Digital Voting Systems Deploying the use of Blockchain Technology

Edward Danso Ansong
Computer Science Department
Kwame Nkrumah University of Science and Technology.
Kumasi, Ghana

Joshua Appiah
Computer Science Department
Kwame Nkrumah University of Science and Technology.
Kumasi, Ghana

Benjamin Odoi-Lartey
Computer Science Department
Kwame Nkrumah University of Science and Technology.
Kumasi, Ghana

ABSTRACT
Blockchain is a list of records called blocks which are linked through the use of cryptography. Each block contains a cryptographic hash of the previous block, a timestamp and transaction data. A summary of the all transactions are kept in a mekle tree root hash and stored as part of the header of a block. By design blockchain is highly resistant to modification of data. The case study uses blockchain technology to deploy a digital voting system and tries to provide a more decentralized approach to digital voting

Keywords
Blockchain; voting; technology; distributed; decentralized;

1. INTRODUCTION
Blockchain technology is an open distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way. Blockchain is managed within a peer to peer network which forms a complete network of distributed mesh of computer nodes that speaks a protocol for inter-node communication and validating new blocks that is added to the chain of blocks. Once a block is recorded it is virtually impossible to alter the transactions that are carried there in due to the use of cryptography which ensures that altering a block will require the altering of all the subsequent blocks. This process is said to be computationally difficult. Before a block can be accepted into the chain, a consensus must be taken within the network where a majority must accept the proposed block.

Current digital voting systems serve all votes from a single server or an unsecure decentralized approach that leaves the system prune to attacks hence the adoption of the digital systems for voting has not been seen in recent years. For any digital voting system, the most fundamental requirements are security, verifiability, openness, non-repudiation and immutable data. This is also the assurances we get from a well-designed blockchain system.

2. DIGITAL VOTING ARCHITECTURE
In the digital voting process we will employ a full In this paper we introduce an intersection of blockchain technology and digital voting systems where all the fundamental requirements of a correct system is fulfilled.

3. ELECTIONS OVERVIEW
In an election, there are voters, candidates and an election commission. A voter is an entity that has been authorized and authenticated to cast his/her votes. A candidate on the other hand is anyone who will be voted on. A commission is an organization that is in charge of the whole election process.

The election commission goes by an organogram spilling out the structure of the commission which might differ from region to region. For our application purposes, we will go by a more general type.

The commission structure is made up of

i. Polling stations

ii. Constituencies

iii. Head commission

The polling station is where the voting process actually takes place. All voters are authenticated and authorized and given a ballot paper containing the candidates. They vote and put the casted votes into a ballot box. After the appointed time for voting, all votes are read and each vote will either be accepted or denied. After the count, all votes are forwarded to the constituencies.

The constituencies are also responsible for collecting all votes from polling stations that falls under the specific constituencies and ascertain the authenticity of all the votes that were casted at the polling station level. The final data is then forwarded to the headquarters where finalizing of the votes occur.

The head commission provides authorized candidates and authorized ballot papers. After election, they take all the votes from constituency level and hence announce the winner for the given election.

We shall make use of similar structure for our digital voting system to ensure total decentralization of the whole process. decentralized approach to the voting process which requires all process from authentication, authorization, casting votes, reading of votes and the declaration of winners take a decentralized fashion. It also leans on the values that the voting process must be more decoupled from any predefined voting process but rather it must be capable of handling different voting process using a generic voting scheme and smart contract principles.

3.1 Parts of the System
They will be made up of (i) voters, (ii) issuers, and (iii) verifiers

(i) voters are all authorized entities that can cast their votes in the digital system. They can be issuers, or verifiers and they are analogues to voters in established voting processes.

(ii) issuers are users of the system that will like the network mesh monitor a voting process for them.

The issuer may be analogues to the electoral commissioner in established voting processes. The issuer also presents a list of voters who can be accepted as authorized voters within the system. This list is used by all verifiers within the system to verify the authenticity of voters.

(iii) verifiers are analogues to polling stations within voting processes and analogues to miners in a typical crypto currency network like bitcoin and ethereum. They authenticate each voter's validity and ensure non-repudiation within the network thus ensuring that each voter votes just once within the system.

3.2 Mesh Network

The mesh network is the largest network within the system that will consist of all nodes within our blockchain. The mesh will be made up of private and trusted nodes due to the lack of incentives that is geared toward establishing trust. Each node must have an authorization list that contains all authorized nodes that can join. If a node (guest) tries to join another node(host), the host node must already have a list of authorized nodes. The guest nodes then must present its own validation token which the host node validates against its list. If the node is authorized, it is given access to the network by giving other nodes for it to pair with. If the pairing process is completed successfully, it then listens for elections advertisements.

0 count and the current STATION_NUM is decreased as well with it. If a current node is already a member of a blue sheet, it sends out the blue sheet to a node it hasn't contacted directly. The node which receives this request to join the network might be or might not be a member of the particular blue sheet. This process occurs until all members have been visited and the current STATION_NUM and POLLS_NUM are 0, or a timeout has occurred during the gossiping process The node that decreases STATION_NUM and POLLS_NUM values to 0 drops the blue sheet and sends an acknowledge message to all nodes within the network using the gossip protocol. If the issuer receives a timeout message before an acknowledgement message, it sends a new STATE of `drp_err` which is encrypted with the issuer's private key. Hence all members can verify the authenticity of the claim to drop the election process with the issuer's public key which comes with the headers of the blue sheet. During the unidirectional spread of the blue sheet, the STATE is set to `idle`. After the acknowledgement message is received, each node whom the election process might concern changes state to `active`. In this state, nodes are ready to accept votes from voters. If the STATE is anything other than `active`, the votes that are sent into the network will be rejected and voters will be able to vote again. All nodes with same station number forms a logical network on top of the bigger network. This helps abstract real life constituencies. While the working nodes serves as polling stations.

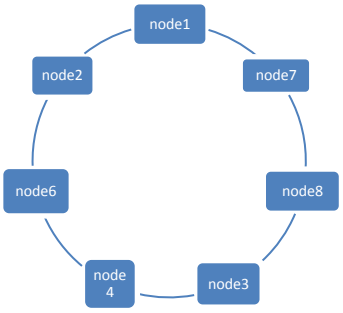

Fig1. Blockchain network of nodes

3.3 Election Network

Election networks are formed on demand. Members of the Mesh Network listens for newly issued elections. When a member of the network issues a new election (blue sheet), this will be passed to each node. The issued blue sheet contains details of how the election network should be formed, the duration of the elections, the candidates, current state of the election and a list of all valid voters.

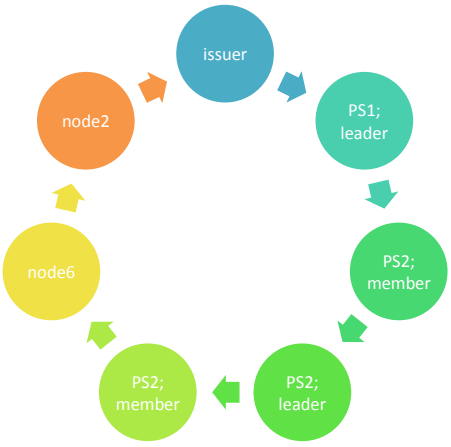

Fig 2. Issued blue sheet with 2 polling stations and two polling nodes

3.4 Issuing Elections (Blue Sheet)

Votes are issued by a single entity within the network. The issuer hence serves as the commissioner of the given election. The blue sheet is distributed through a unidirectional channel. The issuer sends the initial blue sheet to a single entity in the network whom he is connected to. The first recipient reads the header of the blue sheet for the MAX_STATIONS, STATION_NUM, MAX_POLLS, POLLS_NUM values which are integer values that stands for maximum number of polling stations, the current polling station number, the maximum number of validators each polling station can have and the current validator's number for STATION_NUM respectively. As the blue sheet is passed around within the network in a unidirectional flow, the POLLS_NUM is decreased until we reach a

stations. Each voter is assigned a polling station within the network by the election issuer. Each polling station has its own set of voters. Hence verifiers of a different polling station cannot authenticate voters of another polling station. Thus if a voter is not a member of a specific polling station, the verifier assumes that the voter is not legitimate. This can also be said in another way that, the verifier will have no knowledge of the voter in its database of valid voters hence will drop the user as an unauthorized voter. If the verifier gets a voter that belongs to its polling station, it first verifies the authenticity of the

voter and finally checks if the voter has voted just once. If both conditions are true, the voter's casted votes is added to the ballot papers (transactions). If any is false, the vote is discarded.

We check for the authenticity of the votes through the use of mekle trees. The issuer hashes the details of each voter and adds it to a mekle tree. There must be a number of mekle trees each being assigned to a particular polling station. When a node joins a particular polling station, it retrieves the assigned mekle trees which can be one or more depending on the size of the voters. It is recommended that each polling station should have about 1000 voters in each mekle tree and authentication can be done concurrently to improve performance. Each verifier also creates an empty immutable mekle tree. This new mekle tree must be an empty one. And must be immutable in order to avoid unprecedented mistakes with mutability. Whenever an authorized voter casts his votes, the issuer adds the details of the voter to its immutable mekle tree. This will be done across the cluster of verifiers since each verifier will have his own unique copy of the voter's vote. The immutable list is then used to authorize the voter as voting once and only once. If the voter has already voted, it will be verified in the immutable mekle tree. This makes the whole referencing process simple and effective. Each voter must belong to a single polling station. If a voter has more than one polling station assigned to him, the voter is deemed illegitimate and any vote from the voter is discarded.

3.5 Vote Casting

Voters are eligible to cast valid votes after the current state of election network is `active`. All votes that will be casted during the `active` state must be counted once and only once. Any subsequent votes by same voter will be deemed invalid. Voters can access the election network over protocols such as HTTP. In this network, users are given access to an HTTP web server that serves as a gateway into the MESH NETWORK. The server is supposed to contact different nodes randomly on every new vote casted. If a node is not a member of a particular election, it just passes the data to all other nodes it is connected to directly. Each vote will be received by every node of the election group. If a node of the given election receives the election details, it tries to authenticate the user and then saves the data into memory until a consensus occurs for the block of transactions to be accepted and written into database. This occurs after the election timeout is reached as found in the blue sheet header. Hence vote counting happens once within the lifetime of the election process.

3.6 Voter Authentication and Authorization

Voter verification is done by verifiers within the system. Verifiers can be seen as analogues to polling

3.7 Consensus and Votes counting

The issuer sets a timeout in its issued elections headers. When that timeout is reached across the cluster, all nodes changes their current state to `elt_comp` which stands for a completed elections process. When the state is set to `elt_comp`, each node in the election system counts their votes. After the votes are counted, a consensus is reached through the use of **PROOF OF LEADERSHIP.** After the consensus is reached, the members of each polling station submit their counted election data with a sealed consensus count that has been accepted in the network. The issuer then is at liberty to do with the data as he pleases. The election state is now set to 'done' and hence the process is completed.

3.8 Proof of Leadership

This is a simple proof that is used to enable nodes within a polling station to come to a consensus. This proof states that 'the first member of a polling station is automatically selected as the leader of the polling station'. This is a random occurrence since the leader of a polling station cannot be determined beforehand. This is because we can't determine who the issuer is and the nodes that are connected to the issuer directly are connected to it randomly. Again each node chooses the node to pass the blue sheet to randomly.

3.9 Non-repudiation and integrity protection

Each new issued blue sheet will come with a different asymmetric key pair. The issuer will keep the private key and the public key will be distributed to the polling station nodes. All data that flows from the issuer to the polling stations must be encrypted with this private key and vice-versa. This ensures the issuer and polling station nodes communicate secretly between themselves and the issuer cannot deny creating the election process. Again data that is sent from the voters must be readable by only polling station nodes or it can be made open ended. If it is only polling station nodes that can gain access, a special protocol must be built on top of the network to ensure only members of the polling station can understand that language. This protocol must be simple so it does not impact performance too much.

3.9 Assumptions

The system assume building our system on top of blockchain network with all details of existing blockchain features being viable. Hence the voting system must be seen as an abstraction on top of the blockchain and not a new definition to the blockchain technology.

4. STRUCTURE OF BLUE SHEET

Blue sheet is a data structure that defines the election header, election public key, election candidates.

Table 1. Blue sheet general structure

Size	Field	Description
4 bytes	Sheet size	The size of the blue sheet in bytes
Variable	Sheet Header	Several fields describing the sheet
1-9 bytes	Candidate count	Count of all candidates
Variable	Candidates list	All candidates
1-9 bytes	Mekle count	Count of validation mekle tree
Variable	Mekle validation tree	List of validation data for voters
256 bytes	Hashed public key	Public key for the process

Table 2. The blue sheet header

Size	Field	Description
4 bytes	Version	Current version being used
4 bytes	Max_stations	Total number of stations to use
4 bytes	Station_num	The current station number. It is set to Max_stations by default
4 bytes	Max_polls	Max number of nodes in each staion.
4 bytes	Polls_num	Current poll node. It is set to Max_poll by default
4-10 bytes	State	Current state of the election process. Set to 'idle'
4 bytes	Timestamp	The time the blue sheet was released
4 bytes	Net_time	The time given to create a complete network.
4 bytes	Election_time	Total time after net_time for voting

5. STRUCTURE OF THE CASTED VOTE

Table 3. The voter's data general structure.

Size	Field	Description
4 bytes	vote size	The size of the vote casted
4 bytes	Poll_id	Polling station the voter belongs to
256 bytes	Identifier	The hash of the voter's details
Variable	Votes	The votes casted by the voter
256	Hash public key	Voters public key

6. CONCLUSION

The system is expected to make use of the block structure of a blockchain technology to write all the accepted data into the chain of blocks. Thus the general structure of the blockchain still applies and this paper serves as an abstraction on top of the blockchain structure.

7. REFERENCES

[1] Adida, B.: Helios: Web-based open-audit voting. In: Proceedings of the 17th Conference on Security Symposium. pp. 335–348. SS'08, USENIX Association, Berkeley, CA, USA (2008)

[2] Cryptographic Key length recommendation (2015) https://www.keylength.com

[3] Fridrick P. Hjalmarsson, Gunnlangur K. Hreidarsson, (2018) "Blockchain-based E-Voting System"

[4] Gerlach, J., Gasser, U.: Three case studies from Switzerland: E-voting. Berk- man Center Research Publication No. 2009-03.1 (2009)

[5] Gjøsteen, K.: The Norwegian internet voting protocol. IACR Cryptology ePrint Archive, Report 2013/473 (2013)

[6] Hackamoon (2018). "Blcokchain for voting and elections"

[7] Hasinchun C, Storey V, "Business Intelligence and Analytics (2012)

[8] McCorry P., Siamak F. Shahandashti and Fengo H. (2017). "A smart Contract for Boardroom voting with Maximum voter privacy"

[9] Neumann, S., Feier, C., Sahin, P., Fach, S.: Pretty understandable democracy 2.0. IACR Cryptology ePrint Archive, Report 2014/625 (2014)

[10] Philip B., "How blockchain technology could change our lives",

[11] Sagar S., Qaish K., Huaiqian M. "Block Chain Voting System"

[12] Sandra G. "Individual Verifiability in Electronic Data"

[13] Storer, T.W.: Practical pollsterless remote electronic voting. PhD Thesis. Uni- versity of St Andrews (2007)

[14] Vitalik B. (2015). Ethereum White Paper.