Decentralized Access Control Infrastructure Using Blockchain for Big Data

Oussama Mounnan
OSCARS Laboratory, Ensa of Marrakesh,
Cadi Ayyad University
Morocco
oussama.mounnan@gmail.com

Anas Abou El Kalam
OSCARS Laboratory, Ensa of Marrakesh,
Cadi Ayyad University
Morocco
elkalam@hotmail.fr

Lamia El Haourani
OSCARS Laboratory, Ensa of Marrakesh,
Cadi Ayyad University
Morocco
Lamia.elhaou@gmail.com

Abstract—**Big Data technology has demonstrated its effectiveness in several areas, and several projects confirm the existence of innovative opportunities. However, The security and privacy issues are gradually magnified in these environments. Traditional access control mechanisms cannot guarantee the user's anonymity and privacy. This article proposes a new access control infrastructure based on Blockchain technology for Big Data, to publish the policies, deployed in Smart contract, expressing the right to have access to a resource and provide the identification and authentication processes. In our infrastructure the policies are visible to the public, and any user can know the policy paired with a resource and who has the right to access, which compromises privacy. To deal with this problem, we encrypt the attributes in the access policy represented by the smart contract. We also encrypt the attributes in the access requester, and the verification is done solely on the basis of the comparison of encrypted attributes, whether they are identical or not. If so, an authorization token is generated to allow access.**

Keywords— Big Data, Privacy, Blockchain, identity, Smart contract, anonymity, Encryption.

I. Introduction

The amount of data produced everyday by social media, streaming, medical records, cloud services, sensors, system logs, government archives, business transactions exceed the management capability of traditional data management mechanisms. The variety of data provenance obliges the existence of a heterogenous environment capable of processing massive volumes of data with response rate adapted to data emergence velocity.

Volume, Velocity and Varity (3Vs) [1] are the core characteristics that define Big Data. Volume can be seen as the quantity of data stored or generated that should meet a certain threshold to be considered as Big Data. Velocity is the speed of data generation and processing that characterizes the providing applications. Whilst variety describes the nature of data and its source. Some references introduce other characteristics: veracity and value, hence the term 5Vs.

Privacy issues emerge in these environments due to the sensitivity of user's mined information [2]. Big Data characteristics increase the chance of privacy breaches as traditional access control mechanism weren't designed to deal with such constraints. These privacy issues are spread all over the Big Data life cycle, i.e., data generation, storage, processing [3]. Data generation is the mining phase where organizations collect personal and public information about the targeted clients. Storage phase takes the gathered data and stores it generally in distributed file systems. Whereas the processing phase analytically studies, explores and finally exploits the data to respond the organization's needs, e.g., predicting preferred videos on YouTube, needed items on eBay, favorite movies on Netflix…Granular access system on each phase of the Big Data life cycle is mandatory to preserve privacy according to the agreed level between parties.

Blockchain 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. It is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way. The blockchain network has no central authority, it is the very definition of a democratized system. Since it is a shared and immutable ledger, the information in it is open for anyone and everyone to see. Hence, anything that is built on the blockchain is by its very nature transparent and everyone involved is accountable for their actions.

In this article, we take advantage of the consistency guarantees provided by this promising technology to create an access control framework for Big Data called Decentralized Access Control infrastructure using Blockchain for Big Data (DACBBD) .

Our proposed scheme retains the benefits of blockchain to fulfil security and privacy needs, while overcoming the challenges in integrating the blockchain to Access Control in Big Data Context and namely the public visibility. Our proposed solution combines the emergent Blockchain technology, access control and Big Data. We use the smart contract to express detailed contextual access control policies, in order to make authorization decision. Authorization tokens are used, as an access control mechanism, to enforce access policies in distributed environments, i.e. without any central control body and to ensure that the strategies are implemented by all interacting entities.

In order to provide finer protection preserving privacy and confidentiality for our solution: we introduce the notion of encrypting attributes into the security policy predefined by the data owner and the attributes of the access requester which makes it possible to hide the visibility of the attributes to the public to guarantee the confidentiality and the anonymity, and this is one of the constraints imposed by the use of the Blockchain "the public aspect".

The remainder of our paper is structured as follows. In section II we present the related works to our research. In section III, we introduce the preliminaries and background. We expose our solution by explaining its principle and reveal our model architecture in section IV, and finally we conclude in section V.

978-1-7281-5052-9/19/$31.00 ©2019 IEEE

II. Related Work

The Blockchain is a public replicated distributed data repository, always available, irreversible, and tamper-proof. It allows unsuspecting users to agree on immutable and verifiable data without third-party interaction. In other words, blockchain technology makes it possible to create a secure database only by relying on a distributed consensus protocol to decide new valid data to be added in a distributed manner. Historically, blockchain technology was first introduced to support cryptocurrencies and, to date, cryptocurrencies remain its main area of practical application, although several proposals in other areas are still in the pipelines' study. The first blockchain was used by the Bitcoin cryptocurrency protocol [4] and today, Bitcoin is still the most widespread example of the adoption of blockchain technology. In the literature, several models have been proposed, combining this emerging technology with access control.

Shangping Wang et al [5] provide a framework that combines the IPFS distributed storage system [6], the Ethereum blockchain [7], and attribute-based encryption (ABE) technology [8] to provide detailed data access control in distributed storage systems. The owner of the data is the only one to control his own data. It is not necessary to use the trusted private key generator (PKG) in their system. It is also able to distribute a secret key to users, which is more flexible than traditional ABE systems. At the same time, the Ethereum blockchain is used to manage the secret key of users, the problem of key management in traditional ABE systems is solved. When a user forgets his own secret key, he simply reads the corresponding transaction data in the Ethereum blockchain and decrypts it to obtain his own secret key information. By creating encrypted keyword indexes for the shared file, encrypted keyword index information is stored in the Ethereum blockchain. In addition, the smart contract [9] is deployed on the Ethereum blockchain to implement keyword research in decentralized storage systems. Once the smart contract is deployed, it will work in good faith and in accordance with the smart contract logic. Service charges will only be paid if users retrieve the correct search results. The problem that search service providers may not return search results honestly to traditional cloud storage is resolved in their schema.

Hamza ES-SAMAALI et al [10] introduce a new distributed access control framework based on Blockchain technology, which for the first time combines access control models and cryptocurrency blockchain mechanisms. they propose the use of the Smart Contract [11] to express detailed and contextual access control policies in order to make authorization decisions. they opt for authorization tokens as an access control mechanism provided via emerging cryptocurrency solutions. They use blockchain to ensure the application of access policies in distributed environments where there is no central authority / administrator, and ensure that the policies are correctly implemented by all interacting entities. In fact, the use of smart contacts allows the framework to implement expressive and granular access control policies, expressed by any access control model as soon as this model can be transcoded into scripting language.

Potential privacy issues and vulnerabilities in existing health data storage and sharing systems, as well as the concept of autonomous data ownership, Xueping Liang et al [12] offer an innovative health data sharing solution focused on user using a decentralized and authorized blockchain. Protect confidentiality by using a channel training scheme and improve identity management by using the membership service supported by the blockchain. A mobile application is deployed to collect health data from personal portable devices, manual entry and medical devices, and to synchronize cloud data for data sharing with health care providers and companies health insurance. To preserve the integrity of the health data, proof of integrity and validation can be permanently retrieved from each database in each record and anchored to the blockchain network. In addition, for scalability and performance considerations, they adopt a method of batch processing data to manage large sets of personal health data collected and uploaded by the mobile platform.

Guy Zyskind et al [13] describe a system decentralized personal data management ensuring that users own and control their data. They implement a protocol that turns a blockchain into an automated access control manager that does not require the trust of a third party. Unlike Bit Coin, transactions in this system are not strictly financial: they are used to carry instructions, such as storage, querying, and data sharing.

Aafaf Ouaddah et al [14] introduce FairAccess as a management framework Fully decentralized, pseudonymous, and privacy-preserving permissions in IoT, allowing users to own and control their data. To implement their model, they use and adapt the blockchain in a decentralized access control manager. Unlike financial transactions in bitcoins, FairAccess introduces new types of transactions that grant, obtain, delegate and revoke access.

III. Background and Preliminaries

In this section, we highlight the background, technologies behind our proposed solution for a better understanding.

A. Genaral model of access control application

The Internet Engineering Task Force (IETF) defines an abstract model for the application of access control that is implemented in most existing implementations of access control mechanisms. This model makes a clear distinction between the Policy Decision Point (PDP) component and the Policy Enforcement Point (PEP) component:

- The PEP component intercepts the access request and forwards it to the PDP. After receiving the decision of the PDP, PEP applies it.
- The PDP component analyzes the access request, evaluates the contextual conditions, resolves any conflicts between permissions and prohibitions and calculates the final decision (access granted or not).

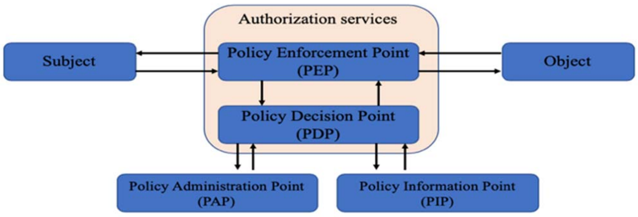

Authorization services
Subject
Policy Enforcement Point (PEP)
Object
Policy Decision Point (PDP)
Policy Administration Point (PAP)
Policy Information Point (PIP)

Fig. 1. Application of access control model

- Policy Information Point (PIP): Serves as a source of attribute retrieval, or data required for policy evaluation to provide information necessary for the PDP to make the decisions.

- Policy Administration Point (PAP): Provides a user interface to create, manage, test, and debug Digital Policy and Meta-Policy, and store these policies in the appropriate repository.

B. *Big Data Security*

Today, Big Data has penetrated various industries and has become a kind of production factor that plays an important role. With the development of rapid processing and analysis technology, the potential information it contains can quickly capture valuable information to provide a reference for decision making. Although Big Data provides opportunities for businesses and society itself by exploiting varieties and volumes of data, the challenges of information security play a significant role. Without good security and encryption solutions, Big Data might be a big problem.

One of the biggest Big Data challenges is to store and analyze all the information. Most of companies use different technologies such as NoSQL, Hadoop, Spark and other analytical software, or artificial intelligence and Machine learning, to find the insights they need.

Security is also an important concern in the Big Data areas. Few companies use additional mechanisms to increase the security level such as identity and access control, encryption and data segregation.

The confidentiality of information is about protecting Data against unintentional, unlawful or unauthorized access, disclosure or theft. Information with high confidentiality concerns is considered secret and must be kept confidential to prevent identity theft, compromise of accounts and systems, legal or reputational damage, and other severe consequences.

C. *Security and Privacy requirements in Big Data*

The security and privacy issues are magnified in the Big Data environment, caused by its specific requirements and features. In this section we analyze the Big Data in terms of security and privacy.

According to the recent published European Union regulation for electronic identification [15], the implementation of security measures is based on these criteria:

➢ Privacy: The information will be used in ways approved by the person who provided it [16]. It is possible to collect and combine personal information from several different sources, known as information aggregation, which has resulted in databases containing data that could be used in ways the original data owner has not agreed to or even knows about. Information aggregation is pieces of non-private data that, when combined, may create information that violates privacy. User privacy (user data and personal information) should be flexibly preserved according to the policy and expectation of Big Data users. To achieve this aim, there are some key factors shaping privacy. Among it. (i) Transparency implies actions of openness and accountability. Transparency doesn't imply success or failure of information security; it dictates actions at questionable cross roads. (ii) Anonymity : Anonymized data is data that can no longer be associated with an individual in any manner [17]. Anonymization, roughly speaking, is the act of removing personal identifiers from data, for example, by converting personally identifiable information into aggregated data. However, applying these techniques is not very easy in the Big Data context [18]. (iii) Decentralization: The Data are shared in the network by its different nodes without any central control body.

➢ Confidentiality is limiting access to only those with sufficient privileges and a demonstrated need to access it. Disclosure confidentiality is breached intentional or unintentional when exposure of an information asset to unauthorized parties. For example confidential information could be mistakenly e-mailed to someone outside the organization. Or an employee discards rather than destroys, a document containing a critical information. To protect the confidentiality of information, different mechanisms [19] are used such as cryptography (encryption) , segregation and classification of information, application of security policies, secure document storage….

➢ Integrity describes how Data are whole, complete and uncorrupted. When it is exposed to corruption, damage or other disruption of its authentic state. When corruption can occur while information is being entered, stored or transmitted.

➢ Availability refers to authorized users have access to information in a usable format, without interference or obstruction. It does not imply that the information is accessible to any users, rather, it means it can be accessed when needed by authorized users.

D. *Blockchain*

The blockchain is an undeniably ingenious invention, The brainchild of a person or group of people known by the pseudonym, Satoshi Nakamoto [4]. But since then, it has evolved into something greater. Blockchain technology created the backbone of a new type of internet. Originally devised for the digital currency, Bitcoin, the tech community has now found other potential uses for the technology.

A blockchain is a time-stamped series of immutable record of data that is managed by cluster of distributed computers not owned by any single central entity. Each of these blocks of data are secured and bound to each other using cryptographic principles (i.e. chain) [20].

1) Blockchain concept

The blockchain is a simple way of passing information from A to B in a fully automated and safe manner. One party to a transaction initiates the process by creating a block. This 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 [21]. The blockchain is maintained by a peer-to-peer network. The network is a collection of nodes which are interconnected to one another. Nodes are individual computers which take in input and performs a function on them and gives an output. The blockchain uses a special kind of network called "peer-to-peer network" which partitions its entire workload between participants, who are all equally privileged, called "peers".

There is no longer one central server, now there are several distributed and decentralized peers. The decentralized nature of a peer-to-peer system becomes critical the simple idea of combining this peer-to-peer network with a payment system has completely revolutionized the finance industry by giving birth to cryptocurrency.

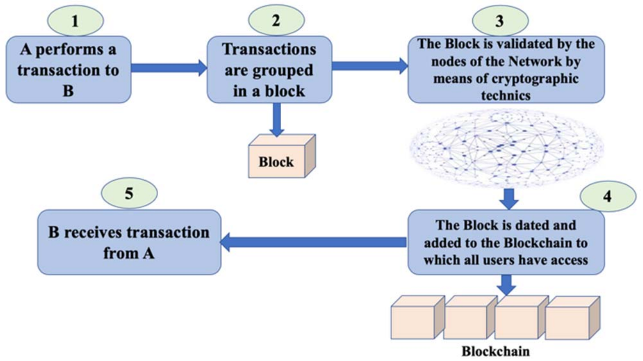

1
A performs a transaction to B
2
Transactions are grouped in a block
3
The Block is validated by the nodes of the Network by means of cryptographic technics
Block
4
The Block is dated and added to the Blockchain to which all users have access
5
B receives transaction from A
Blockchain

Fig. 2. Blockchain technology principle

2) *The Merkle Tree*

Merkle tree is a tree in which every leaf node is labelled with the hash of a data block, and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes. It is also known as hash tree and this concept is named after Ralph Merkle. Hash trees allow efficient and secure verification of the contents of large data structures. Hash trees are a generalization of hash lists and hash chains.

Demonstrating that a leaf node is a part of a given binary hash tree requires computing a number of hashes proportional to the logarithm of the number of leaf nodes of the tree. This contrasts with hash lists, where the number is proportional to the number of leaf nodes itself.

The block is divided into two main categories which are the header and the body. The header has four components, a timestamp, a nonce, a hash reference to a previous block and a hashed list of all transactions that took place since the last created block. The blocks are stored in a multi-level data structure, a tree structure called the Merkle tree [22]. This structure is the key factor of the mining. The Merkle tree or binary hash tree is a type of a binary tree, where the bottom of the tree contains the transactions (hashed), the intermediate tree nodes (leaves) contain the hash of the two

nodes that made it, all the way till the top where it is a single hashed tree-node called the Merkle root (root hash).

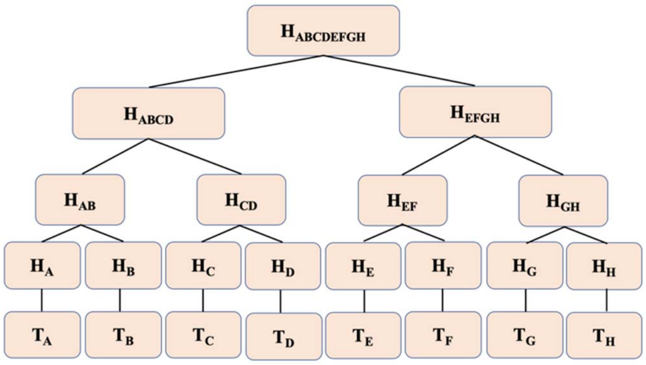


Fig. 3. Merkle Tree principle

3) *The Wallet*

Blockchain wallet is a computer program that allows to monitor and conduct cryptocurrency. In crypto wallets transactions are recorded on the blockchain. It can be compared to how an email account works. You might say that you can store and receive cryptocurrencies like emails. If someone sends you cryptocurrency, it means it is assigned to the address of your blockchain wallet but recorded in a distributed ledger. 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.

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 [23] 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 latter to the network. 3) validating received transactions from the network .

4) *Proof of work*

Proof of work (PoW) is the most popular algorithm being used by currencies such as Bitcoin and Ethereum, each one with its own differences. It is a consensus protocol [24] introduced by pioneer, Bitcoin and used widely by many other blockchain projects. This process is mostly known as “mining” and as such the nodes on the network are known as “miners”. The PoW comes in the form of an answer to a mathematical problem, one that needs considerable work to arrive at, but is easily verified to be correct once the answer has been reached. In order for an actor to be elected as a leader and choose the next block to be added to the blockchain they have to find a solution to a particular mathematical problem. By solving a complex mathematical puzzle that is part of the bitcoin program, and including the answer in the block. The puzzle that needs solving is to find a number that, when combined with the data in the block and passed through a hash function, produces a result that is within a certain range. This is much harder than it sounds. The process involves ensuring every confirmed block in the chain rewards the miner in the cryptocurrency that they are mining through the transaction fees collected for sending currency across the network, as well as any predetermined reward. It ensures that miners are incentivized to continue maintaining a blockchain, as they are being rewarded for doing so.

5) *unspent transaction output*

An unspent transaction output, better known as a UTXO, is an important concept in the world of blockchain. It is the output of a transaction that a user receives and is able to spend in the future. We always need an unspent transaction output UTXO to make a transaction. If we don’t have a UTXO, it simply means we don’t have any Bitcoin. This mainly happens due to the protocol rules which Satoshi Nakamoto had defined in Bitcoin to prevent double spending. Also, there is no way in the bitcoin world to spend partial amounts while completing a transaction. to make this clear : let's consider that we have a balance of 3 BTC on public address and we have to pay 1 BTC to a merchant, we cannot simply send 1 BTC out of our address and keep the rest 2 BTC. Instead, we need to spend whole 3 BTC out of which we will designate 1 BTC to the merchant while providing a signature and sending the rest 2 BTC back to our self on an address that we control. This is called sending the change to the change address. In short, when a bitcoin transaction takes place, there are two UTXOs created: one that is the actual coins sent to the recipient, and one that is the change output, which goes back to the sender's wallet.

6) Smart contract

A smart contract [25] is computer protocol, executable code that runs automatically on the blockchain to facilitate, execute and enforce the terms of an agreement. The main aim of a smart contract is to automatically execute the terms of an agreement once the specified conditions are met. Thus, smart contracts promise low transaction fees compared to traditional systems that require a trusted third party to enforce and execute the terms of an agreement. A smart contract can be thought of as a system that releases digital assets to all or some of the involved parties once arbitrary pre-defined rules have been met [26]. For instance, Alice sends A currency units to Bob, if she receives B currency units from Carl. Many different definitions of a smart contract have been discussed in the literature. According to the [27], all definitions are classified into two categories, namely, smart contract code and smart legal contract. Smart contract code means "code that is stored, verified and executed on a blockchain". Smart legal contract means code to complete or substitute legal contracts. Smart contracts have several advantages.

- Autonomy : It is not necessary to adopt a central control entity or intermediaries to confirm. The execution of the instruction is done automatically.
- Backup: Each node in the blockchain network has the duplication of data.
- Security: Thanks to the cryptography technics.
- Saving: It saves money and time.

Accuracy: avoids error that result from manually manipulation.

7) Blockchain authentication and identification

Nowadays, The potential of the use of the technology Blockchain is gradually increase in several areas. An interesting aspect is that the blockchain can be used as an authentication provider. Blockchain uses the key-pair for the users to register their identity. The personal information is stored in form of hashes which can be used for several identity-related attributes like name, unique identity number or other biometric information. A user can go to a recognized party, which verify the hashes earlier registered on the blockchain and let the recognizing party "sponsor" that piece of information as the truth on the blockchain. Other parties which trust the particular recognizing party can now trust the identity on the blockchain and use it as an authentication or identification mechanism. Thanks to the ECDSA (elliptic curve digital signature algorithm) mechanism. The authentication process is done trusty. 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.

IV. Proposed Solution

A. Our Solution: DACBBD description

Considering the following scenario in which a subject, for example, an access requestor A, identified by the address Rq wishes to perform an action on a protected resource identified by the address Rs, belonging to the data owner B. We assume that the requester already knows the access control policy governing access to resource B. The DACBBD workflow is as follows:

The access requestor fulfills the conditions specified in the access control policy and submits its request via its wallet in the form of a Request Access transaction. Afterwards, the wallet broadcasts this transaction to the nodes of the network until it reaches the minors. These then act as a Policy Distributed Point (PDP), evaluate the transaction, and verify the request with the policy defined by executing a Policy Contract already deployed by the owner in the blockchain via an earlier transaction called GetAccess. Running PolicyContract determines whether the request should be allowed or denied. Finally, if it has been successfully run, Policy Contract generates and assigns an authorization token to the requestor's address through an AllowAccess transaction. Then, the authorization token is saved in the blockchain and appears in the list of authorization tokens of the requester. Finally, the applicant presents the authorization token to the target resource via a GrantAcess transaction. The target resource verifies the validity of the authorization token by referring to the blockchain. If this authorization token has been issued by the Smart Contract corresponding to the Data Owner B, it authorizes the access otherwise it refuses.

And to address the problem of the visibility of the access policy attributes of the data owner and the attributes of the access requesters, we proceed as follows:

Before the access policy is transformed into a Smart contract, the attributes will be extracted by the DO's wallet and the DO will encrypt the attributes, which will then be returned to the policy. When the access requester wishes to access the resource, his wallet retrieves the attributes, encrypt them, and then sends the request with the encrypted attributes through the RequestAccess transaction. The verification of the attributes is done by the comparison of the axes. If the attribute axes are identical, then the requestor meets the specified conditions of the access policy, and therefore access will be granted, otherwise the access request will be rejected.

The main entities of our infrastructure are:

User: One who want to access to a resource identified by an address.

Data Owner: One who defines access policies for his resources identified by different addresses generate through his wallet.

The wallet: It considerate as a web or mobile application, through it the user can manage his credentials, addresses and transactions. It contains all the keys needed to register and identify its resources, sign transactions, request access. In our infrastructure, 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: Each user can generate addresses to identify their resources and it refers to the cryptographic identity. 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 (Elliptic Curve Digital Standard Algorithm) public key and a user in possession of the corresponding private key is the address owner.

Transaction: A transaction in DACBBD is considered as a communication form between network nodes. In fact, Each entity in the network is identified by an addresses and interact with each other via transactions.

DACBBD 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: In DACBBD, we use a SmartContract. It is a representation of an access control policy defined by an Data Owner, to manage access to its resources. It's a set of code lines that runs and executes automatically and stored on the blockchain. 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 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 DACBBD 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.

Blockchain: In DACBBD, 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.

B. *DACBBD: phases and functionalities*

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

Phase 1: Initialization of the system:

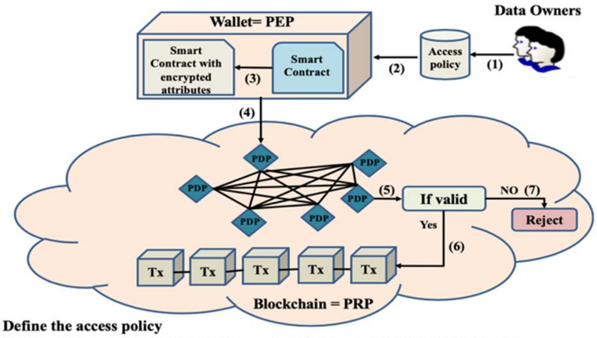
The owner of the data identifies his resources via addresses generated by his wallet, then publishes the addresses of his resources while keeping the confidentiality of the corresponding private keys. In particular, each resource is identified by an address . In addition, the data owner defines the security policy for these resources and it is assumed that each access requester can obtain an authentic copy of the addresses.

Phase 2: Extraction and encryption of attributes:

In this phase the wallet extracts the attributes from the predefined policy and encrypt them, and puts these encrypted attributes into the policy.

Phase 3: Access allocation:

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 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 4A.



(1) : Define the access policy
(2) : User's Authentication and transformation of Access policy to smart contract
(3) : Encrypt attributes in smart contract
(4) : Broadcast the smart contract as transaction into Network
(5) : Reach consensus of the validation
(6) : If the transaction is valid, the transaction will be registered into Blockchain
(7) : If the Transaction is not valid , the Tx will be rejected and notify the sender

Fig. 4. 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 is deployed in the blockchain and ready to be triggered by the requestor who wishes to access this resource.

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 Policy Contract in the blockchain.

The GetAccess transaction is in the following form:

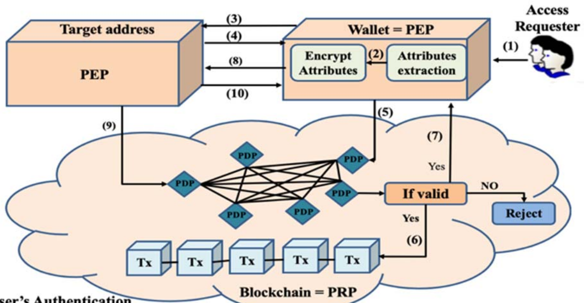
Tx = (m, sigRs (m)) $where$ m = (IDx, $from$ (Rs), to (πx))

- ➢ 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 Smart Contract. 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 Policy Contract after an encrypted attributes check in the access policy and access requestor and obtain the token authorization if the encrypted attributes 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.

Phase 4: RequestAccess: triggering the Smart contract

In this phase, the user creates a new transaction called RequestAccess transaction. This transaction triggers the Smart Contract and follows the access control policy defined in it (Figure 5).



(1) : User's Authentication
(2) : Encrypt the requester's attributes
(3) : Send the request access (@resource, action)
(4) : redirects to Smart contract (@Smart Contract)
(5) : Calls the Smart Contract in GetAccess Transaction
(6) : Execute the smart Contract if the transaction is valid and stores it in the Blockchain
(7) : The Smart contract generates an authorization Token if the attributes satisfy the policy and sends it to the requester
(8) : The wallet resends the request access with the received token to the target address
(9) : checks the validity of the token in an AllowAccess Transaction
(10): allows the requester to access

Fig. 5. Request access process

The RequestAccess transaction sequence can be displayed as follows:

1. The access requester's wallet retrieves the attributes, then encrypt them.

2. The user sends an access request to the target resource identified by its public address by enclosing the access request with the encrypted attributes.

3. The target resource redirects the requestor to the policy contract address to obtain the token if the encrypted attributes are identical.

4. The requester obtains the Smart Contract address (πx)

5. The requestor fulfills the access control condition by comparing the encrypted attributes in the access requester and the security policy in the smart contract and places its response as entries in the RequestAccess transaction.

MeetAccessControlPolicy (πx_x) → ψ

6.The wallet generates a RequestAccess transaction type in the following form:

Tx = (IDx, $from$ (Rq, ψ), (πx)

7. The wallet broadcasts the transaction to network nodes until it reaches minors. They check the transaction and run Smart Contract if the encrypted attributes are identical, to generate the authorization token.

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, $from$ ($A.pk$, ψ), to (B.pk, πx))

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:

$Check$A (Tx, σ) = 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) CheckAtt (EncAtt): checks if the encrypted attributes in the request and in smart contract are identical.

Compare (EncAtt (A) .EncAtt (πx)) = True

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

4) 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 two 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, Smart Contract triggers another Smart Contract named CreatTokenContract, identified by this π'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, $from$ (π'x), to (Rq, TKN (Rq, π'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: Access a resource

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

$$\text{Tx} = (\text{m}, sig\text{A}\ (m))\ wh\text{ere m} = (\text{IDx}, from\ (\text{Rq}), \text{to (Rs}, TKN\ (\text{Rs}, \pi'x))$$

Phase 7: validation of the authorization token

The requestor sends an AllowAccess transaction to the final resource. The latter can check whether the token is valid or not by checking the validity of the AllowAccess transaction and by referring to the Blockchain.

V. Conclusion and Perspectives

The paper defines an approach to create, manage and enforce control policies that are published in the Blockchain. We create an decentralized access control infrastructure based on Blockchain technology for Big Data (DACBBD). The latter one guarantees authentication, non-repudiation and integrity through transactions. Our scheme exploits the salient features of Blockchain 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. To address these issues, the cryptography mechanisms, namely encryption, are used to encrypt attributes in the data owner's access policy represented by the Smart Contract and access requester's attributes to mask the transparency and visibility of the attributes to the public. Smart contracts represent a significant corollary to blockchain and distributed ledger technology systems, adding the potential for automated processes and functionality.

In the future, we plan to extend our work to study how to better embed an access control system in Blockchain technology. We also plan to implement our model using access control tools and private Blockchain. The results of this implementation would help us to evaluate the security level and performance of our proposed infrastructure.

References

[1] M. Hilbert, "Big Data for Development: A Review of Promises and Challenges," Dev. Policy Rev., vol. 34, pp. 135–174, Jan. 2016.

[2] "(PDF) Big Data for Development: From Information- to Knowledge Societies.": https://www.researchgate.net/publication/254950835_Big_Data_for_Development_From_Information_to_Knowledge_Societies.

[3] Securing the Big Data Life Cycle, ORACLE& MIT TECHNOLOGY REVIEW CUSTOM

[4] Satoshi Nakamoto, "Bitcoin." [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[5] 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, vol. 6, pp. 38437–38450, 2018.

[6] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," Jul. 2014.

[7] "Ethereum Project." [Online]. Available: https://www.ethereum.org/.

[8] L. Ibraimi, M. Asim, and M. Petković, "Secure management of personal health records by applying attribute-based encryption," in Proceedings of the 6th International Workshop on Wearable, Micro, and Nano Technologies for Personalized Health, 2009, pp. 71–74.

[9] "Blockchain : qu'est-ce qu'un Smart Contract et à quoi ça sert ?": https://www.lemagit.fr/conseil/Blockchain-quest-ce-quun-Smart-Contract-et-a-quoi-ca-sert.

[10] Hamza ES-SAMAALI et al, "A blockchain based access control for Big Data.": http://www.ijcncs.org/published/volume6/issue7/p1_5-7.pdf.

[11] "Fermat's Library | Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform annotated/explained version.":https://fermatslibrary.com/s/ethereum-a-next-generation-smart-contract-and-decentralized-application-platform.

[12] "(PDF) Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications," ResearchGate: https://www.researchgate.net/publication/320337312_Integrating_Blockchain_for_Data_Sharing_and_Collaboration_in_Mobile_Healthcare_Applications.

[13] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized Computation Platform with Guaranteed Privacy," ArXiv150603471 Cs, Jun. 2015.

[14] "(PDF) Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT.": https://www.researchgate.net/publication/308567618_Towards_a_Novel_Privacy_Preserving_Access_Control_Model_Based_on_Blockchain_Technology_in_IoT.

[15] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, vol. 257. 2014.

[16] Z. Yan and S. Holtmanns, "Trust Modeling and Management: From Social Trust to Digital Trust," Comput. Secur. Priv. Polit. Curr. Issues Chall. Solut., pp. 290–323, 2008.

[17] A. Pfitzmann and M. Köhntopp, "Anonymity, Unobservability, and Pseudonymity — A Proposal for Terminology," in Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings, H. Federrath, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 1–9.

[18] anonymity:Higher Education Information Security Council (HEISC), "Guidelines for data de-identification or anonymization," https://spaces.internet2.edu/display/2014infosecurityguide/Guidelines+ for+Data+De-Identification+or+Anonymization, 2015.

[19] "ISO/IEC 15408-1:2009 - Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model.": https://www.iso.org/standard/50341.html.

[20] Z. B. • I. Y. a 11 Mois, "steemit and blockchain? What is Blockchain really? - Personal search," Steemit, 07-Jan-2018: https://steemit.com/blockchain/@zayin/steemit-and-blockchain-what-is-blockchain-really-personal-search.

[21] [Online]. Available: http://keyrus-prod.s3.amazonaws.com/Avis%20d%27expert/Blockchain/Avis%20d%27Expert_Blockchain-en%20COM.pdf. [Accessed: 15-Jul-2019].

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

[23] "Elliptic curve digital signature algorithm — Wikipédia." https://fr.wikipedia.org/wiki/Elliptic_curve_digital_signature_algorithm.

[24] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "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.

[25] "How Do Ethereum Smart Contracts Work? - CoinDesk.":https://www.coindesk.com/information/ethereum-smart-contracts-work.

[26] V. Buterin, "A next-generation smart contract and decentralized application platform.," Available online at: https://github.com/ethereum/wiki/wiki/White-Paper/

[27] J. Stark, "Making sense of blockchain smart contracts," Available on line at : http://www.coindesk.com/making-sense-smart-contract