ELECTRONIC VOTING SYSTEM USING THUMB VERIFICATION

SHAHANAS BEGAM S K[1], SHAMRUN FATHIMA M[2],
SHANMUGAPRIYA S[3],SHARMILA DEVI M[4]
Asso.Prof.Mrs P Vijayasarathy
Krishnasamy College of Engineering and Technology, Cuddalore.

Abstract—For the current scenario voters are cast their votes through electronic machine which has the details of casted vote count of each candidate. After some time that machine connect with the single system it shows total count of an each area.in this system may have chance to poll proxy votes as well as security problem.in our country this is a major issue in voting system. to overcome this problem we planned to apply a blockchain model in e voting system using edge computing . because of this proxy votes will be blocked and we can achieve data security .this method will work fine even in low latency.

1. INTRODUCTION

Electronic voting is among the key public sectors that can be disrupted by blockchain technology. The idea in blockchain-enabled e-voting with biometric is simple. Usages of new technology in the voting process improve the elections innatural. This new technology refers to electronic voting systems where the election data is recorded, stored and processed primarily as digital information.

The biometric is a technology of measuring, science and it analyze the biological data. In the modern communications approximately it has accessible electronically, users of computer technology, it has increment in electronic services and with the security system. It improves in the election system with the help of new technologies in voting process. The information about election data is stored, recorded and processed the above information as a digital information. In olden days the information security is with the help of military and instructions of the government. The human body characteristic like DNA, fingerprints, voice patterns and hand measurements is used for authentication purpose. The e-services and information security are making sure that data, communication, have the security and privacy enable.

The electronic voting machines are using in Indian general and state elections to implement electronic voting. This machine can reduce the time of voters and easy to count by comparing the ballot system. In earlier days there were rumors about the EVM's but as per the Delhi court and supreme court the electronic voting machines are using. The following image shows the electronic voting machine.

process is automated and does not require the validators to be constantly monitoring their computers. A permissioned blockchain which uses the POA consensus algorithm enables us to set restrictions on a set of selected known entities to validate and certify transactions on the blockchain and censor transactions arbitrarily, with their identity and reputation at stake. This otherwise needs to be done by

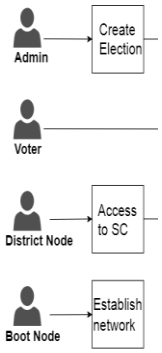

Fig. 1: Election roles and process

3. BLOCKCHAIN AS A SERVICE FOR E-VOTING

In this paper, we consider existing electronic voting systems, blockchain-based and non-blockchain-based, and evaluate their respective feasibility for implementing a national electronic-voting system. Based on this, we devised a blockchain-based electronic voting system, optimizing for the requirements and considerations identified. In the following subsection, we start by identifying the roles component for implementing an e-voting smart contract then, we evaluate different blockchain frameworks that can be used to realize and deploy the election smart contracts. In the last subsection, we will discuss the design and architecture of the proposed system.

A. Election as a Smart Contract

Defining a smart contract includes identifying the roles that are involved in the agreement (the election agreement in our case) and the different components and transactions in the agreement process. We start by explaining the election roles followed by the election process.

1) Election Roles: As can be seen in Figure 1, elections in our proposal enable participation of individuals or institutions in the following roles. Where multiple institutions and individuals can be enrolled to the same role.

(i) companies are enrolled with this role. The election administrators specify the election type and create aforementioned election, configurate ballots, register voters, decide the lifetime of the election and assign permissioned nodes.

(ii) **Voters:** For elections to which they are eligible for, voters can authenticate themselves, load election ballots, cast their vote and verify their vote after an election is over. Voters can be rewarded for voting with tokens when they cast their vote in an election in the near future, which could be integrated with a smart city project.

(iii) **District nodes**: When the election administrators create an election, each ballot smart contracts, representing each voting district, are deployed onto the blockchain. When the ballot smart contracts are created, each of the corresponding district nodes are given permission to interact with their corresponding ballot smart contract. When an individual voter casts his vote from his corresponding smart contract, the vote data is verified by all of the corresponding district nodes and every vote they agree on are appended onto the blockchain when block time has been reached.

(iv) **Bootnodes:** Each institution, with permissioned access to the network, host a bootnode. A bootnode helps the district nodes to discover each other and communicate. The bootnodes do not keep any state of the blockchain and is ran on

a static IP so that district nodes find its peers faster.

2) Election Process:In our work, each election process is represented by a set of smart contracts, which are instantiated on the blockchain by the election administrators. A smart

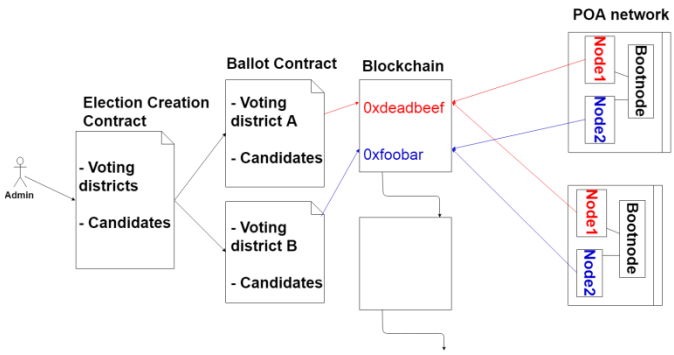

Fig. 2: Election as a smart contract

contract is defined for each of the voting districts of the election so multiple smart contracts are involved in an election. For each voter with its corresponding voting district location, defined in the voters registration phase, the smart contract with the corresponding location will be prompted to the voter after the user authenticates himselfwhen voting.

The following are the main activities in the election process:

(i) **Election creation:** Election administrators create election ballots using adecentralized app. This decentralized app interacts with an election creation smart contract, in which the administrator defines a list of candidates and voting districts. This smart contract creates a set of ballot smart contracts and deploys them onto the blockchain, with a list of the candidates, for each voting district, where each voting district is a parameter in each ballot smart contract. When the election is created, each corresponding district node is given permission to interact with his corresponding ballot smart contract (See Figure 2).

(ii) **Voter registration:** The registration of voter phase is conducted by the election administrators. When an election is created the election administrators must define a deterministic list of eligible voters. This requires a component for a government identity verification service to securely authenticate and authorize eligible individuals. Using such verification services, each of the eligible voter should have an electronic ID and PIN number and information on what voting district the voter is located in.

(iii) **Vote transaction:**Each transaction on the blockchain holds information about whom was voted for, and the location of aforementioned vote. Each vote is appended onto the blockchain by its corresponding ballot smart contract, if and only if all corresponding district nodes agree on the verification of the vote data. When a voter casts his vote, the weight of their wallet is decreased by 1, therefore not enabling them to vote more than once per election. As can be seen in Table I, a single transaction on the public Ethereum blockchain includes the transaction ID, the block which the transaction is located, the age of the transaction, the wallet which sent the transaction and who received it, the total value which was sent and the transaction fee. A transaction in our proposed system doesn't require all of this information, a single transaction only has information of the transaction ID, the block which the transaction is located at, to which smart contract the transaction was sent, in this example N1SC indicates that the vote was sent from the N1 district. Finally the

value of the transaction is the data which was selected to cast, D therefore indicates that the vote casted in this transaction was the the party D. A transaction in our system (see Table II) therefore reveals no information about the individual voter who casted this particular vote. The age of a single transaction is excluded to protect individual voters from a timing attack.

TABLE I: Example of an public transaction(Ethereum)

TxHash	Block	Age	From	To	Value	[TxFee]
0xdead...	1337	33 sec ago	0xbeef...	Token	10 Ether	0.087
0xface...	1337	33 sec	0x4242...	0x1234...	1 Ether	0.056

TABLE II: Example of an transaction in our system

TxHash	Block	To	Value
0xdeadbeef...	1337	N1SC	D
0xG1345edf...	1330	N2SC	P

(iv) **Tallying results:** The tallying of the election is done on the fly in the smart contracts. Each ballot smart contract does their own tally for their corresponding location in its own storage. When an election is over, the final result for each smart contract is published.

(v) **Verifying vote:** As was mentioned earlier, each individual voter receives the transaction ID of his vote. Each individual voter can go to his government official and present their transaction ID after authenticating himself using his electronic ID and its corresponding PIN..

4. Index of functionalities

Below, we will elaborate the functionalities of a novel ballot and election smart contract for an e-voting system, without the integration of a government indentity verification service.

- **Ballot constructor:** Sets the manager of the ballot smart contract to the address of the wallet which created the election, the voting district of the smart contract to the district which the ElectionCreation contract provided and then proceeds to fill the Candidates struct with the list of candidates provided and the number of votes for each candidate to 0. The constructor also stores the time of the creation of the contract along with the time when the contract is to expire.

```
function vote(uint candidate) public{
    require(!voters[msg.sender]); if(now >
    candidates[candidate].expirationDate){ revert();
    } candidates[candidate].voteCount+= 1;
    voters[msg.sender] = true;
}
```

- **vote:** This function allows voters to vote. The requirement for a voter to vote, is that the mapping of the address of the voter is set to its default, false. If that is the case, the function guarantees that the election time limit has not been reached. If both requirements are satisfied, the contract retrieves the index of which candidate was voted for and increases his vote count by 1 and sets the mapping to true, so that the voter can never vote again in this particular election.

```
function getCandidateName(uint index) public
restricted view returns (bytes32)
```

```
    { require(now >candidates[candidate]
        .expirationDate) } return
        candidates[index].name;
    }
    function getVoteCount(uint index) public
    restricted view returns (uint)
    { require(now >candidates[candidate]
        .expirationDate) return
        candidates[index].voteCount;
    }
}
```

- **getCandidateName&getVoteCount:** Both these functions retrieve the name and amount of votes a candidate has recieved from an index. These functions classify as helper functions to determine the election results after the election is finished

4. SECURITY ANALYSIS AND LEGAL ISSUES

In this section we analyze the security of the proposed evoting system and the main legal issues.

A. Security analysis

1) ***Authentication vulnerability:*** Each individual is identified and authenticated by the system by presenting an electronic ID from Auðkenni and the corresponding 6-digit PIN in the voting booth. Without supervision, an individual could vote for multiple people, if the individual had knowledge of the PIN for each corresponding electronic ID he has. To further address this vulnerability in the near future, a biometric scan could be introduced.

B. Legal issues

1) ***Remote voting:*** Remote elections provide no coercion resistance because of the non supervised factor in a remote election. Remote elections can therefore not guarantee the privacy that people have when they cast their vote in a voting booth. If elections are hosted on a website, for eg. It could be easily taken down by people with good hacking skills and the mindset to do so.

2) ***Transparency:*** In the today's election scheme, no method of transparency can beoffered to participants of the election.When an individual places his ballot in the box at his voting district, there is no guarantee from the scheme that his vote was counted and counted correctly. Any individual vote can be misplaced, counted incorrectly because of human error.

3) ***Voter privacy:*** In every pen and paper election scheme, voters privacy is a key element. The law forbids any individual or entity to be able to know from a single vote, who gave aforementioned vote. If such information could be gathered for each vote, such information could then leak to the public which would allow for listing every single individual who voted for a single party/candidate. To satisfy the privacy of each voter, no individual vote should be traceable back to the voter.

5. RELATED WORK

In this chapter we will be examining various research papers and thesis which explored similar fields of study, i.e electronic voting systems.

Anonymous voting by two-round public discussion, proposed an addition of a self-tallying function to the 2-Round Anonymous Veto Protocol (called AV-net). The AV-net provided exceptional efficiency compared to related techniques, the paper was focused on the dining cryptographers network (DC-net) and its

weaknesses and proposed the AV-net as a new way to tackle that problem.

This protocol is divided electronic voting into to two classes:

1) Decentralized elections where the protocol is essentially run by the voters.
2) Centralized elections where trusted authorities are employed to administer the process.

The protocol proposed was focused on the first class, where strong voter privacy was the primary objective which had two challenges. First challenge was that there exists no trusted third party. With a trusted third party, many security problems can be easily solved, but could lead to the 'trusted' third party to become the one who breaks the security policy. The goal therefore was to eliminate the use of a trusted third party altogether. The second challenge was that there would be no voter-to-voter private channels to ensure dispute freeness, i.e everybody could check whether all voters had followed the protocol faithfully.

These challenges were fulfilled in the AV-net, but the new protocol proposed a new solution which solved the downside of the AV-net, heavy computational load for each voter, which increased linearly with the number of voters. The first round in the two-round protocol consisted of every participant to publish his public key and a zero knowledge proof (ZKP) for his private key. When the round finished, each participant checks the validity of the ZKPs and computes.

A Secure and Optimally efficient Multi-Authority Election Scheme, proposed a multi-authority secret-ballot election scheme which would guarantee privacy, universal verifiability and robustness, where voters would participate using a PC, where the main consideration is the effort required of a voter.

In this model, voters cast their vote by posting ballots to a bulletin board. The bulletin board works as a broadcast channel with memory to the extent that any party can access its content but no party can erase anything from the bulletin board. The ballot does not reveal any information on the vote itself but is ensured by an accompanying proof that the ballot contains a valid vote. The final tally, the sum of all votes, which occurs when the deadline is reached, can then be obtained and verified, by any observer, against the product of all submitted ballots. Which would ensure universal verifiability, due to the homomorphic properties of the encryption method used.

6. CONCLUSION

For over a century, fingerprints have been one of the most highly used methods for human recognition; automated biometric systems have only been available in recent years. This work is successfully implemented and evaluated. The arrived results were significant and more comparable. It proves the fact that the fingerprint image enhancement step will certainly improve the verification performance of the fingerprint based recognition system. Because fingerprints have a generally broad acceptance with the general public, law enforcement and the forensic science community, they will continue to be used with many governments,, legacy systems and will be utilized in new systems for evolving applications that require a reliable biometric. Thus the advent of this biometric voting system would enable hosting of fair elections in India. This will preclude the illegal practices like rigging. The citizens

can be sure that they alone can choose their leaders, thus exercising their right in the democracy.

REFERENCES

[1] Sos.ca.gov. (2007). *Top-to-Bottom Review | California Secretary of State. Available at: http://www.sos.ca.gov/elections/voting-systems/oversight/ top-bottom-review/.*

[2] Nicholas Weaver. (2016). *Secure the Vote Today. Available at:https:// www.lawfareblog.com/secure-vote-today.*

[3] Feng Hao, P.Y.A. Ryan and Piotr Zielinski. (2008). *Anonymous voting by two-round public discussion.* Available at: http://homepages.cs.ncl.ac. uk/feng.hao/files/OpenVote_IET.pdf

[4] Feng Hao and Piotr Zielinski. *A 2-Round Anonymous Vote Protocol*Availableat: http://homepages.cs.ncl.ac.uk/feng.hao/files/av_net.pdf .

[5] The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. Available at: https://users.ece.cmu.edu/~{}adrian/ 731-sp04/readings/dcnets.html.