Design of an Electronic Voting System Using a Blockchain Network

Abstract. Design of a scalable electronic voting system, which, based on a generic model designed for this application called **voting cell**, guarantees the integrity of the information through the use of a private network Blockchain.
For the validation of the system, the implementation of a cell was carried out, for which fifty voters and three voting options were enabled. The stored data was intentionally modified to corroborate the error correction method used by the block chain networks and thus ensure the integrity of the voting system results.

Keywords: Blockchain, Transaction, Hash, Database Entity and Attributes, Cryptographic Algorithm.

1 Introduction

In recent years, the term Blockchain has gained popularity as a technology tending to change the world as it is known; mainly due to its wide use in crypto-currency systems such as Bitcoin and Ethereum. However, this technology offers a number of opportunities that go beyond those applications such as: smart contracts, copyright certifications, property certifications and voting systems [1], that is, it can be applied to everyday situations that require guarantee the integrity of the information.

The electoral systems used by nations, be they manual or electronic, generate distrust in the citizenship since the results can be altered if access to the instances that contain the information is obtained; minutes and ballots in case of manual voting and database servers in electronic systems. In addition, with practices of cybernetic espionage and social engineering, you can get to determine the preference of choice of citizens.

Apply Blockchain to electoral systems, allows maintaining the confidentiality of each person's vote and by its decentralized nature, allows preserving the integrity of the results obtained.

Electronic voting systems backed by Blockchain technology have been developed around the world. One of these cases is the POLYS project launched by the Kaspersky Lab incubator firm based on the Ethereum smart contracts model [2]. *Polys is an online voting platform based on blockchain technology and backed with transparent crypto algorithms. Powered by Kaspersky Lab* [3]

2 Literature Review

2.1 What is Blockchain?

It is a set of transactions that are registered in a shared database. When using cryptographic keys and being distributed in multiple computers and/or servers, it presents advantages in security against possible manipulations and frauds. Any modification in one of the copies would be useless, since the change should be made in all copies because the database is open and can be audited by anyone who has access to it [4]. For this reason, it is said to be a decentralized system [5].

In Blockchain each block is identified by a hash placed in the header. This is generated using the SHA-256 Secure Hash Algorithm to create a character string of fixed size (256 bits). The SHA-256 will take a flat text of any size as input, and will encrypt it into a 256-bit binary string [6] represented in 64 hexadecimal characters. Each header contains information that links a block to its previous block, in the string that is known as the base. The primary identifier of each block is in its header; is a fingerprint that is constructed by combining two types of information: the information relative to the new block created and the previous block in the chain. [7]

2.2 Public and private Blockchain.

The Blockchain can be public or private, according to the permits and accesses granted to it:

A public blockchain network is one in which there are no restrictions to read the string. The data - which can be encrypted - and the sending of transactions for inclusion in the Blockchain can be reviewed by any entity that has access. A private blockchain network is one in which the direct access to the data of the chain of blocks and the sending of the transactions, is limited to a predefined list of users [8].

2.3 Blockchain components.

1) Miners are dedicated computers to review and validate the transactions that are carried out [9]. Also, they build and write the new data - called blocks - in the Blockchain. [4]
2) The nodes are computers that, using software, store and distribute a copy of the Blockchain. [9]
3) The blocks are records of all transactions that are packaged in blocks that the miners then verify. Then they will be added to the chain once their validation is completed and distributed to all the nodes that make up the network. [10]

2.4 Merkle tree.

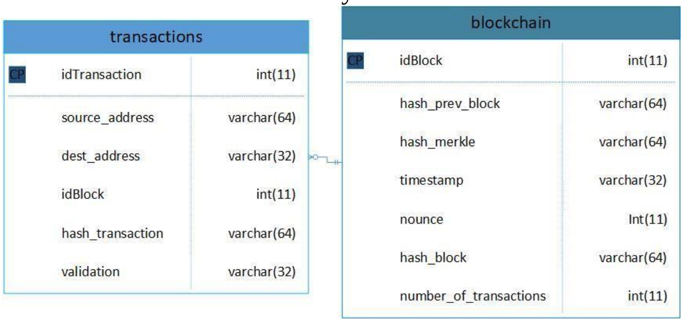
It is a binary structure based on results of the hash function. It is used to obtain a summary of all transactions included in a block. [11]
It is constructed by ordering the transactions in a list and proceeding to count them to check if they are an even number [12], otherwise the last transaction is repeated. Then, hashing of each transaction is performed; the hashes are serialized in pairs and the hashing process becomes recursively until the root of the Merkle tree or root is obtained [13].

3 Voting System Design and Protocols.

The design of the system is based on a *private Blockchain network*, that is, the data cannot be accessed from the internet and the system does not need to use any platform. In other words, the information is stored and processed by the system devices. The voting system delivers results when consulting databases whose records - hereinafter called *transactions* - can be audited and corroborated using cryptographic algorithms. For this solution, SHA-256 was used.

3.1 Data structure.

The system is made up of two unrelated databases. The first contains the information of the people authorized to vote in each voting site and the second contains the Blockchain and is made up of two database entities each one with different attributes [14]: first entity stores the transactions, where each vote is represented by one transaction, and the other one stores the blocks that certify transactions.



transactions		
CP	idTransaction	int(11)
	source_address	varchar(64)
	dest_address	varchar(32)
	idBlock	int(11)
	hash_transaction	varchar(64)
	validation	varchar(32)

blockchain		
CP	idBlock	int(11)
	hash_prev_block	varchar(64)
	hash_merkle	varchar(64)
	timestamp	varchar(32)
	nounce	Int(11)
	hash_block	varchar(64)
	number_of_transactions	int(11)

Fig 1. Blockchain Entities

The transactions entity has the following attributes.

- **idTransaction:** it has characteristics of primary key and auto-increment. Its function is to perform the identification of the transaction.

- **source_address:** this attribute contains the identification number of the user. This is got operating the identification number with a random string character of the a finite vector -which is part of characteristic system signature- the result is encrypted with SHA-256. **Thus, this process ensures that the voter identity will be non-obtained.**

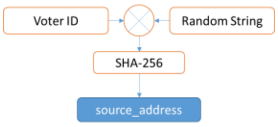


Fig 2. Voter identification encryption.

For the real application the random string method can be changed for information based in biometric identifiers like fingerprint recognition or facial recognition.

- **dest_adrress:** indicates the option by which the user voted. By counting this attribute, you get the result of the vote.
- **idBlock:** this attribute indicates the number of the block that certifies the transaction. A block can certify one or more transactions; if no block validates the transaction, this field has a null value - these will be called orphan transaction -.
- **hash_transaction:** this string data is calculated by concatenating the attributes source_address and dest_address with a random string character of the finite vector which is part of characteristic system signature; this process guarantees that the valid transactions can be only written by system members. The result is encrypted with SHA-256.

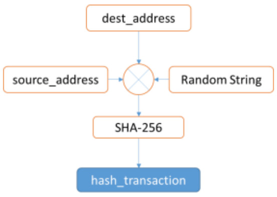


Fig 3. Construction of the transaction hash.

- **Validation:** this attribute is purely informative and is modified each time the Blockchain database is validated. Its value can be "Validated" or "Non-validated".

The blockchain entity has the function to certify the groups of transactions and has the following attributes.

- **idBlock:** it has characteristics of primary key and auto-increment. Its function is to identify the block.
- **hash_prev_block:** contains the hash of the immediately preceding block. When this is the first block in the chain, this field is zero "0".
- **hash_merkle:** based on the characteristics of the Merkle tree, a single hash is generated that contains information on the orphan transactions that are certified by the block under construction.

- **timestamp:** it is the system's time capture at the moment of generating the block. It is necessary that all the participating nodes are synchronized.
- **nounce:** it is a random number of maximum five digits and it helps to increase the difficulty to alter or rewrite a block.
- **hash_block: is** generated from operating the attributes hash_prev_block, hash_merkle, timestamp and nounce of this entity. The result is encrypted with SHA-256 and its function is to validate the transactions contained in the block under construction.

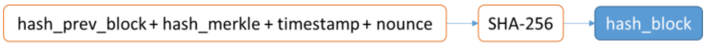


Fig 4. Construction of the block hash.

- **number_of_transactions:** indicates the number of transactions certified by the block.

3.2 Architecture.

To guarantee the reliability of the system, maintaining the secrecy of the vote and protecting the integrity of the results, we have designed **the voting cell system**. The cells represent a generic design implemented in each of the voting sites.

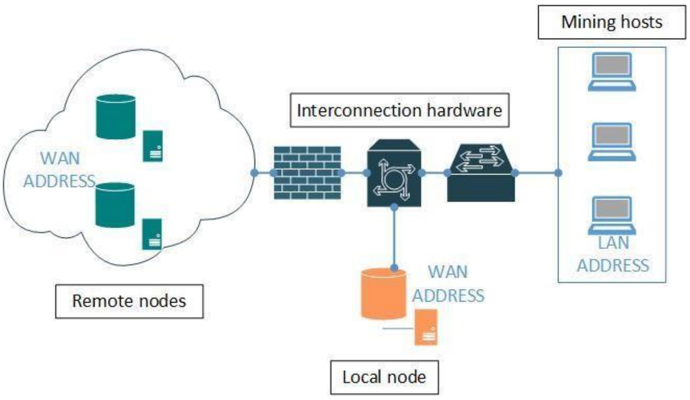


Fig 5. Voting cell architecture.

The cells are made up of the following components.

- **Mining hosts:** they are the user interfaces. Their function is to validate the registration of the voter, in addition, they are responsible for writing each transaction – vote - and generating the blocks that authenticate the groups of transactions.

- **Local node:** it **stores a database with people authorized to vote on the voting site and a copy of the blockchain** this equipment also executes a JavaScript in order to calculate the hash_merkle, calculate the hash_transaction and calculate the hash_block on the copy of the blockchain which is stored in here.
- **Interconnection hardware:** they are focused on guaranteeing connectivity between cells -voting sites-. Additionally, they allow the implementation of policies that guarantee a reliable network -fault tolerance, scalability, quality of service and security-. [15]

The components of the voting cell described so far are the devices that are installed in each voting site.

Taking into account the sensitivity of the information in a actual system application, the use of firewall - such as software or device - is necessary in each voting site in order to implement policies that guarantee the sources and destinations of the traffic of data. [16]

Remote nodes: these are equipment with similar functions to the local node. In practice, they are local nodes of other voting cells, or also, physical and virtual equipment located in places other than voting sites; their function is to store and validate a exactly copy of the Blockchain, therefore, this system is a decentralized one.

3.3 Operation.

The system uses two general processes in which different actors participate: *Data processing and Data storage*.

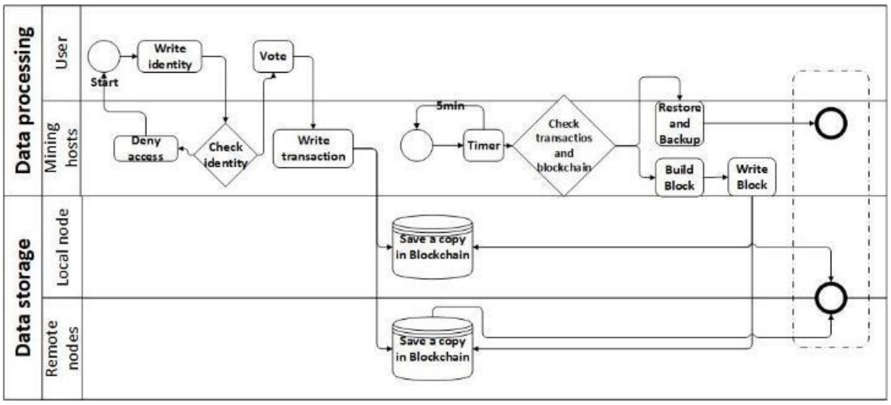


Fig 6. Processes diagram.

Data Processing. It starts by typing the voter's identification number in the user interface. Here the mining host, through which the system is being accessed, performs the data verification that consists of corroborating if the user is authorized to participate in that voting site or if he has already deposited the vote - denying access in both cases - or if the user can actually deposit his vote. For the last case, the mining host shows the card to the user so that he can make his choice.

When the vote is deposited, the mining team establishes a connection with the nodes that are available and starts the following processes:

- Blockchain verification: in this process, the integrity of the data contained in the two entities of the blockchain is validated in all the nodes with which connection was established. If corrupt database [17] copies are found, the miners updates the database of the affected node using the incremental backup method, that is, only the modified information is replaced. [18]
  This verification is done by analyzing the hash_transaction attribute of the transactions entity and the hash_block attribute of the Blockchain entity and this is being carried out all the time both by the miners, who validate any copies of the blockchain with which they have connection, and by the nodes, which validate only the information they have stored.
- Write transaction: once the data has been validated, the attributes of the transaction's entity are written.
- Build block: is activated by a timer in the miner hosts. In this process, the activated miner establishes connection with all the nodes within his reach and blocks the multisession access while performing the verification, that is, during this time the other miners do not have access to the nodes. Once the Blockchain is verified, the remote nodes are unlocked and the block is started on the local node by performing the following sequence.
  a. Search orphan transactions: select those transactions whose attribute idBlock in the transactions entity is null. If these types of transactions are not found, the process is finished.
  b. Build the hash_merkle attribute: the hash_transaction attributes of the orphan transactions are taken, and a single string data is constructed using the Merkle tree.

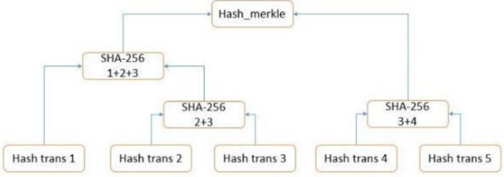


Fig 7. Merkle hash construction.

If a transaction is altered, the attributes hash_merkle and hash_block in the blockchain entity will be modified. Thus, the system can always identify the transactions –votes- that were altered. This process is persistently carried out by all the nodes on the copy that each one stores.

c. Take the time sample and generate the nounce: in order to obtain the timestamp and nounce attributes of the entity Blockchain.
d. Generate the hash of the block: Concatenates all the attributes of the Blockchain entity and encrypts the result with SHA-256, obtaining the attribute called hash_block.
e. Write the idBlock attribute in the transactions entity.
f. Update the version of the block chain in the other nodes: using the incremental backup method.

Upon completion of this process, the mining host releases the local node, allowing the other miners to write transactions. During the process of building the block, the cell cannot write transactions, in other words, the voting site is disabled. However, the execution time of the process is very short.

Data storage. Its actors are the local and remote nodes of the whole system. They contain exact copies of the chain of blocks, which are verified by the mining equipment. When corrupt copies are found, that is, copies with altered data or in outdated versions, the miners replace it with the latest version verified in the system - using the incremental backup method -.

4 A Voting Cell Implementation

A voting cell was implemented, in which fifty people are registered to exercise the vote and one card with three possibilities of election; the nodes store databases managed with MySQL Workbench [19] and the mining teams execute a JavaScript developed in NetBeans IDE 8.2 [20].

The cell implemented is shown in the Figure 9. It is made up of two mining teams, a local server that is a physical team that contains the database of people registered to vote and two remote servers: one physical and the other virtual making use of the Google Cloud Platform [21].

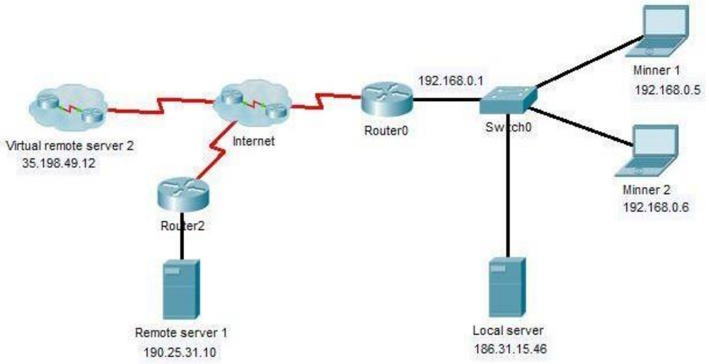


Fig 8. Voting cell implementation

4.1 Results

The script was executed on the mining equipment. First, the system requests the user's identification number. When the user is authorized, the system provides a digital card so that the voter can choice one of three options. By clicking on any of the options, the user deposits his vote.

Table 1. Writing the first transaction.

idTransactions	source_adrress	dest_adrress	idBlock	hash_transaction	validation
1	dd28307d6697aa59d86b530084c08ff047d8f8267b4ab0c156cbb02f0d09bfd7	Option1	NULL	0a9a603d50f4476d07b81269eff1e786a31ef8efdacb656702537b93ad0c7f1f	Validated

The identity of the voter cannot be known unless the signature keys of the system are obtained. When the same voting process is repeated through the two mining teams, multiple **orphan transactions** –with idBlock attribute in NULL state- are written.

Table 2. Orphan transactions.

idTransactions	source_adrress	dest_adrress	idBlock	hash_transaction	validation
1	dd28307d6697aa59d86b530084c08ff047d8f8267b4ab0c156cbb02f0d09bfd7	Option 1	NULL	0a9a603d50f4476d07b81269eff1e786a31ef8efdacb656702537b93ad0c7f1f	Validated
2	4b0e8bd3b5ad09b06243169a76c3d86bbb206e689b2b1b5cea257ffe6f216414	Option 2	NULL	f1707eb7aa3b34657f05d466d1ab82a51f6a29a037151078b98360dc88ff800d	Validated

3	70cd531d9a16d110523ecb9a15e252724e1bc1ea41f3d819038ac4ae3926370a	Option 1	NULL	b57e035444da5be7d4de9b5f5442a660609ffa7060fa546a74d4519f6bc6c488	Validated
4	dc89e5db37ea511fceecf49aa267612c27e9486b531f0d6ebd28d849e3810052	Option 3	NULL	c93163ac4c08fdae6d226d1136b94be67e76770139a512e6d20a143baf604d49	Validated

For the generation of a block, a five-minute timer was programmed. At the end of this time the process of building the block begins with all those orphan transactions existing in the entity.

Table 3. Building the first block.

idBlock	hash_prev_block	hash_merkle	timestamp	nounce	hash_block	number_of_transactions
1	0	2445e0bafe55e7c6b65e342c4765db82537b8da3bf23dbfaf81c3eafc98f4e51	6/02/2019 16:01	67901	4a16b448cf96280a02b15e017b45c39375078bc639f061ba513b35246e995928	4

The transaction showed in the table 3 is the first one, therefore, its attribute **hash_prev_block is "0"**. For this case the block construction process took less than **three seconds.**

The blocks building forces a change in the transaction entity specifically in the attribute **idBlock** of the transactions which was certified by the block recently built, therefore, the value NULL in the transactions 1,2, 3 y 4 is replaced for de number of the block which certificates the transactions, for this case block number one. This process permit that the system can do a relation between block and transactions.

4.2 B. Alteration of transactions.

As an exercise, one of the transactions in the chain of blocks stored on the remote server 1 was altered. The vote registered in the four transaction was changed; "Option 1" was written instead of "Option 3", that is, **a vote was removed for Option 3 and a vote was added for Option 1.** Thus, when the **Remote server 1** recalculates the attributes hash_merkle and hash_block the blockchain is invalidated at that moment, a miner evaluates the blockchain until it finds the block whose hash_merkle attribute is corrupt. Once it is identified, the transactions supported by the corrupt block are invalidated. Then the hash_transaction attributes of the invalidated transactions are recalculated; in this way it is possible to identify specifically the transaction that was altered and proceed to update it using the incremental backup method. So this system manages to maintain the integrity of the information without significantly increasing the traffic between the local and remote nodes.

4.3 Vote counting.

Once the elections are over, the system validates the chain of blocks again and performs the restores or updates of the version of the databases that are necessary, that is, if it finds a corrupt version of the chain of blocks, it is updated or restored. Then the system performs a count of the dest_address attribute of the transaction entity of the blockchain. For the case study, the version of the block chain contained in the local server was exported, from MySQL Workbench to Microsoft Office Excel 2013 and the results obtained were plotted.

5 Conclusions

With the proposed system, by using of the blockchain technology and databases distributed in an electronic voting systems, maintaining the secrecy of the vote and the integrity of the results is possible in an election activity.
The use of Blockchain technology allows easy detection of corrupt data; is achieved by checking the hash attributes of the previous block and merkle hash in the block header. What allows to detect alterations in the chain of blocks quickly and efficiently - with respect to the use of the processor -.
Despite this, there are vulnerabilities in the system. They are basically presence of malware or virus [22] in the mining computers and in the interconnection devices, which can alter the transactions before their registration in the Blockchain, that is, corrupt transactions can be validated and written as long as the information is modified before Your first record in the database. For this reason, it is necessary to establish conventional multilevel security policies, with which, the alteration of transactions is the last security instance of the system.
The use of the scalable model that we called voting cell, reduces the hardware and software requirements in the storage equipment compared with a centralized electronic voting system; the computing capacity is distributed in the mining equipment that is installed in the cells arranged to satisfy the elections. This implies decrease of costs of implementation of the voting system.

References

1. Ocampo M Carmen, 2017 “BLOCKCHAIN LA NUEVA BASE DE DATOS NO SQL EN BIG DATA,” Mgr. dissertation, Dept. Sist. Eng. Guadalajara Univ, Libre Univ. Bogotá, COL.
2. J. Rivero, 2018. “TRANSPARENCIA ELECTORAL: 5 PLATAFORMAS BLOCKCHAIN PARA VOTACIONES”.
3. Polys ONLINE VOTING SYSTEMS. [Online]. Available: https://polys.me

4. Yahari N Benjamin, 2016 “Blockchain y sus aplicaciones,” Eng. dissertation, Dept. Eng. Universidad Católica Nuestra Señora de La Asunción. Asunción, PAR.
5. Ayed, A. Ben, 2017. A CONCEPTUAL SECURE BLOCKCHAIN- BASED ELECTRONIC VOTING SYSTEM. [Online]. Available: http://aircconline.com/ijnsa/V9N3/9317ijnsa01.pdf.
6. J. Black, P. Rogaway, and T. Shrimpton, “Black-box analysis of the block-cipher-based hash-function constructions from PGV”, Advances in Cryptology – Crypto’2002. Lectures Notes in Computer Science, vol. 2442, Springer-Verlag, 2002, pp. 320-335.
7. Raval, S, 2016 Decentralized Applications: Harnessing Bitcoin's Blockchain Technology.O'Reilly Media, Inc.
8. BitFuryGroup, Garzik, J. Public versus private blockchains. –October 20 2015.
9. Baldeon C Valeria, Zambrano H Joel, 2018 “IMPLEMENTACIÒN DE UN PROTOTIPO DE UNA RED DESCENTRALIZADA BLOCKCHAIN PARA EL VOTO ELECTRÓNICO EN LA UNIVERSIDAD DE GUAYAQUIL”, Eng. dissertation, Fac. Ciencias Matemáticas y Física. Universidad de Guayaquil. Guayaquil, EC.
10. da Silva D. Carlos, 2017. “¿Qué es Blockchain y cómo funciona?” IBM Systems Blog para Latinoamérica – Español.
11. Ralph Merkle (1988). A digital signature based on a conventional encryption function. In: Advances in Cryptology – CRYPTO '87 (Lecture Notes in Computer Science), Vol. 293, pp. 369–378.
12. A. Narayanan, J. Boneau, E. Felten, A. Miller y S. Goldfeder, 2016, Bitcoin and Cryptocurrency Technologies, Princeton: Princeton University Press.
13. Aguirre Joffre, 2017, “Cadena de bloques: potencial aplicación a Historias Clínicas Electrónicas” Esp. dissertation, Fac . Cs. Ecnomicas, exactas, naturales e ingeniería. Uni. De Buenos Aires. Buenos Aires, AR.
14. “Entidades y atributos” In: Universidad Autónoma de Yucatán.
15. Wendell Odom, 2015. CCNA Routing and Switching official cert guide library, pp 200-125.
16. FORTINET NSE institute, 2018. FortiGate Security Study Guide for FortiIOS 6.0.
17. “Corrupción de datos” In: Spanish Wikipedia.
18. Benchimol Daniel, 2011, “Introducción,” in “Hacking desde cero” 1st ed. Ed. Fox Andina collaboration with Gradi S.A Buenos Aires, AR.
19. MySQL Workbench [Online] Available: https://www.mysql.com/products/workbench/
20. NetBeans IDE [Online] Available: https://netbeans.org/
21. Google Cloud In: Spanish Wikipedia Available: https://es.wikipedia.org/wiki/Google_Cloud.
22. Microsoft TechNet. “Defining Malware: FAQ” In: English