliary revocation tree (T) is and updated by adding the Uid to the tree. This auxiliary revocation tree (T) is constructed to keep track of and manage non-revoked users. For example, in Figure 4, assuming currently five users are having access to "File1".

b) If access is granted to the user with UID6, then auxiliary tree (T) would be modified as Figure 5.

Once the tree (T) has been constructed, the Access Control Contract (ACC) sends a response to enable Write Component of the web application. This entire process was represented in Figure 6.

3.4 Access revocation

Initially, the data owner sends a revocation request to the Access Control Contract (ACC). The format of the revocation request is given as:

revoke = {Uid,Action_list} or revoke = {Uid, "*" }

Upon receipt of the request "revoke", the Access Control Contract (ACC) transfers it to the Revocation Management Contract.

The Revocation Management Contract (RMC) fetches the ciphertext (CT) stored within the blockchain

RMC uses the secret key (Sk) to perform the decryption of the ciphertext (CT).

Keygen (Msk, Attributes)→Sk
Decrypt (Sk, CT) → {R, A, T, URL}

If the Uid ⊄ R, then RMC execute the following actions.

a) If Action_ list ⊆ {Specific_ Actions}, then

- update Access policies(A) to A'
- Perform encryption of {Pk, R, A', T} → CT'

b) if Action_ list == "*", then

- Remove all privileges and update A to A'
- Update R' =R ∪{Uid}
- Update T to T', Figure 6
- Perform encryption of {Pk, R', A', T'}→CT'

c) Rewrite the CT' in to blockchain storage

d) Response "Disable" to ACC and update appropriate components of webpage

Else, response "Already Revoked" to ACC and update appropriate components of web application.

Let's consider an example of updating T' during a complete revocation process in Figure 7. Assume that ACC has received a 'revoke (UID2, *')' request, which indicates to remove all permissions from the user with 'UID2'.

3.5 Problem of forward security

In order to ensure forward security using ABE scheme, the secret keys of non-revoked users are to be updated during revocation process. In our method, the revocation tree and the revocation list are sufficient to achieve user access revocation. Our method does not require the explicit removal of attributes from the secret keys of every non-revoked user.

However, just relying on a revocation tree and revocation list doesn't guarantee forward secrecy. Additionally, the immutable nature of the blockchain can assist in preventing revoked users from accessing old ciphertexts and help us achieve forward secrecy.

User revocation related information are always written in to the blockchain and can be referred during access control assessments. The proposed system verify users' revocation status recorded on the blockchain and can refuse access to revoked users, thus preventing them from decrypting old ciphertexts.

4. RESULTS AND DISCUSSION

To experiment and validate our BAB-SDMM model, we utilized a local Ethereum blockchain network of ten nodes. Ganache software is employed to establish this local blockchain network. We also developed a user-friendly web application interface using AngularJS, which acts as a bridge between users and blockchain network.

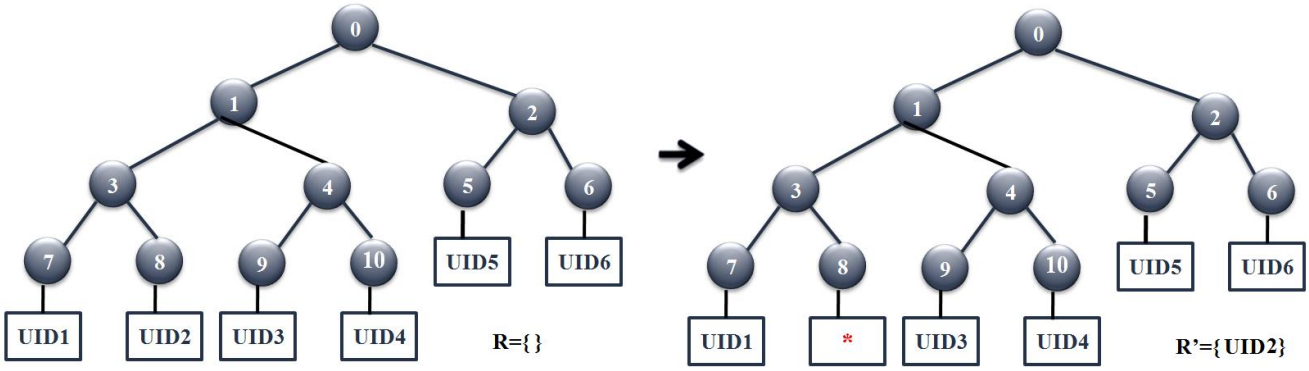

Figure 7. Auxillary Tree(T) updation during complete revocation

Web3JS integrated web application is used to create interaction between the web interface and the blockchain. Web3JS helps us to communicate with MetaMask extension. In turn MetaMask makes it easy for our WebApp to connect with the local Ganache blockchain network.

Our proposed model, BAB-SDMM, has been compared against two related models in the literature, namely the TR-AP-CPABE model [10] and the ReLAC model [9], using various metrics. Experimental results have demonstrated that our BAB-SDMM model outperforms the existing works in terms of results.

4.1 Encryption time

Both the TR-AP-CPABE model [10] and the ReLAC model [9] perform encryption twice during the setup phase as well as during the revocation process. In contrast, the proposed method requires only one-time encryption during both setup and revocation. Additionally, encryption time is further reduced, especially in the case of partial revocation. The Figure 8 below illustrates that encryption time has been reduced by 18% in comparison with existing methods.

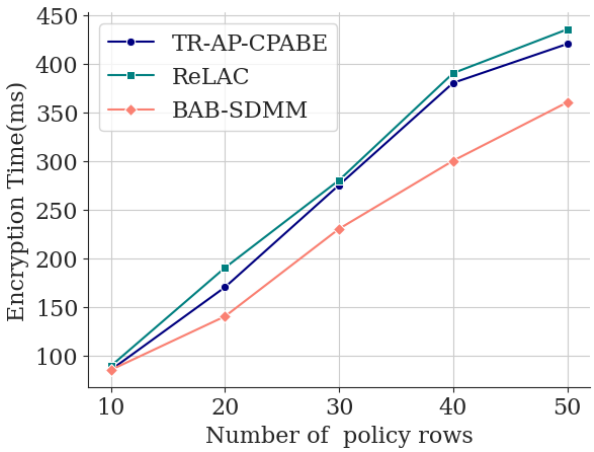

Figure 8. Encryption time analysis

4.2 Unrevoked user discovery time

Existing models use two functions, 'cover(R)' and 'path(u),' to track down users who have not been revoked.

$$\text{i.e., } k=\text{cover}(R) \cap \text{path}(u)$$

The term "path(u)" indicates the path from the root node “0” to a particular user leaf node "u". The “cover (R)” function represents the minimum set of intermediary nodes necessary to locate all users who are not listed in the revocation list R. When performing the intersection operation between the 'cover(R)' and 'path(u)' functions, if user 'u' is not part of the list 'R,' there will be one common node denoted as 'k' present in both 'cover(R)' and 'path(u)'. The Table 2 contains time complexities analysis for these operations.

Table 2. Time analysis

Model	Time Complexities Analysis	
ReLAC [9] & TR-AP-CPABE [10]	cover(R)	$O(n^2)$
	path(u)	O(log n)
	Overall complexity	$O(n^2) + O(\log n) = \mathbf{O(n^2)}$
BAB-SDMM Method		**O(n)**

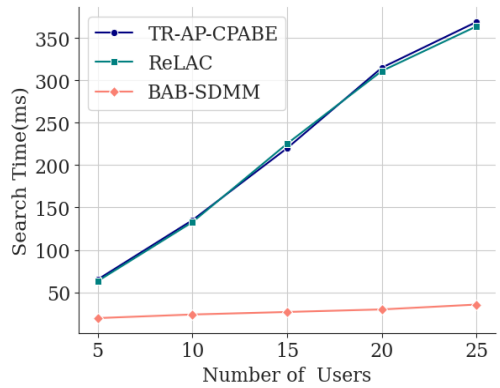

Figure 9. Unrevoked user discovery time analysis

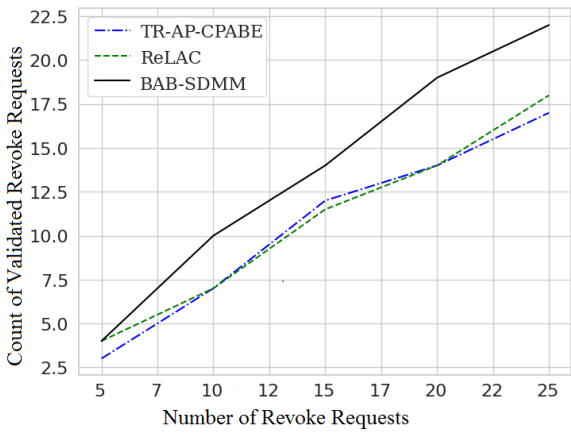

Figure 10. Revoke request verification

BAB-SDMM utilizes Breadth-First Search (BFS) traversal to determine the user revocation status, which only requires O(n) time. This underscores the efficiency of the proposed method. The comparative results for the search time analysis of the proposed method increase by 95% compared to the existing methods. These results are presented in Figure 9.

4.3 Revoke request verification

Unlike 'ReLAC [9]' and 'TR-AP-CPABE [10]' methods which only support complete revocation, our method supports both partial and complete revocation, making it a more flexible and efficient solution for revocation. The Figure 10 illustrates the revoke request verification results for both existing methods and the proposed method, considering metrics such as the 'Number of revocation requests' and 'Number of revocation requests validated.

The experimental findings reveal that the revocation-request validations have almost increased by 30%. This is due to its support for both complete and partial revocation.

4.4 Turn around revocation time

Turnaround time for revocation requests represents the duration from request submission to the final completion of the revocation process.

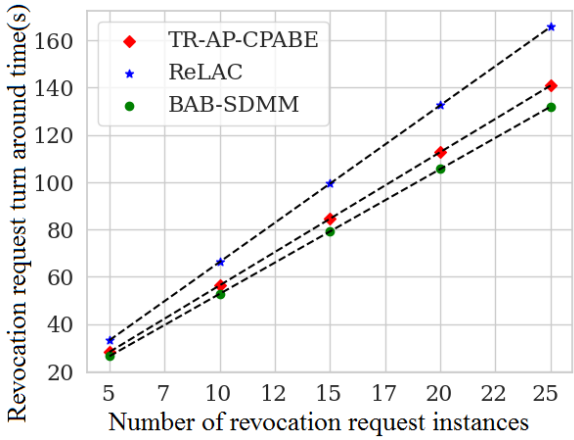

Figure 11. Turn around revocation time

This turnaround time includes the duration required for various updates, policy evaluations, encryption, decryption, as well as read and write operations on the blockchain.

With 20 access policy rows, Figure 11 Illustrates a comparison of turnaround times for revocation requests among the existing models and the proposed method 'BAB-SDMM'. The scatter plot above clearly indicates that in the proposed method, revocation request turnaround time is reduced by 14% because it completes the revocation requests with a shorter turnaround time.

4.5 Policy evaluation time

The ReLAC [9] and TR-CP-ABE [10] methods currently apply partial hiding of access policies, which means they hide only the attribute names while omitting specific attribute values. In contrast, the BAB-SDMM approach implements complete hiding of access policies. Partial access policy hiding enhances transparency, whereas complete hiding ensures strict confidentiality and security.

Figure 12 illustrates 29% reduction in policy evaluation times between the existing models and the proposed BAB-SDMM model. In this case of partial hiding of access policies, the policy evaluation time for access control is relatively faster than compared to BAB-SDMM complete policy hiding which involves both attribute names and attribute values. However, attribute values are essential for making correct access control decisions.

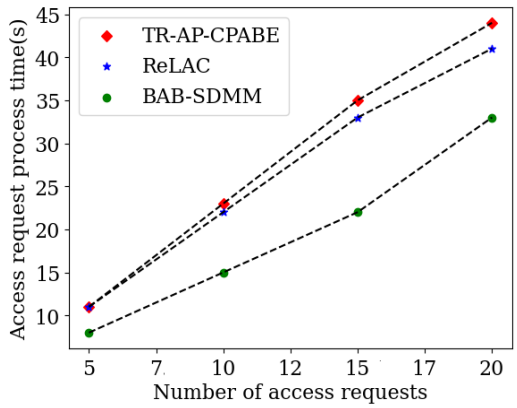

Figure 12. Policy evaluation time

Attribute values provide the specific information needed to determine whether a user should be granted or denied access to a particular resource or action. Without attribute values in access policies, it can be challenging or impossible to make accurate access control decisions.

5. FUTURE SCOPE AND RESEARCH DIRECTION

Considering the merits and limitation of proposed Blockchain Attribute-Based Secure Data Management Model (BAB-SDMM), our future research directions could focus on improving the model's effectiveness. This includes incorporating multi-granular and context-aware access control mechanisms, such as incorporating in parameters like time and location, as well as enhancing the policy evaluation time of BAB-SDMM.

6. CONCLUSION

The Blockchain Attribute-Based Secure Data Management Model (BAB-SDMM) was introduced as an innovative approach that integrates Attribute-Based Access Control (ABAC), attribute-based encryption (ABE), and Blockchain technologies to enhance the security of digital systems. Our proposed method was discussed within various scenarios, demonstrating its efficiency in making access control decisions while also supporting access policy hiding, partial and complete revocation mechanisms through the use of auxiliary trees and revocation lists. Experimental results and performed time analysis have demonstrated that our proposed method yields improved results compared to existing research works.

REFERENCES

[1] Ding, Y., Feng, L., Qin, Y., Huang, C., Dong, P., Gao, L.,

Tan, Y. (2020). Blockchain-based access control mechanism of federated data sharing system. In 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Exeter, United Kingdom, pp. 277-284. https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00060

[2] Satish Babu, B.V., Suresh Babu, K. (2020). Materializing block chain technology to maintain digital ledger of land records. In: Raju, K., Govardhan, A., Rani, B., Sridevi, R., Murty, M. (eds) Proceedings of the Third International Conference on Computational Intelligence and Informatics. Advances in Intelligent Systems and Computing, Springer, Singapore, 1090. https://doi.org/10.1007/978-981-15-1480-7_16

[3] Babu, B.V.S., Babu K.S. (2021). The purview of blockchain appositeness in computing paradigms: A survey. Ingénierie des Systèmes d'Information, 26(1): 33-46. https://doi.org/10.18280/isi.260104

[4] Gao, S., Piao, G., Zhu, J., Ma, X., Ma, J. (2020). Trustaccess: A trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain. IEEE Transactions on Vehicular Technology, 69(6): 5784-5798. https://doi.org/10.1109/TVT.2020.2967099

[5] Ghaffaripour, S., Miri, A. (2019). Cryptographically enforced access control in blockchain-based platforms. In 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, pp. 1-7. https://doi.org/10.1109/AICCSA47632.2019.9035271

[6] Fan, K., Pan, Q., Zhang, K., Bai, Y., Sun, S., Li, H., Yang, Y. (2020). A secure and verifiable data sharing scheme based on blockchain in vehicular social networks. IEEE Transactions on Vehicular Technology, 69(6): 5826-5835. https://doi.org/10.1109/TVT.2020.2968094

[7] Wang, S., Zhang, Y., Zhang, Y. (2018). A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. IEEE Access, 6: 38437-38450. https://doi.org/10.1109/ACCESS.2018.2851611

[8] Ying, Z., Jiang, W., Liu, X., Xu, S., Deng, R.H. (2021). Reliable policy updating under efficient policy hidden fine-grained access control framework for cloud data sharing. IEEE Transactions on Services Computing, 15(6): 3485-3498. https://doi.org/10.1109/TSC.2021.3096177

[9] Zong, J., Wang, C., Shen, J., Su, C., Wang, W. (2023). ReLAC: Revocable and lightweight access control with blockchain for smart consumer electronics. IEEE Transactions on Consumer Electronics. https://doi.org/10.1109/TCE.2023.3279652

[10] Han, D., Pan, N., Li, K.C. (2020). A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. IEEE Transactions on Dependable and Secure Computing, 19(1): 316-327. https://doi.org/10.1109/TDSC.2020.2977646

[11] Jiang, Y., Xu, X., Xiao, F. (2022). Attribute-based encryption with blockchain protection scheme for electronic health records. IEEE Transactions on Network and Service Management, 19(4): 3884-3895. https://doi.org/10.1109/TNSM.2022.3193707

[12] Yang, Y., Shi, R.H., Li, K., Wu, Z., Wang, S. (2022). Multiple access control scheme for EHRs combining edge computing with smart contracts. Future Generation Computer Systems, 129: 453-463. https://doi.org/10.1016/j.future.2021.11.002

[13] Hoang, V.H., Lehtihet, E., Ghamri-Doudane, Y. (2019). Forward-secure data outsourcing based on revocable attribute-based encryption. In 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, pp. 1839-1846. https://doi.org/10.1109/IWCMC.2019.8766674

[14] Patra, L., Rao, U.P., Choksi, P., Chaurasia, A. (2022). Controlling access to eHealth data using request denial cache in XACML reference architecture for ABAC. In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, pp. 1-8. https://doi.org/10.1109/GCAT55367.2022.9971895