Efficient Distributed Access Control Using Blockchain for Big Data in Clouds

Oussama Mounnan
Oscars Laboratory, ENSA
Cadi Ayyad University
Marrakech, Morocco
e-mail: oussama.mounnan@gmail.com

Anas Abou elkalam
Oscars Laboratory, ENSA
Cadi Ayyad University
Marrakech, Morocco
e-mail: elkalam@hotmail.fr

***Abstract*—Big Data is a growing concept, offering us opportunities, that were not available before in many fields. However security and privacy issues are magnified by large amounts of heterogenous data. Ciphertext-Policy Attribute Based Encryption is a promising cryptographic primitive for the security of cloud storage system, which can bring fine-grained access control. The blockchain is a distributed ledger that records transactions in a secure, flexible, verifiable and permanent way. In this paper, we propose a distributed, scalable and fine-grained access control scheme with efficient decryption for the Big Data in clouds. Blockchain technology is used to manage identities and provide the authentication, store and execute a smart contract that incorporates the contextual and detailed access policy defined by the data owner, which is triggered by an access requester, that gives data owners the sovereign right to effectively manage their data sets and manage the policy. We also used the ciphertext-Policy Attribute Based Encryption scheme for supporting the efficient decryption outsourcing as another security layer for managing the policy. The analysis shows that our scheme is correct, complete, secure and efficient.**

Keywords-access control; encryption; blockchain; cloud; CP-ABE.

I. INTRODUCTION

Nowadays, the increase and the proliferation of large amounts of heterogenous data, also known as Big Data, generated in many fields, such as agriculture, business, finance/banking [1], education, medicine and healthcare, provide us with opportunities which did not exist before [2]. This technology opens various doors of another era of innovation and production. However, these heterogenous data are continuously increasing and pose evident challenges, which necessitate more flexible data processing tools and platforms, in order to find useful information in data [3]-[5].

The continuously increasing exchange of sensitive data has made the security of Big Data compulsory. These data are considered as a capital in its own right, and its potential is multiple. Indeed, some companies are literally creating new data-centric activities (monetizing data) while others are optimizing their supply chains or maintaining their equipments. The data is thus becoming generalized and used in all processes. Hence, these data are a valuable asset in today's economy. That is what makes us focus on its security and privacy. The latter is becoming an important issue, especially, when we deal with data in distributed cloud storage. Cloud Security Alliance (CSA) published a document that lists the top ten challenges for protecting Big Data systems [6], and granular access control is one of the stated and most critical ones.

Blockchain technology makes use of cryptography in multiple different ways, for wallets, transactions, security, and privacy-preserving protocols. It is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree). In addition, it is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way [7] [8]. The blockchain is a simple yet ingenious way of passing information from A to B in a fully automated and safe manner. The transmitted information or transactions are grouped in blocks. Each block is verified by thousands, perhaps millions of computers distributed around the net. The verified block is added to a chain, which is stored across the net, creating not just a unique record, but a unique record with a unique history. Falsifying a single record would mean falsifying the entire chain in millions of instances. That is virtually impossible. Bitcoin uses this model for monetary transactions, but it can be deployed in many others ways [9].

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is considered as one of the most applicable technologies to achieve fine-grained data access control in the cloud environment [10]. In a CP-ABE scheme [11][12], each user's key is associated with attributes and each ciphertext is related to an access policy, thus data owners can determine the access policies for their own data and control them directly. If a user's attributes satisfy the access policy in the ciphertext, the user can decrypt the ciphertext correctly. Moreover, CP-ABE has a complementary structure called key-policy Attribute Based Encryption (KP-ABE) [13], in which each user's key is associated with an access policy and each ciphertext is related to attributes. The application of CP-ABE to the cloud environment can bring data leakage prevention and access control together, which

 ISBN: 978-1-61208-719-1

are essential requirements for Big Data security and privacy [14].

Most access control solutions adopt a centralized architecture. They outsource the control of data to trusted third parties, which prevents the user from controlling his own data. This can cause problems of ethics and confidentiality. Unfortunately, when we share our information with third parties, we immediately lose control and ownership. Our new scheme executive breaks this custom and gives people what belongs to them in a fair way. In fact, we believe that Big Data needs a new access control framework that meets its specific requirements and features, allowing users to control their own privacy. This "change" will require rethinking access control technologies and creating a new solution that addresses the security and privacy requirements of Big Data. Hopefully, we are on the threshold of a new phase of decentralization, which has resulted in the emergence of a new technology, known as blockchain, that could transform fundamentally our notions of centralized authority.

In this article, we take advantage of the consistency guarantees provided by this promising technology. Our solution consists in presenting a lightweight and privacy respectful access control scheme based on the emergent blockchain technology, to ensure access control with a strong guarantee of user anonymity and data privacy in the context of Big Data. Blockchain technology is used to manage identities and provide the authentication, store and to execute a smart contract that incorporates the contextual and detailed access policy defined by the data owner, which is triggered by an access requester, that gives data owners the sovereign right to effectively manage their data sets and manage the policy. It is also used as a new multi-authority ciphertext-Policy Attribute Based Encryption scheme for supporting the efficient decryption outsourcing.

The rest of this paper is organized as follows. Section II defines the Background and preliminaries. In Section III, We expose our system model by explaining its principle and reveal our model architecture. Afterwards, Section IV presents the features analysis of our scheme. Finally, Section V concludes this paper.

II. BACKGROUND AND PRELIMINARIES

In this section, we provide a few technical backgrounds that will help ensure better understanding of our proposed work.

A. Big Data

The quantitative explosion of digital data has forced researchers to find new ways of seeing and analyzing the world. It's about discovering new orders of magnitude for capturing, searching, sharing, storing, analyzing and presenting data. So, the "Big Data" was invented as solution by the giants of the web, designed to allow everyone to access real-time databases giant. It aims to offer a choice to the classic solutions of databases and analyses.

1) Storage and analyze

Clearly, one of the biggest Big Data challenges is to store and analyze all the information. Most of this data is unstructured (documents, photos, videos and audio files…), and are difficult to analyze it. To manage the constant increase of data, companies use different technologies. In terms of storage, converged and hyperconverged infrastructures and software-defined storage make it easy to scale hardware. Technologies such as compression, deduplication or tiering also reduce the space required and the costs of Big Data storage. With regard to management and analysis, companies use tools such as NoSQL, Hadoop, Spark, and other Big Data analytical software, or artificial intelligence and Machine Learning to find the insights they need.

2) Validate the data

Data validation is also one of the major Big Data challenges. Many companies receive similar data from different systems, and this data is sometimes contradictory. Businesses can allocate a group of people to monitor data, and define rules and procedures. They can also invest in data management solutions designed to simplify governance.

3) Secure Big Data

Security is also an important concern in the field of Big Data. Business data can be attractive to hackers. However, very few companies use additional security measures for their data directories. Some of the most popular additional measures include access and identity control, data encryption, and data segregation.

4) Privacy Big Data

The confidentiality of information is about how the data is stored and how it is collected. The theft of data when transferring data via the Internet is a serious problem of data protection. Like today, the most difficult task is to secure sensitive data such as government data, medical data, space and research statistical data and military data.

B. Cloud Computing

Cloud computing has different service models, which are divided into three categories: (1) IaaS, which allows users to take advantage of the infrastructure without mentioning the hardware running behind it; (2) PaaS, which builds on IaaS and provides clients with access to basic operating software and optional services to develop and use software applications without software installation; and (3) SaaS, which enables clients to use software applications without having to install them on their personal computer, by offering these as a service through the Internet [15]. We can categorize cloud computing consistent with the deployment model into: (1) a public cloud, in which the resources are sold or rented to the public by the service provider, who is at the same time is the owner; (2) a private cloud owned or rented by an organization; (3) community

 ISBN: 978-1-61208-719-1

clouds, in which some closed communities share the same cloud resources; and (4) a hybrid cloud, which has the characteristics of two or more deployment models [16]. Several features are available in cloud computing, for example: on-demand broad network access, self-service, measured service, resource pooling, and rapid elasticity. Self-service means that the customers can manage and request their own resources. On the Internet or in private networks, the services offered are known as broad network access. In pooled resources, the customer draws from a pool of computing resources, usually in a remote data center. The services can be scaled larger or smaller, and customers are billed according to the measured use of a service [17].

C. *Blockchain*

In 2008, a person or group of persons known under the name of Satoshi Nakamoto published a paper [18] dealing with a new decentralized peer-to-peer electronic cash system. This paper introduces the blockchain as a new data structure to store financial transactions, as well as an associate protocol to ensure the validity of the blockchain in the network.

1) *Definition*

The blockchain is a technology of storage and transmission of information, transparent, secure, and functioning without a central control body (definition of Blockchain France). By extension, a blockchain is a database that contains the history of all the exchanges made between its users since its creation. This database is secure and distributed; it is shared by its different users, without intermediaries, which allows everyone to check the validity of the chain.

Blockchain technology utilizes cryptography, namely public-key cryptography, also known as asymmetric cryptography, that is exposed in Figure 1, as a means of ensuring transactions are done safely, while securing all information and storages of value. Therefore, anyone using blockchain can have complete confidence that once something is recorded on a Blockchain, it is done so legitimately and in a manner that preserves security.

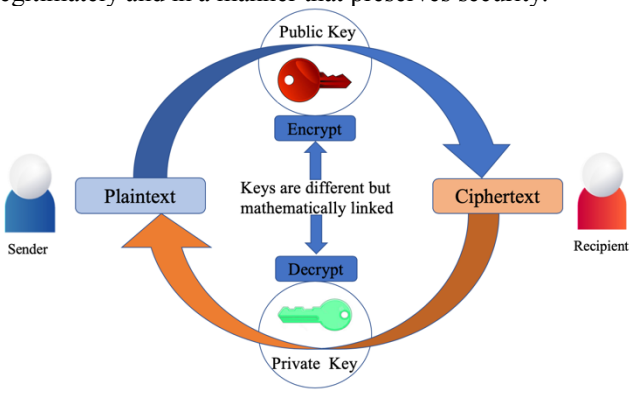

Figure 1. Public Key Cryptography

There are public blockchains, open to all, and private blockchains, whose access and use are limited to a certain number of actors. A public blockchain can therefore be likened to a large public accounting book, anonymous and unfalsifiable. As the mathematician Jean-Paul Delahaye [19] writes, one must imagine a very large notebook, which everyone can read freely and freely, on which everyone can write, but which is impossible to erase.

2) *Wallet*

Every user owns at least one wallet that stores his credentials, addresses and the transactions related to them. It contains all the keys needed to register and identify his resources, sign his transactions, ask for access.
The main functionalities of a wallet are: 1) generating keys ”secrete and public” and addresses” is a cryptographic identities of users, An address is basically the hash of an ECDSA [20] public key and a user in possession of the corresponding private key is said to own the address”. 2) Transforming the access control policies to a transactions and broadcast those last ones to the network. 3) validating received transactions from the network .

3) *Proof of work*

In the principle of blockchain, it is necessary to obtain the consensus of the majority of the actors in the network. And for that purpose, several methods exist. The most widespread cryptocurrency is called Proof of Work (PoW) [21]. This process was theorized in 1992, well before the arrival of Bitcoin and was implemented for the first time with HashCash, a solution that was to limit the proliferation of spam, but was never adopted on a large scale.

In the case of crypto-currencies, the miners, who make their computing power available to the network, must carry out energy-intensive operations. They must "chop" all the transactions to add to a block, as well as the hash of the previous block by respecting various constraints set by the level of difficulty requested by the network. The first step in in finding the solution to this problem diffuse it to the rest of the network, which verifies it.
Example:

A processor is asked to produce a proof of work consisting of coding a variation of "Hello, world!" using the SHA-256 hash function to find a fingerprint that starts with 4 zeros. The variation is to add a number at the end of the string. The processor will have to make 33,681 attempts to succeed.

4) *Concept of Blockchain*

In his paper, Nakamoto describes the blockchain as a database modeled by a linear sequence of blocks, each one containing cryptographic hashes corresponding to the previous and current block to ensure continuity and immutability. Bitcoin uses the blockchain to store financial transactions and contracts.

 ISBN: 978-1-61208-719-1

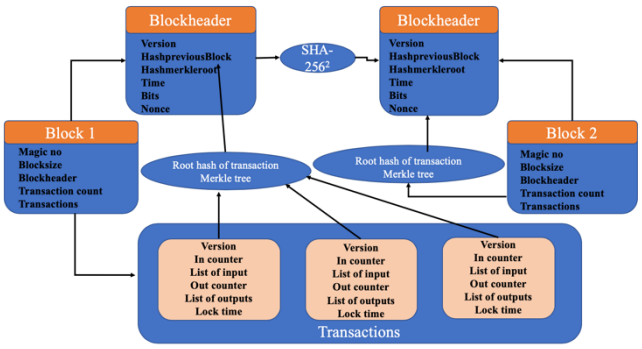

Figure 2. Blockchain infrastructure

The Merkle root of all transactions is included in the block header and then used as input for the next block in the chain. The chaining method used in Bitcoin ensures the immutability by using the hash of the previous header block hash in the current block. The header includes the root hash of the Merkle tree [22] of all transactions in the block. This way transactions cannot be changed without changing the root Merkle hash and then invalidating the block. Due to the way the blockchain is built, fork chains can append with different valid blocks storing different transactions. The Bitcoin protocol resolves this issue by selecting the longest blockchain as the correct one. Note that due to this choice, even after being included in a valid block, transactions can be considered valid only after a subsequent block has been calculated and successfully included in the blockchain by the majority of the network.

The blockchain data structure can be considered outside of its application in Bitcoin, as a generic decentralized secured data storage structure. It is possible to use any data payloads other than transactions as parts of the block. The block is then divided in two parts, (a) the block constants and header and (b) the data payloads.

A single modification in one payload of a block will change its Merkle root hash value, and then invalidate it. This solution thus provides secure and reliable storage distributed among all peers in the network. Note that this implies that the complete blockchain and all data linked to it must be duplicate on all peers.

5) *Blockchain authentication and identification*

New companies have now begun to harness the potential of the blockchain and develop a variety of services using the technology. The center of blockchain authentication would be a blockchain ID. This ID is essentially a block of data on the chain that can be both verified by any third and can display necessary information such as date of birth. The secret to this verification is the ECDSA (elliptic curve digital signature algorithm). When adding an ID to the blockchain, an identification issuing service binds a public key by default and then transfers ownership of the private key to the user. This allows the user, and only the user, to sign a signature that can be verified against the public key stored in the blockchain.

This identification of a user would serve as a decentralized source of authentication. It would essentially be a single-sign-on portal that can be accessed by any app while not being owned by any single entity. A protected app would only have to request a digital signature and an ID from a user requesting access. The app could then verify that the signature is valid and that the user's ID verifies who they say they are.

6) *Smart contract*

The Smart contract [23] is a computer protocol that allows the automatic execution of contracts whose clauses have been defined programmatically. An electronically registered contract on a distributed register (DLT) [24] cannot be altered, destroyed or contested.

One of the best things about Blockchain is that, because it is a decentralized system that exists between all authorized parties, there is not necessary to pay intermediaries, which saves money, time and conflict. Blockchains are undeniably, faster, cheaper and safer than traditional systems. These are some of the reasons why banks and governments are turning to them. In Figure 3, there are some use cases of smart contract.

Contracts can be converted into computer code, stored and replicated on the system and supervised by the network of computers running the blockchain.

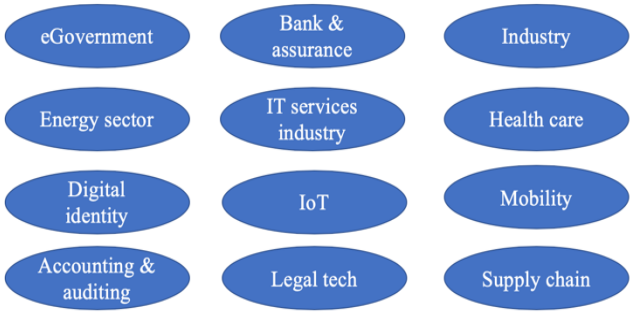

Figure 3. Use cases of smart contracts

a) *Characteristics of smart contracts*

Smart contracts have the following characteristics:

- they're self-verifying due to automated possibilities;
- they're self-enforcing when the rules are met at all stages;
- they're tamper-proof, as no one can change what's been programmed.

b) *Advantages*

Autonomy: Thanks to the smart contract, it is not necessary to rely on a broker, a lawyer or other

intermediaries to confirm. Moreover, this also strikes the danger of manipulation by a third party, because the execution is managed automatically by the network, rather than by one or more individuals possibly with bad intentions which can deceive you.

- ***Trust:*** Data is encrypted on a shared ledger. There is no way anyone can say that they lost it.
- **Backup:** The data is duplicated many times, there is no risk of being lost, With the Blockchain, the data is in several nodes in various places in the world.
- ***Security:*** Document encryption keeps data safe. There is no hacking. In fact, it would take an abnormally intelligent hacker to crack the code and infiltrate.
- ***Speed:*** for manual contract processing you have to spend more time. Smart contracts use software code to automate tasks, reducing work hours in no time.
- ***Savings*** :Smart contracts save money by eliminating the presence of an intermediary.
- ***Accuracy***: Automated contracts are not only faster and less expensive, but also avoid errors that result from manually filling a bunch of forms.

D. Attribute based Encryption ABE

In traditional cloud storage, attribute-based encryption technology [25] can achieve fine-grained access control over data. In this technique, attributes are used instead of identities. Data owner can assign the users groups that can access the data by setting an access policy. Only the users whose attributes set meet the access policy can access the data. Since attribute-based encryption technology was proposed, many research works have been done in many ways driving by actual needs, and have achieved many significant research results. For example, in a practical application, if the users attributes change, the corresponding users secret key must also be changed accordingly, so attribute revocable attribute-based encryption schemes [26]–[27] are proposed; as access policies may be revealed important privacy information of users, attribute-based encryption schemes with hidden access policy [28], [29] were proposed; in many commercial application scenarios, multiple attribute authorities are required for attribute distribution and management, so multi-authority attribute-based encryption schemes [30]–[31] has been developed. In addition, with the widespread application of mobile devices with limited computing and storage resources, outsourced decryption in attribute-based encryption schemes [32], [33] are proposed. At present, attribute-based encryption technology has been well developed and applied in traditional cloud storage systems.

1) Access structure

Definition: Let $U = \{ a_1, a_2, .., a_n \}$ be a set of attributes. For $a_i \in U$, $S_i = \{ v_{i,1}, v_{i,2}, \ldots, v_{i,ni}, \}$ is a set of possible values, where n_i is the number of possible values for a_i . Let $L = [L_1, L_2, .., L_n]$ $L_i \in S_i$ be an attribute list for a user, and $W = [W_1, W_2, .., W_n]$ $W_i \varepsilon S_i$ be an access policy. The notation $L \models W$ express that an attribute list L satisfies an access policy W, namely $L_i = W_i$ (i=1,2..,n). The notation $L \nvDash W$ implies L not satisfying the access structure W.

2) Ciphertext-policy attribute based access control

A cipher text policy attribute based encryption scheme consists of four fundamental algorithms: Setup, Key Generation, Encryption and Decryption.

Setup: The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK.

Key Generation (MK,S): The key generation algorithm takes as input the master key MK and a set of attributes S that describe the key. It outputs a private key SK.

Encrypt (PK,A, M): The encryption algorithm takes as input the public parameters PK, a message M, and an access structure A over the universe of attributes. The algorithm will encrypt M and produce a ciphertext CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. Assume that the ciphertext implicitly contains A.

Decrypt(PK,CT,SK): The decryption algorithm takes as input the public parameters PK, a ciphertext CT, which contains an access policy A, and a private key SK, which is a private key for a set S of attributes. If the set S of attributes satisfies the access structure A then the algorithm will decrypt the ciphertext and return a message M.

3) Security Model for CP-ABE

Init. The adversary sends the two different challenge access structures W_0^* and W_1^* to the challenger.

Setup. The challenger runs the Setup algorithm and gives the public parameters, PK to the adversary.

Phase 1. The adversary sends an attribute list L to the challenger for a Key Gen query, where ($L \nvDash W_0^*$ and $L \nvDash W_1^*$) or ($L \models W_0^*$ and $L \models W_1^*$) The challenger answers with a secret key for these attributes.

Challenge. The adversary submits two equal length messages M0 and M1. Note that if the adversary has obtained SK_L where ($L \models W_0^*$ and $L \models W_1^*$) then M0 = M1. The challenger chooses d randomly from {0,1} and runs Encrypt(PK, M_d, W_d^*). The challenger gives the ciphertext CT* to the adversary.

Phase 2. Same as Phase 1. Guess. The adversary outputs a guess d' of d.

The advantage of an adversary A in this game is defined as

$$\Pr[d'=d] - 1/2 .$$

Definition: A ciphertext-policy attribute based encryption scheme is secure if all polynomial time adversaries have at most a negligible advantage in the above game.

III. SYSTEM MODEL

In this section, we will present our system model by explaining its principle and entities.

A. System model

As shown in Figure 4, our system model is made of four main entities, i.e., the data owner (DO), data consumers (users), cloud server and Blockchain including other own entities.

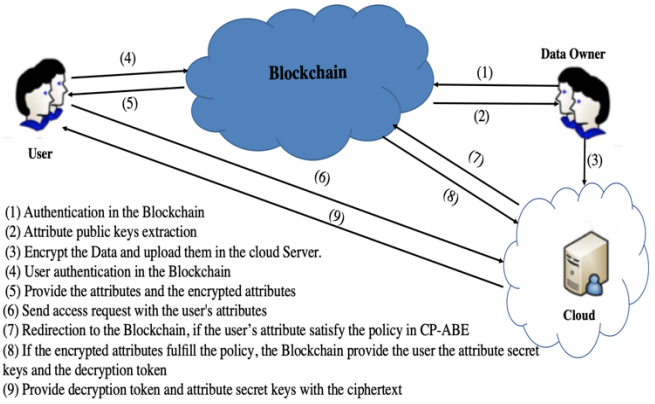

Figure 4. System model of distributed access control for Big Data

1) The main entities

Data Owner: One who sets access policies for these resources identified by different addresses generate through his wallet to ensure pseudo-anonymity.

User: A user in our infrastructure is the one who requests access to a resource identified by an address and stocked in the cloud.

Server. The cloud server is assumed to be semi-trusted. More specifically, the server is curious and honest, that is, the server is curious about the encrypted data stored on it and executes the assigned tasks properly. The server is responsible for storing the ciphertexts and providing data access service to users. It also obtains the user's attributes secret key from the Blockchain. When the server receives a ciphertext access request from a user, it searches the user's attributes secret key from the blockchain and generates a decryption token (TK) for the user.

Blockchain: The blockchain is considered as a database that stores all processed transactions and access control policies for each pair (owner, requestor) as a smart contract in chronological order shared by all participating users or nodes. A blockchain is a specific path in a tree structure of generated blocks, each referencing exactly a previously generated block.

2) Blockchain entities

The wallet: Each user has at least one wallet containing their identifiers, addresses and transactions. It contains all the keys needed to register and identify its resources, sign transactions, request access. In our framework, we consider a wallet as an Authorization Management Point (AMP), whether it is a web or mobile application, through the wallet, the system manager could record its resources to protect and define its access control policies. Then, the main features of a wallet are:

1) Generate keys and addresses.
2) Attributes Extraction and encryption.
3) Transform access control policies into transactions and broadcast them to the network.

Address: In our framework, users are either a DO (data owner) or a requestor and their resources can have an almost unlimited number of cryptographic identities, called addresses. The addresses are public and shared on the network. They are used to grant and request an access token. An address is basically the hash of an ECDSA public key and a user in possession of the corresponding private key is the address owner.

Transaction: A transaction in our framwork is considered as a communication form between network nodes. In fact, all the nodes of the network: minors, DO, Rq and their resources are identified by addresses and interact with each other via transactions. Each transaction has an identifier, at least one input and one output and a value to transfer from the sender of the transaction to its recipient. The input can be seen as an address source, it refers to the address of the entity that creates the transaction. The output can be viewed as the target or destination of the transaction value stream and is an address itself.

Our infrastructure introduces four types of transactions:

1) GetAccess Transaction: Created by an DO, it is used to deploy a smart contract in the blockchain. Then, its address source corresponds to an address of a DO and its address destination corresponds to an address of a SmartContract.

2) RequestAccess Transaction: is created by a requestor to interact with SmartContract. Then, its address source corresponds to an address of a Rq and its address destination corresponds to an address of a SmartContract. We trigger an intelligent contract by sending this requestAccess transaction. It then runs independently and automatically as follows on each node of the network, depending on the input included in the trigger transaction. In the data included in the RequestAccess transaction corresponding to the access control policy defined in SmartContract, a token is generated and included in a list of authorization tokens. Otherwise, access to the application is rejected

3) AllowAccess Transaction: Uses the authorization token. Then, its address source corresponds to an address of a Rq and its address destination corresponds to an address of a resource.

SmartContract: we use a SmartContract called PolicyContract. It is a representation of an access control strategy defined by an DO, to manage access to one of its resources. It's a script stored on the blockchain. Since he lives on the channel, he has a unique address. This SmartContract is triggered by addressing a RequestAccess transaction type. It then automatically executes on each node of the network according to the data included in the

 ISBN: 978-1-61208-719-1

trigger transaction. If the data completes the access control policies, the policy contract will be executed correctly, and then generate and assign an authorization token to the sender of the RequestAccess transaction. For each data, the DO defines a PolicyContract that is responsible for managing its access control functions. The way in which authorization tokens are generated will be explained in detail in the next chapter.

Authorization token: In our infrastructure, an authorization token represents the right of access or right defined by the owner of the policy contract to the sender of the RequestAccess transaction who successfully triggered the policy agreement in order to access to a specific resource identified by its address. Each applicant has a list called Auto Tokens List.

Block: Blocks are types of data used to store data permanently in the blockchain. The main reason for permanently storing data on the network is the way transactions are verified by the network, always keeping all information about them open to the public. A block consists of several transactions and SmartContracts that should not be contained in another block. Each block always refers to exactly one previous block by containing a hash of the referenced block. This characteristic is what creates the blockchain which consists of several blocks. The most recent block contains some or (ideally) all transactions and SmartContract that have been broadcast on the network but are not so far stored in the previous blocks that are already part of the blockchain.

B. Our solution description

Our framework consists of two processes. The first concerns the data owner, who encrypts their data with CP-ABE before outsourcing it to the cloud, and the second concerns the access requester who wants to have to access a resource stored in the cloud.

1) The process of the Data Owner:

The data owner authenticates with his wallet in the blockchain, which provides him with the necessary keys (public and secret, and attribute public key) for encrypting data with the CP-ABE system. the Data Owner defines the access policy and the wallet takes charge of encrypting the attributes in the policy. So that they are not visible in the blockchain and transforms it into the smart contract, then it broadcasts it as a transaction in the Blockchain until it reaches the minors, and then the transaction will be stored in the Blockchain, otherwise it will be rejected.

2) The process of the user:

Before the user submits an access request to desired resource in the cloud, he first authenticates with his wallet, the user retrieves their attribute keys and their encrypted attributes and sends them with his request to the cloud. If the requester's attributes satisfy the security policy, then the cloud redirects the request to the blockchain, which in turn verifies the encrypted attributes, whether they are authentic or not. if so, it provides the cloud with the secret attribute keys and generate a decryption token, which is sent to the cloud which in turn provides the requester access with the secret attribute keys, authorization token and the encrypted text and the latter one will combine them to decrypt the data.

C. Discussion

Numerous efforts have been emerged in adapting traditional access control models to meet new requirements in terms of security. Xiao et al [34] proposed a decentralized multi-authority CP-ABE scheme, which can support efficient decryption outsourcing and user revocation by leveraging a key separation technique. In this scheme, each Attribute Authority is independent and can dynamically join and leave the system. They also apply the scheme to achieve efficient and scalable distributed access control for Big Data in clouds. Unfortunately, those typical security and access control standards today are built around the notion of trust where a centralized trusted entity is always introduced, which harm user transparency and privacy. In addition, access control becomes a distributed problem. We therefore turn our attention to blockchain, the technology behind bitcoin protocol, to conceive our new framework as efficient solution that solve all challenges extensively highlighted in [35]-[37]. Actually, the blockchain is a technology breakthrough that has fundamentally changed our notions of centralized authority. It is a universal digital ledger that functions at the heart of decentralized financial systems, such as bitcoin, and increasingly, many other decentralized systems. The main contributions of this work can be summarized as follows.

1) We construct an efficient and scalable distributed access control scheme using Blockchain for Big Data in clouds, in which there is no need of a central authority and no entity is capable of decrypting ciphertexts individually. The scheme is more efficient than existing schemes and provably secure in the generic group model.

2) We design a decentralized CP-ABE scheme with efficient decryption outsourcing, as another security layer for managing the policy defined by the Data owner.

This framework is efficient and more suitable for Big Data access control than existing solutions, because it relies on the two security factors, which improve and increase the security level in this context.

IV. THE FEATURES ANALYSIS OF OUR SCHEME

Our framework is composed of 7 phases which are: system initialization, attributes extraction and Encryption, grant access, access request, access decision making, obtaining access and validation of authorization token .

Phase 1: Initialization of the system:

The data owner retrieves their attribute public keys by his wallet after authentication process, then encrypts their data with CP-ABE system before outsourcing them in the cloud. In addition, the data owner defines the security policy for these resources.

 ISBN: 978-1-61208-719-1

Phase 2: Extraction and encryption of attributes:

In this phase the wallet extracts the attributes from the predefined policy, encrypts each attribute, and puts these encrypted attributes into the policy.

Phase 3: Grant Access:

Reload the access control policy as a smart contract in the blockchain with GetAccess Transaction: After setting the access policy and encrypting the attributes, the DO wallet transforms these policies with encrypted attributes into SmartContract called PolicyContract and broadcasts it in the Blockchain via the GetAccess transaction, which is signed with the owner's private key. Then, the wallet broadcasts the GetAccess transaction to the Blockchain, as shown in Figure 5.

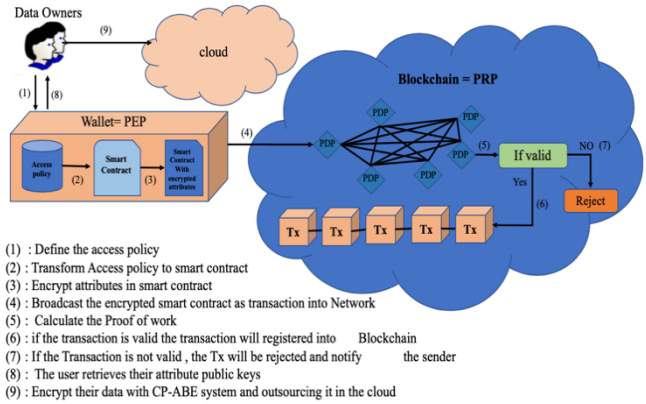

Figure 5. Reload access policies process

Peer-to-peer nodes check the transaction and save it to the Blockchain if successful. At this point, the policy contract, managing access to the resource in cloud, is deployed in the blockchain and ready to be triggered by the requestor who wishes to access a resource in the cloud.

The GetAccess transaction sequence can be displayed as follows:

- The DO (data owner) defines for his resource identified by the address Rs an access control policy:

  Policy (Rs)
- The wallet transforms this access control policy into a subscription agreement:

  Policy (Rs, Rq) ⟶ πx
- The wallet generates a GetAccess transaction to deploy this PolicyContract in the blockchain.

The GetAccess transaction is in the following form:

Tx = (m, sigRs (m)) where m = (IDx, $from$ (Rs), to (πx))

- Each node verifies the transaction in the process of validating the transaction. If the transaction is valid, the access control policy is saved in the blockchain as SmartContract. Otherwise, the transaction will be rejected.

At the end of this phase, if the transaction appears in the blockchain, it means that the network is witnessing that the data owner (DO) is protecting access to his resource through this policy agreement. As a result, anyone wishing to access this resource must unlock the access condition by successfully running the deployed PolicyContract after an encrypted attribute check in the access policy and in access request, if the encrypted attribute are identical. To do this, the requestor must trigger the policy contract and prove to the network that it actually fulfills the access requirements in a new transaction called RequestAccess transaction, which is the goal of the next phase.

Phase 4: RequestAccess: triggering the policy contract

In this phase, the user creates a new transaction, which he will call a RequestAccess transaction. The RequestAccess transaction triggers the PolicyContract if the user's attributes fulfill the policy in ciphertext stored in the cloud and follows the access control policy defined in PolicyContract (Figure 6).

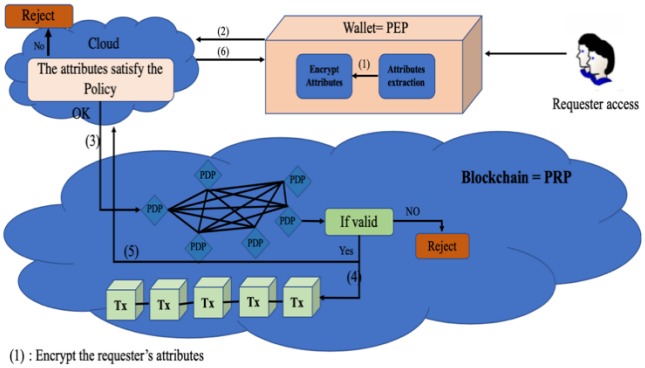

Figure 6. Request access process

The RequestAccess transaction sequence can be displayed as follows:

1) The access requester's wallet retrieves the attributes and encrypts them.

2) The user sends an access request to a resource in the cloud by enclosing the access request with the attribute public keys and the encrypted user's attributes.

3) The cloud redirects the requestor to the policy contract in Blockchain as a transaction, if the user's attributes satisfy the policy in the ciphertext and broadcasts the transaction to network nodes until it reaches minors. They check the transaction and run PolicyContract.

4) The smart contract runs automatically if the access requester attributes and the policy attributes in the smart contract are identical. In this case, the transaction will stored in the blockchain and places its response as entries in the RequestAccess transaction.

MeetAccessControlPolicy (πx_x) ⟶ ψ

The RequestAccess transaction type is in the following form:

Tx = (IDx, $from$ (Rq, ψ), (πx))

5) Provide the cloud with the attribute secret keys and generate an authorization token.

6) The cloud provide the requestor access with the ciphertext, attribute secret keys and the authorization token in order to combine them to decrypt the ciphertext.

Phase 5: Evaluating the Access Control Policy by Running PolicyContract

Each minor receives a signed transaction in the following form:

$(Tx, sigA\ (Tx))$ where Tx = (IDx, $from\ (A.pk, \psi)$, $to\ (B.\text{pk}, \pi x)$)

where A represents the access requester, Tx represents the transaction, IDx represents the transaction identifier, $A.pk$ represents the access requester's public key, and ψ represents the policy verification response, B.pk represents the public key of the target and πx represents the address of the smart contract.

To validate the transaction and evaluate the access control policy, the node performs the following functions:

1) CheckIdentity: This function provides the following properties:

- Authenticate the sender.
- prove his property to the resource
- prove his non-repudiation.

This function is performed by verifying the sender's signature using this procedure:

$Check\text{A}\ (Tx, \sigma) = \text{True}$

where σ is the signature of the requester.

2) CheckIntegrity(Tx): chop the transaction and compare it by its ID to make sure that the transaction has not been modified when it was propagated in the network. This function is performed by performing this procedure: Compare (H (Tx). IDx) = $True$

3) Checkattribute (Att): checks if the encrypted attributes in the policy and in the user request are identical.

Compare (Enc (Att (A) .Enc (Att (πx)) = True

where Att represents the attributes and Att (πx) represents the attributes in the smart contract.

CheckPolicy: Checks whether the sender follows the access control policy by running PolicyContract, an identifier that is permanently registered in the blockchain. This function is provided by the execution of these procedures:

a) Address GetPolicyContract (Tx) → πx

b) GetRequestAcess entry: (Tx) → ψ

c) Execute (ψ, πx) = True

If a result other than "True" remains after performing the described function, the transaction is considered invalid. It will be rejected, access will be refused and a notification will be sent to the sender. If successful, PolicyContract triggers another SmartContract named CreatTokenContract, identified by this $\pi' x$ address, to generate an authorization token and assigns it to the requester via an AllowAccess transaction in the following form.

Tx = (m, sigA (m)) where m = (IDx, $from\ (\pi' x)$, to (Rq, TKN (Rq, $\pi' x$))

1) CreateToken Contract Releases AllowAccess Transaction

2) The nodes of the network validate the transaction

3) If the transaction is valid, the unspent transaction output: TKN (Rq, $\pi' x$) is saved in the Blockchain and added to the requestor's token list.

Phase 6: Token generation

When the requester wants to access this resource, he creates a GetAccess transaction that uses the authorization token and the attribute secret keys obtained in the previous phase and sends them to the cloud. The GetAccess transaction has the following form:

Tx = (m, sigA (m)) where m = (IDx, $from$ (Rq), to (Rs, TKN (Rs, $\pi' x$))

Phase 7: Access to resource target

The cloud provides the user with the authorization token, attribute secret keys and ciphertext in order to decrypt this latter and obviously access to the resource.

V. CONCLUSION AND PERSPECTIVES

We created an efficient access control system in a Big Data environment using Blockchain. Our system allows granular access control based on the Blockchain scheme. The latter uses transactions to guarantee authentication, non-repudiation and integrity. We have demonstrated the feasibility of using blockchain technology to manage access control process for Big Data through the description of our proposed framework. The latter leverages the salient features of Blockchain that are, distribution, full-fledged and append-only ledger to make a promising solution for addressing the access control challenges in Big Data. However adopting the blockchain technology to handle access control functions is not straightforward and additional critical issues emerge that are: The public aspect of the blockchain versus the private aspect of some access control policies and the inherent traceability problem. To address these issues, the encryption was used to encrypt attributes in the data owner's security policy and access requester's attributes to mask the transparency and visibility of the attributes to the public. Regarding the problem of traceability, we planned to ensure it in our next work.

We also used the ciphertext-policy Attribute Based Encryption (CP-ABE) scheme in order to add another security layer in our framework. This technology is a promising cryptographic primitive for the security of cloud storage system, which can bring fine-grained access control. In the future, we envision implementing our model using access control tools and Multichain for the blockchain. The results of this implementation would help us to evaluate the security level and performance of our proposed infrastructure.

REFERENCES

[1] S. Hosseinzadeh , P. Hosseinzadeh, and S. E. Najafi, " introducting a hybrid model of DEA and data mining in evaluating efficiency. Case study: BankBranches," Acad; J; Res. Econ. Manag. , vol. 3, no. 2, 2015

[2] P. Cato, P. Gölzer, and W. Demmelhuber, "An investigation into the implementation factors affecting the success of big

 ISBN: 978-1-61208-719-1

data system," in 2015 11th International Conference On Innovations Technologiy (IIT), 2015, PP. 134-139

[3] D. Xia, Z. Rong, Y. Zhou, B. Wang, Y. Li, and Z. Zhang, "Discovery and analysis of usage data based on hadoop for personalized information access," in Proceedings - 16th IEEE International Conference on Computational Science and Engineering, CSE 2013, 2013, pp. 917–924.

[4] V. N. Gudivada, R. Baeza-Yates, and V. Raghavan, "Big data: Promises and problems," Computer, vol. 48, no. 3, pp. 20–23, 2015.

[5] P. Kassani, A. Teoh, and E. Kim, "Evolutionary-modified fuzzy nearest-neighbor rule for pattern classification," Expert Syst. Appl., vol. 88, pp. 258–269, Dec. 2017.

[6] Big Data Working Group, "Expanded top ten big data security and privacy challenges : Cloud Security Alliance.", [Online]. Available:https://downloads.cloudsecurityalliance.org/initiatives/bdwg/Expanded_Top_Ten_Big_Data_Security_and_Privacy_Challenges.pdf. [Accessed: 06-Jun-2019].

[7] W. Meng, E. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When Intrusion Detection Meets Blockchain Technology: A Review," IEEE Access, 2018., vol. 6, PP. 10179-10188, Jan. 2018.

[8] M. Samaniego and R. Deters, "Internet of Smart Things - IoST: Using Blockchain and CLIPS to Make Things Autonomous," in Proceedings - 2017 IEEE 1st International Conference on Cognitive Computing, ICCC 2017, PP. 9-16. 2017.

[9] K. Yang, X. Jia, and K. Ren, "Dac-macs: effective data access control for multi-authority cloud storage systems," in IEEE INFOCOM'13, 2013, pp. 2895-2903.

[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in IEEE S&P'07, 2007, pp. 321-334.

[11] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in PKC' II, 2011, pp. 53-70.

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in ACM CCS'08, 2006, pp. 89-98.

[13] C. Tankard, "Big data security," Network Security, pp. 5-8, 2012.

[14] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," Int. J. Inf. Secur., vol. 14, no. 6, pp. 487–497, Nov. 2015.

[15] F. Lombardi and R. Di Pietro, "Secure virtualization for cloud computing", *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1113-1122, 2010.

[16] S. Zawoad and R. Hasan, Cloud forensics: A meta-study of challenges approaches and open problems, Feb. 2013.

[17] D. Kumar and K. Morarjee, "Survey on insider data theft misuse attacks in the cloud", *Int. J. Comput. Sci. Mobile Appl.*, vol. 2, no. 2, pp. 26-29, 2014.

[18] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System,". Oct. 2008. [online]:Available: https://bitcoin.org/bitcoin.pdf [Accessed: 03-Jun-2019].

[19] J.-P. Delahaye, "Les blockchains, clefs d'un nouveau monde," Pourlascience.fr. [Online]. Available:https://www.pourlascience.fr/sd/informatique/les-blockchains-clefs-daposun-nouveau-monde-8354.php. [Accessed: 03-Jun-2019].

[20] D. Johnson, A. Meneze, and S. Vansto, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," certicom. [Online]. Available:http://www.cs.miami.edu/home/burt/learning/Csc609.142/ecdsa-cert.pdf. [Accessed: 05-Jun-2019].

[21] A. Gervais et al, "On the Security and Performance of Proof of Work Blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2016, pp. 3–16.

[22] "What is a Merkle Tree? Beginner's Guide to this Blockchain Component." [Online]. Available: https://blockonomi.com/merkle-tree/. [Accessed: 06-Jun-2019]

[23] "How Do Ethereum Smart Contracts Work? - CoinDesk." [Online]. Available: https://www.coindesk.com/information/ethereum-smart-contracts-work. [Accessed: 06-Jun-2019].

[24] D. Mills et al, "Distributed ledger technology in payments, clearing, and settlement," Distributed ledger technology in payments, clearing, and settlement, 2016. [Online]. Available: http://ccl.yale.edu/sites/default/files/files/Mills%20et%20al%20Distributed%20Ledger%20Technologies.pdf. [Accessed: 05-Jun-2019].

[25] A. Sahai and B. Waters, ''Fuzzy identity-based encryption,'' in Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2005, pp. 457–473.

[26] L. Zu, Z. Liu, and J. Li, ''New ciphertext-policy attribute-based encryption with efficient revocation,'' in Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT), Sep. 2014, pp. 281–287.

[27] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, ''User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage,'' IEEE Syst. J., vol. 12, no. 2, pp. 1767–1777, Jun. 2018.

[28] A. Kapadia, P. P. Tsang, and S. W. Smith, ''Attribute-based publishing with hidden credentials and hidden policies,'' in Proc. NDSS, vol. 7, 2007, pp. 179–192.

[29] Y. Zhang et al, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," Inf. Sci., vol. 379, pp. 42–61, Feb. 2017.

[30] M. Chase, ''Multi-authority attribute based encryption,'' in Proc. Theory Cryptogr. Conf. Berlin, Germany: Springer, 2007, pp. 515–534.

[31] H. S. G. Pussewalage and V. A. Oleshchuk, ''A distributed multi-authority attribute based encryption scheme for secure sharing of personal health records,'' in Proc. 22nd ACM Symp. Access Control Models Technol., 2017, pp. 255–262.

[32] J. Li, Y. Wang, Y. Zhang, and J. Han, ''Full verifiability for outsourced decryption in attribute based encryption,'' IEEE Trans. Services Comput., to be published.

[33] J. Li, X. Lin, Y. Zhang, and J. Han, ''KSF-OABE: Outsourced attributebased encryption with keyword search function for cloud storage,'' IEEE Trans. Services Comput., vol. 10, no. 5, pp. 715–725, Sep. /Oct. 2017.

[34] M. Xiao, M. Wang, X. Liu, and J. Sun, "Efficient distributed access control for big data in clouds," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015, pp. 202–207.

[35] U. Feige, J. Killian, and M. Naor, "A Minimal Model for Secure Computation (Extended Abstract)," in *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 1994, pp. 554–563.

[36] A. Ouaddah, H. Mousannif, A. Abou elkalam, and A. Ait Ouahman, "Access control in The Internet of Things: Big challenges and new opportunities," *Comput. Netw.*, vol. 112, Nov. 2016.

[37] H. Wang, X. Jiang, and G. Kambourakis, "Special Issue on Security, Privacy and Trust in Network-based Big Data," *Inf Sci*, vol. 318, no. C, pp. 48–50, Oct. 2015.

 ISBN: 978-1-61208-719-1