# An Attribute-Based Access Control System for Cloud Storage with Blockchain

Vasant S. Kakade[1] and Vaibhav D. Dhore*[2]
[1]*Masters Student at Dept. of Computer Engineering and Information Technology VJTI Mumbai, India*
[2]*Assistant Professor at Dept. of Computer Engineering and Information Technology VJTI Mumbai, India*
[1]vskakade_m17@ce.vjti.ac.in, [2]vddhore@ce.vjti.ac.in

***Abstract***

*The Role-Based Access Control (RBAC) framework is a mechanism that describes entry-level access methods. Access control systems are used in computer security to control access to critical or valuable resources such as data protection, computers, computer systems, storage spaces. Attribute-based access control (ABAC) policies include topics, resources, environment, etc. included in the request. There is a set of conditions on properties that describe the characteristics. It offers new approaches based on Blockchain technology to publish policies expressing the right to access data and resources, and EVM to allow distributed transfer of such rights among users in smart computing using smart contracts for self-enforceable policy on Ethereum Virtual Machine. In this paper, the system introduces role-based access control using Smart Contract (RBAC-SC), a platform that takes into account the organizational use of the roles using Ethereum's smart contract technology. Ethereum is a free blockchain platform designed to be safe, adaptable and flexible. This led to smart contracts. Policies and rights exchanges in our proposed model are publicly visible on blockchain as blocks of encrypted storage and data will be stored on the cloud, resulting in any time the user can add policy with data or resources and with the subject. And can verify who currently have the right to access the data or resource. This measure allows distributive audit, which prevents the party from cheating on fraudulent rights imposed by the applicable policy.*

***Keywords:*** *RBAC, ABAC, CP-ABE, Cloud Storage, Blockchain.*

## 1. Introduction

Roles and titles are always used to differentiate the user's eligibility to access certain services. The role of such a mechanism is played a role-based access control (RBAC) [1] framework, which describes access controls between users and services. In RBAC, users are related to roles and are related to role services. Many organizations and companies use such a framework to apply their internal access control requirements to their computer systems. For example, if the programmer of a company has access to the backend and frontend source code, the quality assurance staff only have access to the source code above. This access control is commonly used within an organization, but it must be noted that RBAC is a versatile framework; that is, roles are often used in a trans-organizational manner. For example, students are often allowed to purchase books at discounted prices.

Some scenarios require that access rights can be transferred from a subject to another for some reasons. For instance, a user could sell its access right to another user. Another example is that the employee who performs computing on the virtual machine will give his machine to another employee if he will be going on leave for some days, and another user will get access to the same virtual machine. We propose role-based access control using Smart Contract (RBAC-SC), which, as a trans-organizational RBAC mechanism using blockchain technology and smart contracts. Attribute-based access control (ABAC) is also implemented to share data from data owners to data users having the same attributes.

*Corresponding Author

## 2. Literature Survey

Smart Contracts [1] also called crypto-contract, it is a computer program used for transferring/controlling the property or digital currents in specific parties. It does not only determine the terms and conditions but may also implement that policy/agreement. These smart contracts are stored on the blockchain and it is an ideal technology to store these contracts due to the ambiguity and security. Whenever a transaction is considered, the smart-contract determines where the transaction should be transferred/returned or since the transaction actually happened.

According to [2] this work, system aims to make attribute-based encryption (ABE) more suitable for access control to data stored in the cloud. For this purpose, system concentrate on giving to the encryptor full control over the access rights, providing feasible key management even in case of multiple independent authorities, and enabling viable user revocation, which is essential in practice. The main result is an extension of the decentralized CP-ABE scheme of Lewko and Waters with identity-based user revocation. This revocation system is made feasible by removing the computational burden of a revocation event from the cloud service provider, at the expense of some permanent, yet acceptable overhead of the encryption and decryption algorithms run by the users. Thus, the computation overhead is distributed over a potentially large number of users, instead of putting it on a single party (e.g., a proxy server), which would easily lead to a performance bottleneck.

According to Ilya Sukhodolskiy et. al. [3], their system presents a prototype of a multi-user system for access control over datasets stored in incredible cloud environments. Like other unreliable environments, cloud storage requires the ability to share information securely. Our approach provides access control over data stored in the cloud without the provider's investment. Access Control Mechanism The main tool is the dynamic feature-based encryption scheme, which has dynamic features. Using blockchain based decentralized ledgers systems provide an irrevocable log for accessibility requests for all meaningful security incidents like large financing, access policy assignment, alteration or cancellation. We offer a set of cryptographic protocols that make the secret or secret key of cryptographic operation confidential. The hash code of the ciphertext is only transmitted by the blockchain ledger.

In [4] authors study data storage and sharing schemes for decentralized storage systems and offer a framework that combines decentralized storage system interplanetary file system, Ethereum blockchain, and ABE technology. In their framework, the data owner has the ability to encrypt the shared data by distributing the secret key for data users and specifying an access policy, and this plan acquires access control precisely on the data. At the same time, on the basis of the smart contract on Ethereum Blockchain, the keyword search function is executed on the ciphertext of decentralized storage systems, which resolves this problem that the cloud server cannot return all the results searched or give false results.

In [5] they developed new cryptosystems to share encrypted data properly, which we call key-policy attribute-based encryption (KPABE). In their cryptosystem, the ciphertext is labeled with a set of properties and controls that it connects to private key access configurations that a user can decrypt the encryption. We display the utility of our product to share audit log information and broadcast encryption. Our creation supports private key providers, which subscribe to categorized identification-based encryption (HIBE).

According to Axin Wu, et.al [6] Attribution-based encryption, especially ciphertext-policy attribute-based encryption, plays an important role in data sharing. In the data sharing process, the secret key does not contain specific information of users, who can share their secret key with other users without looking for a profit. In addition, the Specialty Authority

can generate a secret key from a feature set. If the secret key is misused, it is difficult to determine whether the private key used in abuse comes from users or the specialty authority. In addition, the Access Control Structure usually leaks sensitive information in a distributed network, and the efficiency of attribute-based encryption is a hindrance to its applications. Given the efficiency of ABE, specialty protection based on privacy protection and secret key misuse, blockchain enabled and confidential-protection characteristic-based encryption is proposed.

In [7] authors proposed an approach based on blockchain technology to represent the right to access a resource and to allow the transfer of such right among users. This paper proposes a new approach based on blockchain technology to publish the policies expressing the right to access a resource and to allow the distributed transfer of such right among users. Also, the policies and the rights exchanges are publicly visible on the blockchain. Consequently, any user can know at any time the policy paired with a resource and the subjects who currently have the rights to access the resource. The problem with this approach is that the standard policy language XACML is a verbose formalism and policies can be relatively big. So, in this approach, the link to the policies and a cryptographic hash of the policies are stored in the blockchain to make it tamper proof and policies are stored on the cloud to avoid space occupation problem. Policy Create Transaction (PCT), Rights Transfer Transaction (RTT), Rights Update, Revoke are stored in the blockchain.

According to Axin Wu et. al. [8] they successfully address the issues which are the search time in most searchable attribute-based encryption schemes increases with the number of attributes, which increases the burden on the server and reduces the user experience and another problem of preventing the revoked users from decrypting the previous ciphertext by offering a clear policy feature-based data sharing plan with direct revocation of attributes and fast keyword search and hidden policy using AND gate access control in which When the ciphertext is uploaded, the access control structure does not need to be uploaded. In their scheme, the non-terminated user's private key is not required to be updated during the cancellation of direct revocation of features. In addition, a keyword search has been realized in their plan, and the search is stable with the increase in time features. Specifically, the policy is hidden in the plan, and therefore, the privacy of users is preserved. Their security and performance analysis show that their proposed plan can deal with security and efficiency concerns in cloud computing.

According to S. Khan et. al. [9], embedded power transactions in blockchain are based on their defined characteristics through the signature of many manufacturers. These signatures have been verified and customers are satisfied with the features that do not open any information that meets those features. The public and private key manufacturers have been created for these customers and using this key ensures that the support process is authorized by customers. There is no central authority required in this perspective. To protest against collision attacks, the makers are given secret pseudo-functional work seeds. Comparative analysis shows the efficiency of the proposed approach to existing people.

According to Rui Guo et. al. [10] To guarantee the validity of the EHR surrounding the block channel, he has submitted a special-based signature scheme with multiple officials, in which the patient supports the message according to the specifications, but there is no evidence that he does not have any other information. In addition, there are many officers without generating a reliable individual or a central person in order to generate and deliver a public/private key, which avoids the escrow problem and adapt to the mode of data storage distributed in the block. By sharing the secrecy of the secret pseudo-festive festivals in the authorities, this protocol opposed the attack of N-1 affiliated with officials. Under the computational Diffie-Hellman concept, they also formally demonstrate that, in relation to the specialty-signatory's enforceability and complete privacy, this specialty-based signature

scheme is safe in random decorative models. The comparison shows the efficiency and qualities of the proposed methods and methods in other studies.

## 3. System Design

The proposed system design the ABE system, hybrid cloud architecture is introduced to solve the problem. The private keys for privileges will not be issued to users directly, which will be kept and managed by the private cloud server instead. In this way, the users cannot share these private keys of privileges in this proposed construction, which means that it can prevent the privilege key sharing among users in the above straightforward construction. To get a file token, the user needs to send a request to the private cloud server. The intuition of this construction can be described as follows. To perform the duplicate check for some file, the user needs to get the file token from the private cloud server. The private cloud server will also check the user's identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file.

The system also carried out blockchain based Security Transaction during the communication for CPABE, role-based access control define the specific file access to the end user according to designed smart contracts. Once data owner uploads any file with specific attributes it will be stored into the cloud database with the encrypted format. The basic AES algorithm has views for data encryption as well as description respectively. When any user request for specific file access system reduced the number of coins according to design smart contract. During the execution, we generate each block according to our customized blockchain algorithms like hashing, mining as well as the consensus for proof of validation. The entire system has deployed on multiple virtual machines in the P2P environment. The various experiment analysis has done to evaluate the performance of the proposed system which is discussed in the result section in brief. The system can able to Defence various internal as well as external attacks and provide the execution with minimum computation time.

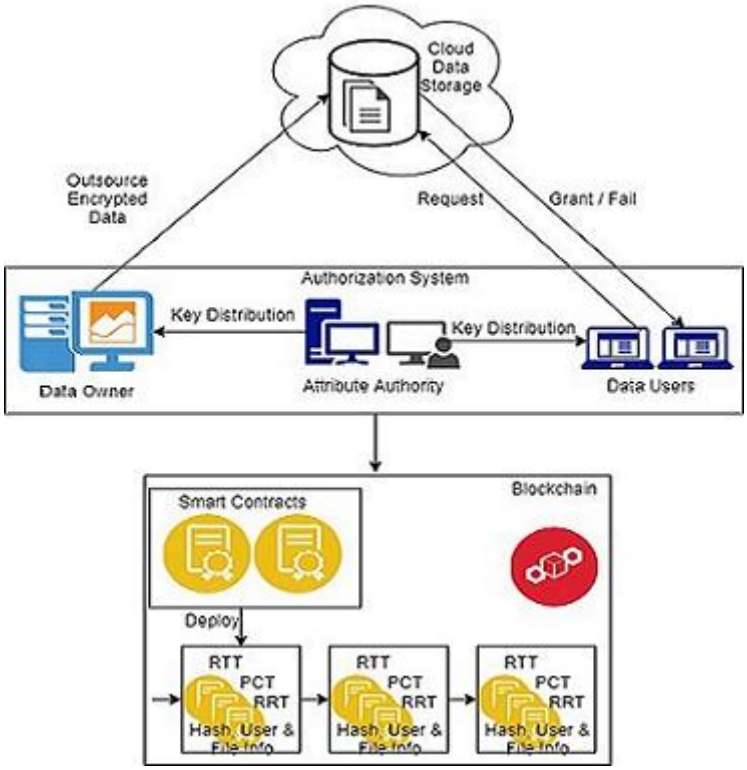


**Figure 1: Proposed System Architecture**

The participants of the system in Figure 1 and their functioning in the system is given as follows:

Data Owner (DO): A DO is an authorized user of the system, who owns data to be uploaded and shared. A DO defines an access policy for accessing the data so that only the desired users with matching attribute sets are granted permission to decrypt and get access to the plaintext data.

Cloud Service Provider (CSP): CSP is a semi-trusted environment responsible for data storage.

Attribute Authority (AA): An AA is responsible for granting a set of users and a set of attributes which we will call as a domain to users and key distribution to them. Each AA may register users in its domain and hand out the attribute keys of its domain to users. Besides creating users, attribute assignment is the main purpose of an AA. It may assign attributes to users outside its own domain, i.e. a user created by AA may receive attributes given out by the AA. We assume that each AA is semi-trusted in our system, i.e. it might be curious about the value of plaintext in the system, but has no intention of tampering with it.

Data User (DU): Data user is an authorized user who intends to access encrypted data. The user registers with an Attribute Authority and obtains one or more attribute sets. If the attribute sets satisfy an access policy associated with ciphertext, the end user will be able to get access to cipher data and by entering the valid key it can decrypt the ciphertext and get access to the plaintext.

Distributed Blockchain: The Blockchain is the distributed ledger used to represent the current state of delegated access rights in the system. Permissions to interact with the Blockchain are handled by the Administer and the Attribute Authorities. Right transfer transaction (RTT), Right revoke transaction (RRT), Hash of Transaction, User info and File info are also stored in the blockchain.

## 4. Algorithm Design

Algorithm 1: Hash Generation
Input: Genesis block, Previous hash, data d
Output: Generated hash H according to a given information

Step 1: Input data as d
Step 2: Apply SHA 256 from SHA family
Step 3: Current Hash= SHA256 (d)
Step 4: Return Current Hash

Pseudocode: Protocol for Peer Verification
Input: User Transaction query, Current Node Chain CNode[chain], Other Remaining Nodes blockchain NodesChain[Nodeid] [chain]
Output: Recover if any chain is invalid else execute the current query

Step 1: User generate any transaction DDL, DML or DCL query
Step 2: Get current server blockchain
Cchain←Cnode[Chain]
Step 3: For each

$$NodesChain\,[Nodeid, Chain] \sum_{i=1}^{n} (GetChain)$$

End For
Step 4: For each (read I into NodeChain)
If (!equalsNodeChain[i] with (Cchain))
Flag 1
Else Continue Commit query
Step 5: If (Flag == 1)
Count = SimilarNodesBlockchain()

Step 6: Calculate the majority of server Recover invalid blockchain from a specific node
Step 7: End If
End If
End For

Mining Pseudocode for valid hash creation
Input: Hash Validation Policy P[], Current Hash Values hash_Val
Output: Valid hash
Step 1: System generate the hash_Val for an ith transaction using Algorithm 1
Step 2: If (hash_Val.valid with P[])
Valid hash
Flag =1
Else
Flag=0
Mine again randomly
Step 3: Return valid hash when flag=1

**Mathematical Model**

First, we consider a

A= {A1, A2, A3……..An} each set holds the specific module activity of the system.

A1= {file uploading phase or file sending phase}
A2= {data encryption and re-encryption phase}
A3= {Share and Access control for delegates}
A4= {Revocation and proxy key re-generation}
A1 defines the first module in which user upload the multiple documents

$$\text{Data}[d] = d[k] + \sum_{k=0}^{n} (\text{a1}, \text{a2} \ldots \ldots \text{an})$$

d[k] ← {Att1, Att2……Attn} each document contains the set of attributes
keys[] ← Keygen(RandomText)
Enc[c1] [c2] ← encryption(Data, keys[])
DecData← decryption ([c1] [c2], keys[])
Role base access control for each ith user has been defined using below formula

$$\text{U[i]} \leftarrow \text{file(x)} = \sum_{\text{n=1}}^{\text{m}} \left( \text{u}_{[\text{n}]}[\text{read}, \text{write}, \text{update}, \text{delete}] \right)$$

User revocation has done using below formula
U[i] ← Revoke(F) : DataOwner

## 5. Results and Discussions

For results we here show the transactions recorded as Blocks to form blockchain as Transaction Hash of block 1 will be Previous Hash in block 2 and for block 1 Previous Hash will be 0 as it is the first block which is also called as the genesis block.

| Lable | Values |
|---|---|
| ID | 2 |
| Transcation Hash | 0x00000a327782b1af9ae034ab79cf4c69c6e814ac1045de67172fc89a9ffc9657 |
| From Owner | 0x1f76581a05a6dde25df7e53149c4cced333275ea |
| To User | 0x9b7248e3245cb391227e23db747c0b6880d4fe61 |
| File Name | Amazon S3.txt |
| Action | FileShare |
| Timestamp | Fri Jul 12 14:54:04 IST 2019 |
| Time Duration | 30#Day |
| Owner Cost | 55 |
| Transfer Cost | 0.003 |
| Nonce | 2763823 |
| Previous Hash | 0x0000001df555d21de053af9abf80ab3097b7b5af65f02cb72183e69405a60cef |

| Lable | Values |
|---|---|
| ID | 3 |
| Transcation Hash | 0x0000094f4913c8dea91e6d1568126e0850d4758cd3928c54ce090f15727b2eb9 |
| From Owner | 0x1f76581a05a6dde25df7e53149c4cced333275ea |
| To User | 0x9b7248e3245cb391227e23db747c0b6880d4fe61 |
| File Name | Test.txt |
| Action | FileRevoke |
| Time | Fri Jul 12 15:55:30 IST 2019 |
| Time Duration | 0 |
| Owner Cost | 0 |
| Transfer Cost | 0 |
| Nonce | 3658019 |
| Previous Hash | 0x00000a327782b1af9ae034ab79cf4c69c6e814ac1045de67172fc89a9ffc9657 |

**Figure 2: Transactions recorded in blocks to form Blockchain**

As shown in Figure 3 two blocks are shown in that Transaction Hash in block 2 is Previous Hash in block 3. In block transaction is recorded in which From Owner is sender address and To User is the receiver address, then File Name, Action performed which will be File Share or Revoke, Timestamp is time at which transaction is performed, Time Duration for what time period access to file is given and according to that Owner Cost, Transfer Cost paid to get access is recorded and for Revoke transaction Time Duration, Owner Cost, Transfer Cost value will not be there, and Nonce of the transaction is also calculated. And in this work Proof of Authority will be used as in our System Authorities are present which will be there to grant or reject the request so we used Proof of Authority in our work.

For the system performance evaluation, calculate the matrices for accuracy. The system is executed on java 3-tier architecture framework with INTEL 2.4 GHz i3 processor and 4 GB RAM with the distributed environment. The below figure 3 shows the time required for a consensus algorithm to validate the blockchain in 4 nodes. The x-axis shows the size of blockchain and Y shows the time required in milliseconds with respective 4 nodes.

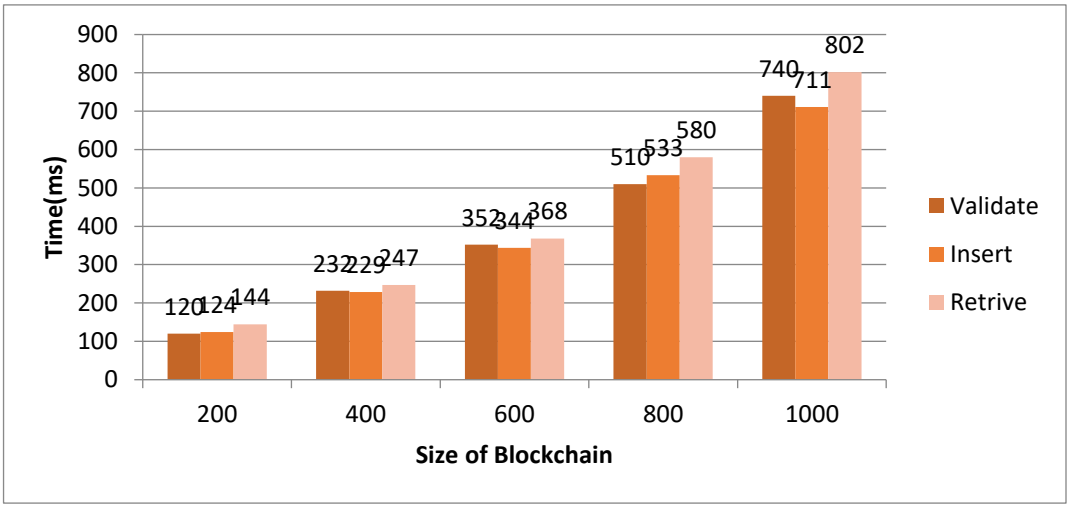


**Figure 3: Time required (in milliseconds) for the complete transaction with different records blockchain using 4 data nodes in the P2P Network**

In the second experiment, we evaluate the proposed system with smart contract validation by consensus algorithm in a different number of peer to peer nodes.

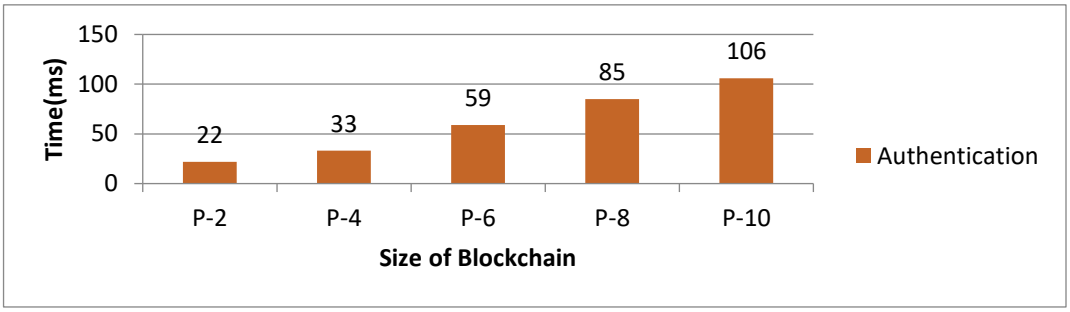


**Figure 4: Time required for smart contract validation with different no. of P2P network in the blockchain.**

In the third experiment evaluate a number of variation taken by algorithm from proposed SHA value. Basically, this experiment has done to evaluate the proposed hash string is valid or not according to a given mining policy. In many times when the system generates SHA code for given transactional data it never fulfills the mining policy. To fulfill the proposed mining policy according to given scenario mining to generate the multiple variations on a given string. The below figure 5 shows numbers of time taken in milliseconds to generate the valid SHA string for a specific transaction.

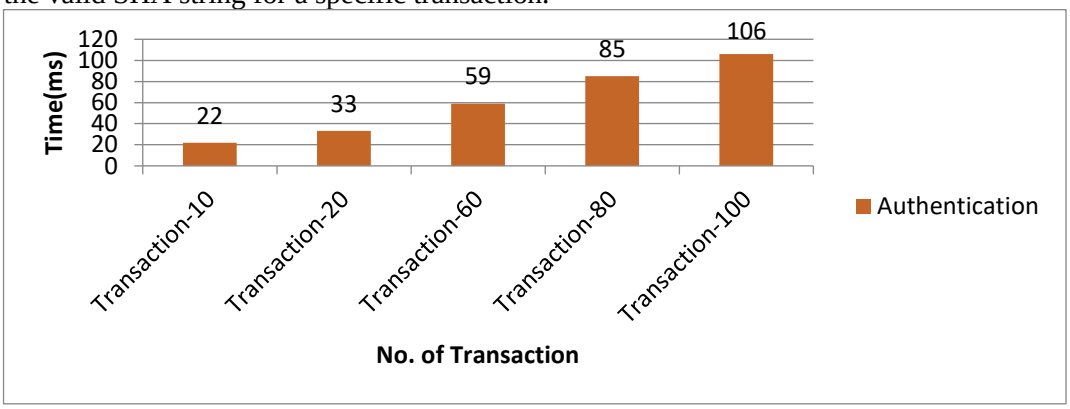


**Figure 5: Time required for mining**

## References

[1] "Smart Contracts," http://searchcompliance.techtarget.com/definition/smart-contract, 2017, [Online; accessed 4-Dec- 2017]

[2] M. Horváth. "Attribute-based encryption optimized for cloud computing." In International Conference on Current Trends in Theory and Practice of Informatics 2015 Jan 24 (pp. 566-577). Springer, Berlin, Heidelberg.

[3] S. Ilya, and S. Zapechnikov. "A blockchain-based access control system for cloud storage." Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018 IEEE Conference of Russian. IEEE, 2018.

[4] S. Wang, Y. Zhang, and Y. Zhang. "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems." IEEE Access 6 (2018): 38437-38450

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Secur., 2006 pp. 89-98.

[6] Axin Wu, Y. Zhang, X. Zheng, R. Guo, Q. Zhao, and D. Zheng: "Efficient and privacy-preserving traceable attribute-based encryption in blockchain." Annals of Telecommunications (2019): 1-11.

[7] D. D. F. Maesa, P. Mori, and L. Ricci. "Blockchain Based Access Control." In International Federation for Information Processing 2017 by Springer International Publishing AG 2017.

[8] Axin Wu, D. Zheng, Y. Zhang and M. Yang: "Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing." Sensors 18.7 (2018): 2158.

[9] S. Khan, R. Khan. "Multiple authorities' attribute-based verification mechanism for Blockchain microgrid transactions." Energies.2018 May; 11(5):1154.

[10] R. Guo, H. Shi, Q. Zhao, and D. Zheng "Secure attribute-based signature scheme with multiple authorities for Blockchain in electronic health records systems." IEEE Access 776.99 (2018): 1-12.