BlockVOTE : An Architecture of a Blockchain-based Electronic Voting System

Chinnapong Angsuchotmetee[1] and **Pisal Setthawong**[2]

ABSTRACT

Voting is an essential activity in modern democracy. To facilitate the voting process, there were several attempts on proposing an electronic voting system such that, the voting and tallying processes can be done efficiently and the results would be accountable to the public. To date, however, an online electronic voting system has been rarely adopted in practice due to the possibility of having the voting result tampered through vote-rigging or cyber-attacking. In 2009, the blockchain algorithm was proposed by Satoshi Nakamoto. Blockchain is a technique for recording transactions between self-auditing ledgers in an open, distributed, permanent, and verifiable manner. Even though blockchain was originally designed for a financial applications, it is possible to apply blockchain to other domains, including in the implementation of an online decentralized-based electronic voting system. In this study, the architecture of a blockchain-based electronic voting system, named *BlockVOTE*, is proposed. The architecture design and all related formal definitions are given. To validate the proposal, two BlockVOTE prototypes were implemented using two different blockchain application frameworks. The performance analysis of both versions of the prototypes are given. The analysis of both technical and management aspects on the possibility of adopting the proposed decentralized voting system in an actual voting scenario is also given at the end of this study.

Keywords: Blockchain, Voting System, Electronic Voting Systems, Decentralized Applications

1. INTRODUCTION

Democracy is arguably considered the best form of government. Democracy as a system provides safeguards against the total corruption of those in power and guarantees the citizens the rights and freedom of the majority. One of the mechanisms in democracy is the process of elections. Election is a process in which the population chooses individuals to hold public office, in which the elected individual will represent the needs of the public usually though the voting process.

As the voting process itself is considered as one of the most important mechanisms of democracy, it is important that the voting system used should be as accurate, efficient, and effective as possible. From the past, voting systems are usually based on a centralized system, in which there is a voting authority that oversees organizing voting, tallies the voting, validates the voting, and declares the results of the voting. The process of holding elections of a huge population is considered a time-consuming, expensive, and fraud-prone undertaking, but a necessary undertaking in the democratic process.

There had been attempts to improve the voting process by application of technology [1] by using electronic voting systems and online voting system to help streamline the centralized voting system, but the adoption has not been widespread due to a myriad of reasons ranging from the cost, lack of flexibility, oversight approval, and the lack of infrastructure. Another concern of electronic voting systems relies mostly on a centralized-based architecture. Voting results could potentially be tampered by personnel who have access to the electronic voting systems, or could potentially be tampered by cyber-attacks if the system is online.

Due to the challenges, many elections still rely on simple paper-based ballot systems in voting, and technology aids in an election has not been used extensively. As many elections still utilize traditional voting systems, there are many issues that persist. The issues of fraud, voting authority bias, expensive undertaking, and slow process of tallying the voting are issues that will plague voting with these traditional systems. In an ideal case, a voting system should be efficient as an online voting system and the result to be resilient to tampering similarly to an audited offline ballot-based voting system.

In 2009, an emerging financial technology so-called *Bitcoin*, has been proposed by Satoshi Nakamoto [2]. Bitcoin allows financial transactions, typically processed in a centralized manner, to be processed in a decentralized approach through the underlying mechanism called the *blockchain*. The blockchain mechanisms enable users to make and record transactions in a distributed, efficient, permanent, non-violable, and verifiable manner. By adopting blockchain, users

Manuscript received on December 1, 2019 ; revised on March 30, 2020.
Final manuscript received on April 1, 2020.
[1] The author is with Department of Computer Science, Faculty of Science, Prince of Songkla University, Songkhla, Thailand., Email: chinnapong.a@psu.ac.th
[2] The author is with Department of Management Information Systems, Assumption University, Samut Prakarn, Thailand., Email: pisalstt@msme.au.edu
DOI: 10.37936/ecti-cit.2020142.227455

in Bitcoin can exchange cash directly in a peer-to-peer manner while every transaction is guaranteed to be trustable without relying on a centralized control from a central banking system. The capability of recording global transactions in a distributed, yet secured and verifiable manner, makes researchers and industrialists have applied blockchain approaches in many different applications domain, not limited to only financial application. For example, a study in [3] applies blockchain in a medical data sharing application. A study in [4] applies blockchain in a logistic monitoring application. To date, however, the suitable method, design, and implementation of the application of blockchain in an electronic voting system are still being proposed.

In this study, the requirements for a suitable decentralized-based electronic voting system are identified. The proposal of the architecture design and all the related data models of such a system are described next. The proposed architecture is named the *BlockVOTE: A Blockchain-based Electronic Voting System*. The implementation detail of *BlockVOTE* is also given and the validation experiment has also been conducted.

The organization of this paper is as follows. Section 2 describes the motivation scenario. The selected scenario is a *political election*. Challenges on proposing a high-security voting system are given and discussed against the given scenario in this section. This section follows by Section 3 which describes related studies and state of the art of blockchain. The proposed BlockVOTE is described in Section 4. The implementation and validation experiments of BlockVOTE are described in Section 5. The experiment results and challenges are discussed in Section 6. Section 7 concludes the study.

2. MOTIVATING SCENARIO: A POLITICAL ELECTION

An election is a process in which the population chooses individuals to hold public office. The elected individual will represent the needs of the public usually through the voting process. In general, a central voting authority is responsible for overseeing, organizing, validating, and announcing the result of the election in a centralized manner. The choice of how the voting system is organized can be varied depending on the choice of the voting authority of each organization or country. In general, there are three main types of the voting system which are (i) ballot-based voting system, (ii) electronic-based voting system, and (iii) online voting systems. The votings systems are centralized in nature and are described as follows.

2.1 A Ballot-based Voting System

The most common centralized voting system is a ballot system. In a ballot system, every authenticated person that is approved by the voting authority would be considered a voter. Each of the voters would be given a ballot, which is usually a small piece of paper. The voter would then select one of the possible choices before casting their ballot into a box at a polling station. An example ballot taken from the political election of Thailand is given in Figure 1.

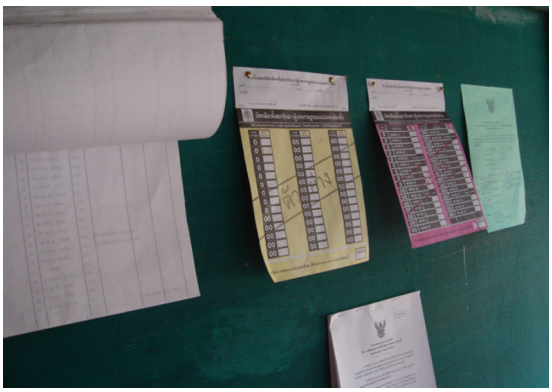

Fig.1: *Ballots used in the political election of Thailand*

After the voting period is over, the voting authorities would usually carry the boxes that are filled with the voters ballots and bring them to a public place that was reserved for voting tallying. The boxes would be open, and each of the ballots would be checked for correctness, and the votes would be classified publicly. To help authenticate the tallying process, stakeholders in the voting would usually be presented to witness the vote tallying process.

Traditional ballot-based voting systems suffer from many disadvantages. For the voting process, the preparation of a paper-based voting ballot is an expensive endeavor. The paper ballots must be prepared beforehand to account for all potential voters for every polling station. These ballots are not reusable. Preparing the ballots, in turn, becomes an expensive undertaking that is also time-consuming to print out all the ballots. Another major issue found is during the voting tally. The voting tally is a manually intensive process. Due to the nature of the tally process, it is time-consuming to examine each of the ballots, classify the ballots, and tally the votes. The tallying process itself becomes time-consuming, error-prone, and is easy to tamper with if the voting authorities are biased and there are no stakeholders monitoring the tallying process.

2.2 An Electronic-based Voting System

To improve paper-based ballot systems, the usage of electronic voting systems is one possibility. Electronic voting systems utilize electronic ballot machines or electronic voting machines (EVM) in voting. These machines are electronic devices that are used in balloting purposes and could help improve tradi-

tional ballot-based systems in the voting and/or the tallying process.

EVM comes in many types and forms [5]. Simple EVM may provide a small degree of automation such as allowing voting in the machine and the machine will prepare the ballot that would be placed in a box for later tallying. Some of the EVM stores the voting results in electronic format, allow faster tallying. Some of the EVM systems allow even the usage of private networks, or Internet services such as the transmission of tabulated results after voting results, or even in online voting systems that would be discussed in the next section. An example of an EVM is depicted in Figure 2.

The usage of EVM must comply with the set of standards that are established by the voting authorities, and be approved for usage after extensive tests. The EVM in nature would have detailed requirements that should offer a high degree of security, privacy, accuracy, swiftness, verifiability, accessibility, and scalability.

Though EVM offers many advantages, the high cost of EVM machines is one major factor behind the slow adoption of EVM in voting systems. Another issue that has been widely reported is that many EVM has been poorly implemented, and voting fraud was reported on the platform [6]. This should not reflect against the performance of EVM in general as the issues of voter fraud are usually from poorly implemented security schemes behind the EVM machine that could be rectified with more secure software and hardware design [7].

Fig.2: *Example of Electronic Voting Machine from the Political and Electoral Development Institute, Thailand*

2.3 Online Voting Systems

A variation of electronic voting systems is to allow the voting to be conducted online, via online network systems such as the Internet [8][9] or private networks that are set up by the voting authorities. By utilizing online voting systems, there are many advantages when compared to electronic voting systems and traditional ballot systems. Due to the online nature of the voting systems, the ballots are done online, which removes the expensive requirements of the preparation of the ballots found in traditional ballot systems. When tallying the results, the votes would be sent to the central server, which makes the tallying process simple, efficient, and effective when compared to early systems.

Though online voting systems offer many advantages over older centralized voting systems, the voting authorities must tackle an increased number of technical issues. One of the issues is the implemented voting system and server infrastructure. The system should be robust enough to authenticate and allow many simultaneous users to cast their votes without significant delays or errors. The system should also be robust enough to prevent unauthorized access to the server and the results of the voting. The security aspects of online voting systems are extremely complex, contain many points of attacks and vulnerability, in which attackers only need to exploit one of the points to change the voting results or potentially invalidate parts or all of the voting [9]. Attacks such as a distributed denial of service (DDOS) attacks can disable the election network. Attacks on the DNS can potentially cause election stations to be unable to route the voting to the election server. Malware installed on client-side machines can cause the votes to be incorrectly cast. Unauthorized access on the voting server could allow modifications to the voting, leading to a different result. Due to the complexity, there are many attacks that could be done, and the attacks could be difficult or even impossible to recover from.

When considering security, it is easy to explore only external threats to online voting systems. However, the paradox is that the most dangerous security threat of online voting systems is from the voting authorities themselves. It is possible for members from the voting authorities with access to the server to conduct voting fraud. Voting fraud can consist of removing or manipulating the results of the voting. This type of voting fraud cannot be easily discovered by stakeholders when infractions happen, and there must be complete trust with the voting authorities. This issue would not happen if the voting authorities could be trusted. If not, additional features on the voting system would have to be implemented to allow the back checking of the voting results. This itself is a complex system if the requirement of voting anonymity must be maintained.

2.4 Challenges: A High-Security Voting System

In an ideal case, every election prefers a voting system that is guaranteed to be highly secured and tampered-proof, while the tallying process and result announcing are kept to be as efficient as possible. Based on the existing systems, none of the major approaches would be able to provide all the aspects required. Traditional ballot systems are expensive and time-consuming but provides some form mecha-

nisms against tampering. EVM addresses some of the issues of making the tallying process more efficient, but some implementations are prone to tampering. Online systems though offering many advantages are generally easier to tamper and offer less accountability than other approaches.

Due to the shortcomings of existing approaches, the proposal of an alternative approach is recommended. The new approaches should be able to address the following challenges:

- *Securing Network Infrastructure:* A highly secured electronic voting system must ensure that all network infrastructure related devices must be able to withstand all kind of cyber-attacking;
- *Securing Voting Data:* A highly secured electronic voting system must keep all the voting data in such a way that the result can always be available for tallying while ensuring that all the data cannot be tampered either through cyber-attacking nor cheating by internal officers;
- *Trust Management:* A highly secured electronic voting system must be able to prove to every voter that all the vote records and results are 100% reliable and trustable to all related parties.

3. RELATED STUDIES

This section is dedicated to discussing related studies on blockchain and its application. This section begins with the introduction to blockchain and applications related to blockchain. It follows by the survey and discussion on the existing studies which propose to use blockchain in a voting system.

3.1 Blockchain: Background

The term *Blockchain* refers to the algorithm for handling transactions between peer-to-peer ledgers in such a way that every ledger within the network can all agree on the same consensus without relying on a centralized server. At first, blockchain was designed to be used for supporting transaction processing in Bitcoin [2]. The main idea of blockchain and Bitcoin are depicted in Figure 3.

The main purpose of Bitcoin is to propose a decentralized banking system where each transaction between ledgers can be made directly without relying on a centralized server for validating transactions. In a centralized system, there is a central authority that validates transactions in the system. In a financial system, there may be an authority such as a central bank. In Bitcoin, instead of relying on a central bank, other ledgers in the Bitcoin ecosystem will attempt to validate the validity of the transaction on their own. The reason why the algorithm behind Bitcoin is named *blockchain* is the fact that each transaction within the ecosystem is modeled as a *Block*. A newly created block can only be connected to the shared global chain of blocks only if the block is validated by all ledgers within Bitcoin ecosystem to be

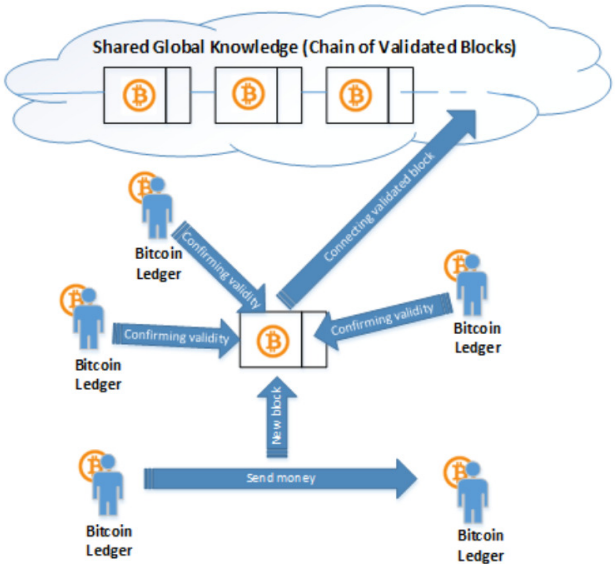

Fig.3: *Bitcoin: Architecture*

valid. This helps Bitcoin to keep all transactions to be highly-secured, non-violable and tampered-proof without relying on a central bank to validate a transaction.

Blockchain is proven to be effective in making Bitcoin to be a decentralized cryptocurrency-based ecosystem which ledgers can freely exchange their cryptocurrency credits without relying on any centralized-based banking system while ensuring that every transaction made is secure and trustable to every ledger [10]. Such a capability of blockchain makes both researchers and industrialists envisioned that it is possible to apply blockchain in other application domains. However, the original proposal of *blockchain*, as proposed in [2], cannot be extended to support a business logic that is more complex than a simple financial transaction. This has led to the extension of *blockchain* algorithm such that, a block can be embedded with a built-in programmable function to support a custom business logic. Such an extension is regarded as a *smart contract* feature of *blockchain*. A *smart contract* is an autonomous process that is capable to regulate the flow of transactions within a blockchain network such that a specific set of programmable instructions that every ledger is agreed on can be executed each time a predefined action or events within blockchain network are made [11].

Nowadays, blockchain and smart contract are deemed to be suitable for storing global knowledge and business logic for any application domain in a distributed manner, while maintaining the security and the privacy of the stored knowledge. Some example application domains which utilizes blockchain include medical record storage [12], supply chain management [13], video-on-demand [14], and authentication systems [15].

3.2 Blockchain in a Voting System

The possibility of adopting blockchain in a voting system is mentioned in several studies [16][17]. To date, even though blockchain is still recognized by as an *unnecessarily complicated technology*[1], its benefits on being able to secure all the result, ensuring anonymity and yet tampered proof, can still attract researchers to study the possibility on adopting blockchain to a voting system in practice.

One of the significant study in recent years is in [18]. This study is one of the earliest which proposes the usage of a smart contract mechanism for using in an election. The preliminary architecture design is presented and discussed. The suggestion on the possibility of implementing the proposed architecture in practice is also briefly described. Another significant study is in [19]. This study addresses a more precise implementation detail on using a smart contract mechanism on designing an electronic voting system. However, the proposal as given in this study is too limit to the *Ethereum* framework[2]. Hence, the architecture as proposed in [19] might not be generic enough to be adapted to other blockchain implementation frameworks, or other application domains beside a political election (e.g., internal poll, referendum).

Studies in [20][21][22] are also proposed their own design of a blockchain-based election system. Each of these studies proposes their own version of an election system using blockchain. However, most of these studies, similar to [18] and [19], do not proposed a complete design of a voting system framework which includes (i) a data model design (ii) a smart contract design (iii) generic implementation guideline such that, the system can be easily replicable or reimplemented publicly.

4. BLOCKVOTE: AN ARCHITECTURE OF A BLOCKCHAIN-BASED ELECTRONIC VOTING SYSTEM

In this study, the architecture of a blockchain-based electronic voting system called *BlockVOTE* is proposed. This section describes the architecture overview, data models, algorithms, and all related formal definitions. The details regarding the implementation of the proposed systems are later described in the next section. The architecture of *BlockVOTE* is depicted in Figure 4. The architecture is designed based on the *smart contract* capability of blockchain. To improve the clarity of the explanation, the architecture would be explained in a step-by-step manner according to the three main steps of any voting process, which are (i) Poll Creation, (ii) Voting, and (iii) Result Tallying. The details are as follows.

https://www.wired.com/story/theres-no-good-reason-to-trust-blockchain-technology/
http://www.ethereum.org

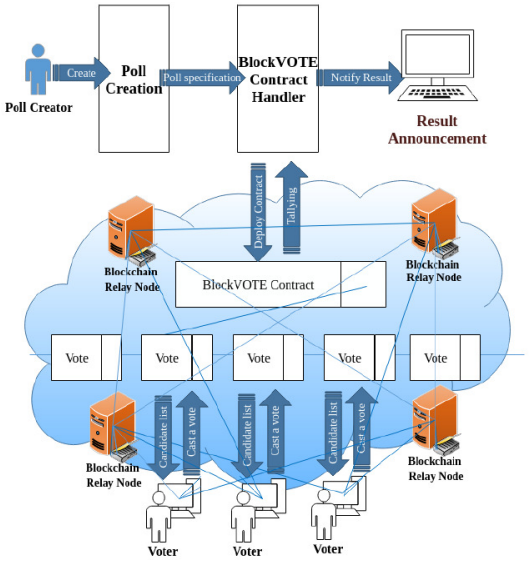

Fig.4: *BlockVOTE: Architecture Overview*

4.1 Poll Creation

Before creating a poll, a number of processes would have been accounted for. The first process would deal with the list of candidates that would be allowed to contest in the poll. This would be covered in this subsection. The second process would deal with the creation of a voter registry - who would be permitted to vote for the candidate. This process would be discussed in the Voting subsection.

The process of creating the list of candidates would require the candidates to declare their interest, and the voting authorities would have to verify the qualifications of the candidate before adding to the candidate registry list. The process of declaring the interest and checking the qualification of the candidates before approval would not be dealt by the system due to the difference between voting authorities, and the unsuitability of encompassing the process within the architecture. The architecture then would only be provided with the list of the candidate registry list before the voting.

When the list of candidates is ready, the poll creator must send the list of candidates to the *Poll Creation* (see Fig. 4) module of the architecture. This module is used for modeling the list of candidates into a machine-readable format. The data model used for modeling a candidate is described in Def 1.

Definition 1: Candidate: A *Candidate*, c, is a choice within a poll that a voter can cast a vote to. It is modeled as a 3-tuple $c =< id, desc, score >$, where:

- id: is a unique number for identifying c;
- $desc$: is a user-friendly short textual description for describing c;
- $score$: is a number representing number of votes that c got from voters.

After all the candidates are modeled, the *Poll Creation* would automatically pass the list of modeled candidates to the *BlockVOTE Contract Handler* module. *BlockVOTE Contract Handler* is responsible for creating a *BlockVOTE Contract*, which is a smart contract designed specifically for the proposed architecture. The formal definition of the *BlockVOTE Contract* is given in Def. 2.

Definition 2: BlockVOTE Contract: A *BlockVOTE Contract*, BV, is a smart contract which is responsible for managing a poll, handling vote activities, and tallying the result automatically. It is defined as a 3-tuple, $BV =< C, V, exp >$ where as:

- C: is a set of candidates defined as $C = \{c_1, c_2, c_3, ..., c_n\} \wedge \forall c_i \in C, c_i =< i, desc, score >$ (see Def. 1);
- V: is a set of voters who have cast a vote to one of candidates in C, $V = \{v_1, v_2, v_3, ..., v_n\} \wedge \forall v_i \in V, v_i$ is a blockchain-based cryptographic hash representing an identity of a given voter;
- *exp*: is a date-time value indicating that when voters can still cast a vote to C;

When *BlockVOTE Contract Handler* receives a poll creation request, it creates a new BlockVOTE contract according to Def 2, while keeping V empty, and setting *exp* according to the request from a poll creator. When a new contracted is created successfully, it is automatically deployed onto a pre-deployed blockchain-based infrastructure on the Internet cloud in such a way that, users can cast a vote to the contract. It is to be noted that the we propose to keep only the cryptographic hash identity of the voter only in order to ensure the privacy and the anonymity of the voter. The detail of the *vote* function of BV is described in the next sub-section.

4.2 Voting

The process of creating the voter registry is another important process in the poll before the voting. The voter registry should contain the list of people that are eligible for the voting and the voters would need to be authenticated. In a traditional voting scenario, the voting authorities would come up with the list of potential voters, and would be responsible for the verification of voters before they cast their votes. This process would unlikely to be changed by any voting authority due to the importance of voter registry and verification in voting, and that they are unlikely to trust external sources in this process. Voter verification by using the blockchain is the other alternative, but the approach is unlikely to be adopted by voting authorities, as that would surrender too much authority to the blockchain.

To compromise with the current situation of voting, the architecture of the proposed system proposes to defers to the voting authorities for identity verification processes so that the voting authorities could fulfill the voter validation/auditing. Each of the voters would have to be verified by the voting authorities before they can cast their votes. The degree of anonymity is not lost after the voting authorities have verified the voters as the voters would cast their votes to the blockchain without the requirement of the voting authority to oversee their votes. Based on that, the architecture adopts the identity verification process outside the voting contract, in which would fit the process better than adopting the identity verification process inside the voting contract.

Algorithm 1 Vote: A BlockVOTE built-in function

Require:

BV: a BlockVOTE contract that a voter invokes
id: an ID number of a candidate that a voter who invokes the contract wants to vote
v: a blockchain-based cryptographic hash of a voter who invokes BV

1: **if** current date-time¿ $BV.exp$ **then**
2: return $False$ ▷ Poll is expired
3: **else if** v is in $BV.V$ **then**
4: return $False$ ▷ v has already casted a vote
5: **else if** none of $c.id$ in $BV.C$ is equal to id **then**
6: return $False$ ▷ invalid candidate id
7: **else**
8: $c.score = c.score + 1$ where $c \in BV.C \wedge c.id = id$
9: add v to $BV.V$
10: return $True$ ▷ Successfully cast a vote
11: **end if**

After the poll creator uses *Poll Creation* module and *BlockVOTE Contract Handler* module to create a new BlockVOTE contract, voters can invoke the *vote* functionality of the contract to cast a vote. The functionality is defined as a built-in function which uses BV as an input. The algorithm of the *vote* function is described in Alg. 1. In short, a voter casts a vote by passing, (i) a contract to be voted, (ii) an ID of a candidate that a voter wants, and (iii) their own cryptographic hash identity, to the *vote* function. The vote is a success only if when that voter has not yet cast a vote, and provided candidate ID is valid. The vote function rejects a vote when a voter has already cast a vote, or when provided candidate ID is invalid.

Each time a vote has been cast by using a function as described in Alg. 1, a new *Block* is created. One block within BlockVOTE architecture stores one vote transaction from one voter, and the most recent state of the contract (i.e., a list of all candidates and their corresponding score received).

4.3 Result Tallying

The result tallying is a process that is done when the poll deadline has been reached. The tallying pro-

cess is modeled as a smart contract-based function that is executed automatically by the *BlockVOTE Contract Handler* module when a poll deadline is reached. The formal definition of the result tallying function is given in Def. 3.

Definition 3: Result Tallying Function: A result tallying function, tf, is a function for tallying the result of a given poll. It is defined as

$$tf(BV, cdate) = \begin{cases} false & \text{if } cdate < BV.exp \\ R & \text{otherwise} \end{cases}$$

where as:

- BV: is a BlockVOTE contract to be tallying tf, $BV =< C, V, exp >$ (see Def. 2);
- $cdate$: is a date-time when a tallying process is requested.
- R: is an ordered set, $R = \{c_1, c_2, c_3, ..., c_n\} \wedge \forall c_i \in R, c_i \in BV.C \wedge c_i.score < c_{i+1}.score$

In short, the result tallying function returns a list of candidates within a contract ordered by the score that each candidate is received in descending order. The function returns *false* to indicates that the tallying cannot be done if the poll has not yet ended.

So far, the architecture has been described by using all related formal definitions and algorithms. The following section describes the implementation of the proposed architecture.

5. BLOCKVOTE: IMPLEMENTATION & VALIDATION

To avoid developing everything from scratch, it is always a common practice to reused an existing framework or a library as much as possible. In this case, to implement BlockVOTE, we choose to implement our proposal using an existing Blockchain application framework. So far, to ensure that our architecture design is generic, we implement two separate prototypes using two different Blockchain application frameworks. Our first prototype is developed using *Ethereum* framework[3]. The other prototype is developed using *HyperLedger* framework[4]. This section describes the implementation detail and the validation experiment of our prototypes.

5.1 BlockVOTE: Ethereum-based Implementation

Ethereum is the name of one of the most famous cryptocurrency ecosystem. Unlike Bitcoin, Ethereum supports a smart-contract capability in such a way that, any Ethereum wallet owner can deploy their own smart contract onto the Ethereum network globally to support their own business logic. To deploy BlockVOTE in Ethereum, we have to follow the smart contract development framework of the Ethereum foundation so-called the *Truffle Framework*[5]. The architecture of our Etherum-based BlockVOTE implementation is given in Figure 5.

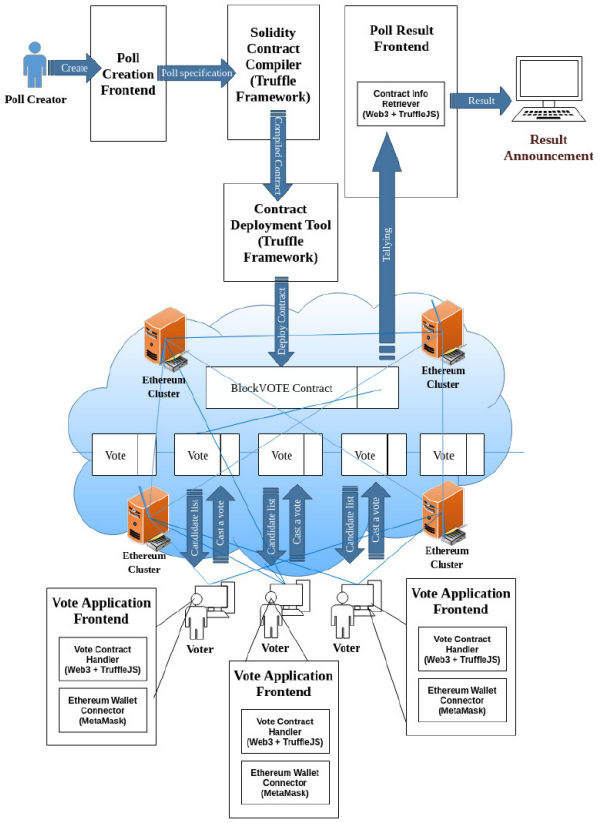

Fig.5: *BlockVOTE: Ethereum-based Implementation*

In our prototype, as depicted in Figure 5, a poll creator creates a poll using a simple web-based application frontend. This frontend application takes the list of candidates given by users to generate a smart contract sourcecode that can be later used by Ethereum. Ethereum specifies that a smart contract must be written in *Solidity* language. The template of our *BlockVOTE* contract in Solidity language is given as follows.

```
pragma solidity ^0.5.0;
contract BlockVote {
    struct Candidate {
        uint id;
        string desc;
        uint score;
    }
    mapping(address => bool) public voters;
    mapping(uint => Candidate)  public candidates;
    uint public candidatesCount;
    uint256 public exp_date;

    function addCandidate (string memory _name) private {
        candidatesCount ++;
        candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
    }
    function vote (uint _candidateId) public  {
        require(!voters[msg.sender]);
        require(_candidateId > 0 && _candidateId <= candidatesCount);
        require (now > exp_date);
        voters[msg.sender] = true;
        candidates[_candidateId].score++;
    }
}
```

[3] https://www.ethereum.org/
[4] https://www.hyperledger.org/
[5] https://truffleframework.com

After the contract code is generated, it is compiled using the Solidity compiler tool provided by the Truffle Framework. The compiled contract is later deployed onto the Ethereum network using the contract migration tool which is also provided by the Truffle Framework.

For each voter, in our prototype, they must have their own voting application in their machine. The voting machine can be a PC, a smartphone or a tablet, which is capable of accessing the Internet. The application utilizes Web3[6] and TruffleJS library for interacting with the deployed BlockVOTE contract.

Despite the fact that voting must be free of charge, according to Ethereum platform specification, a user must spend some amount of *Ether* (the name of the currency in Ethereum ecosystem) in order to create a new block. The amount of Ether that a user needs to spend depends on the size of data within the created block. Therefore, in our Ethereum-based voting application, each user needs to connect their own Ethereum wallet to the voting application before they can cast a vote. The cost per vote in BlockVOTE application, according to the automatic cost calculation in Ethereum network, is 0.000652 ETH per vote.

For the result tallying process, we create a separate frontend for such a purpose. The library used for tallying the result is also Web3 and TruffleJS. The tallying process is done by listing all candidates and their scores as noted within the contract to the frontend. Our frontend rejects the tallying request if the poll has not ended yet according to the expiry date of the poll given in the contract.

5.2 BlockVOTE: HyperLedger-based Implementation

HyperLedger is an opensource blockchain platform hosted by Linux foundation. Unlike Ethereum, the designed of HyperLedger is not designated for creating a public cryptocurrency-based ecosystem. Instead, it is designed for developing a private blockchain ecosystem where a private organization can create its own blockchain network internally, and deploy their business logic onto their network using a smart contract-based mechanism. The architecture overview of the HyperLedger-based BlockVOTE implementation that we develop in this study is given in Figure 6.

The architecture of BlockVOTE in Ethereum and Hyperledger implementations are mostly similar. The difference between both implementations are the languages for modeling a contract and a set of tools required. In HyperLedger, data models and built-in functions are needed to be programmed separately. Data models are written using a language specific to HyperLedger named *CTO* language, while built-in functions are written using Javascript. The template

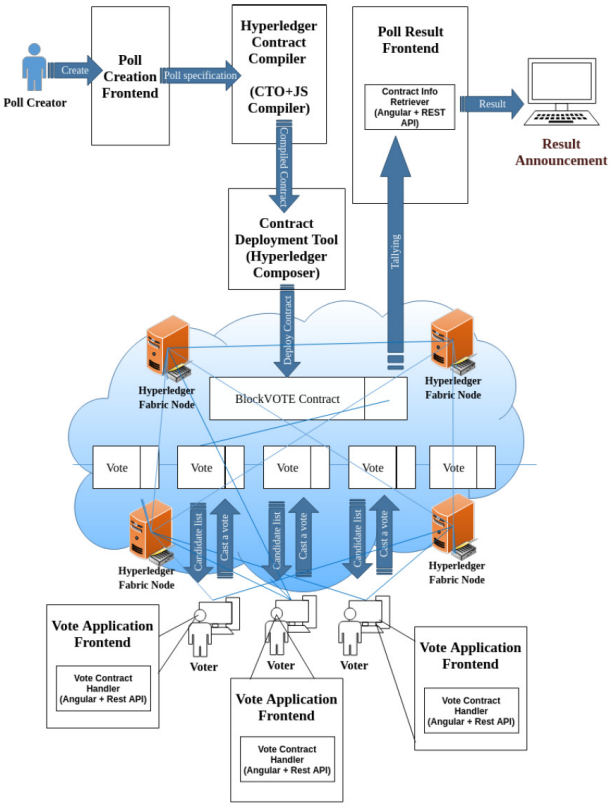

Fig. 6: *BlockVOTE: HyperLedger-based Implementation*

that our architecture uses for generating a data model and built-in function for creating a new BlockVOTE contract is as follows.

```
//Data Model Template//
namespace org.acme.blockvote

participant voter identified by voterID {
  o String voterID
  o String fullName
}

asset ifVoted identified by voterID {
  o String voterID
  o Boolean isvote
}

asset candidateVote identified by candidateID {
  o String candidateID
  o String short_desc
  o Integer totalVote
}

transaction vote {
  --> candidateVote candidateVoteAsset
  --> ifVoted ifVotedAsset
}

//Script Template//
'use strict';

function vote(tx) {
    if (!tx.ifVotedAsset.isvote) {
        tx.candidateVoteAsset.totalVote = tx.candidateVoteAsset.totalVote + 1;
        return getAssetRegistry('org.acme.blockvote.candidateVote')
            .then(function (assetRegistry) {
                return assetRegistry.update(tx.candidateVoteAsset)
                  .then(function(){
                      return getAssetRegistry('org.acme.blockvote.candidateVote')
                  })
            })
            .then(function () {
                return getAssetRegistry('org.acme.blockvote.ifVoted')
                    .then(function (assetRegistry) {
                        tx.ifVotedAsset.isvote = true;
                        return assetRegistry.update(tx.ifVotedAsset);
                    })
            });
    } else {
        throw new Error('Vote already submitted!');
    }
}
```

https://web3js.readthedocs.io/en/1.0/

After the contract is generated, it is compiled and deployed using *HyperLedger Composer* toolsets. The compiled BlockVOTE contract is deployed in a network of a pre-deployed *HyperLedger Fabric*[7] nodes. It is to be noted that this step is obligatory when using HyperLedger because HyperLedger is a private blockchain-based platform. Hence, a set of nodes is required for creating the infrastructure. This step is not necessary for the Ethereum-based prototype because Ethereum already has an extensive network of Ethereum node clusters available.

To cast a vote, as in our Ethereum-based prototype, voters need to have a frontend application to cast a vote. The frontend application is developed using Angular[8] and a set of web-based REST API provided by HyperLedger Fabric for connecting to the nearest HyperLedger Fabric node. A major different between HyperLedger and Ethereum based prototype is the fact the voting frontend application does not require a cryptocurrency wallet because HyperLedger assumes that all the related infrastructure is owned by the application owner themselves. Hence, voters do not need to pay any fee to cast a vote in HyperLedger-based implementation of our architecture.

5.3 Validation Experiment

To validate our proposal, we conduct an actual poll using both versions of our prototypes. A mock-up poll with 10 candidate choices is created and deployed in both of our Ethereum-based and HyperLedger-based prototypes.

We consider the performance of the tallying process to be crucial for a voting application. Hence, we test our prototype by conducting three sets of experiments with different number of votes in each set of experiments. The number of votes used is 10, 50, and 100 votes. The time delay between the time that the result tallying function is requested and when the result is actually available on the result notification front end is measured. Five rounds of experiments are conducted. The average delay of all five rounds of experiments is recorded. The result is given in Figure 7.

Figure 7 shows that the tallying delaying for all cases is almost constant for each respective implementation. The reason comes from the fact that every block in the network always contains the latest status of the contract. Hence, the tallying process does not require accessing the whole data starting from the beginning to the end of the chain. Only the latest block is required for the result tallying. Additional experiments were planned with larger number of votes to stress test the system. However the preliminary results suggests that the delay in tallying would not be significantly affected by increasing the number of votes. Due to that, additional testing with higher number of votes were deemed not necessary.

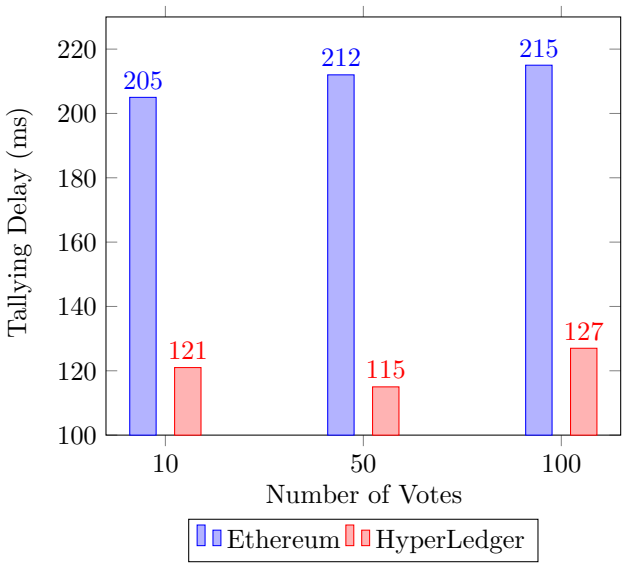

Fig.7: *Tallying Delay of Ethereum-based and HyperLedger-based BlockVOTE implementation*

It can also be observed from the result that the tallying process of HyperLedger is slightly faster than Ethereum. This comes from the different of the internal consensus handling algorithm between Ethereum and HyperLedger. Ethereum adopts a Proof-of-Stake based consensus which requires a user to pay some amount of Ether to warrant a transaction [23]. HyperLedger uses a certificate-based consensus handling algorithm through the Proof-of-Authority consensus [24] using Apache Kafka library[9]. Proof-of-Authority consensus works faster than Proof-of-Stake consensus so, it leads to the faster tallying process delay when implementing BlockVOTE on HyperLedger. Nevertheless, Proof-of-Stake has still recognized among blockchain researchers and developers that it is a more secure consensus handling mechanism [25] than the Proof-of-Authority.

6. DISCUSSION

To deploy our proposal in an actual election, technical and management challenges are needed to be carefully addressed and discussed. This section begins with the technical discussion of our BlockVOTE system. Next, the discussion on management issues concerning the possibility and feasibility of adopting BlockVOTE in practice. This section ends with the comparison of our proposal to an existing electronic voting system, and the traditional paper ballot-based voting system.

HyperLedger Fabric is a software that is used for creating a blockchain server node for HyperLedger

https://angular.io/

https://kafka.apache.org/

6.1 Technical Aspect

Major technical challenges of an electronic voting system, as mentioned earlier are related to securing network infrastructure and voting data. The major vulnerability of an existing electronic voting system comes from the fact that their architecture is mostly centralized. A centralized-based architecture makes it possible for cyber-attackers to target the main central server to tamper with the result.

This study proposes the decentralized electronic-voting system architecture using blockchain. By doing so, voting data is now stored among the decentralized blockchain nodes. Tampering with voting results in one of the node is not likely to be possible to do so because the tampered result will not be accepted by other nodes through the consensus handling mechanism of Blockchain. Hence, the challenges in securing both infrastructure and voting data can be overcome.

The proposal system was implemented using Ethereum and HyperLedger in this study. Both of the Ethereum and HyperLedger-based prototypes can be used for conducting a poll and tallying for the result in our validation experiment. One major issue to be considered in both of the prototypes are the fact that the Ethereum-based prototype requires voters to own an Ethereum coin wallet before they can cast a vote, while there is no such a requirement in our HyperLedger-based prototype. This comes from the fact that deploying BlockVOTE on Ethereum means that all the voting data has to be stored in the public Ethereum network. Hence, all voters need to follow the *Proof-of-Stake* consensus protocol of Ethereum by paying some amount of Ether before a vote can be cast. This may consider being a disadvantage on adopting Ethereum over HyperLedger because it is not feasible to require all voters to own an Ethereum wallet.

On the other hand, deploying BlockVOTE using HyperLedger does not require voters to own any kind of cryptocurrency wallet. However, a major disadvantage of adopting HyperLedger is the fact that there is a requirement to deploy several HyperLedger Fabric nodes before the development of applications on the network. This is required as there is no free public extensive network of HyperLedger in a similar manner with Ethereum. Developers who seek to implement an electronic voting application on blockchain need to weight these mentioned limitations of Ethereum and HyperLedger before starting any implementation.

6.2 Management Aspect

Regarding the management aspect, there is a need to be ensured that the result of an electronic voting system needs to be trusted by all related parties. Even though adopting blockchain mechanisms help guarantee that all voting data is secured, trustable and non-violable, there are still many management factors, especially human factors to be addressed before the system can be deemed trustable. The issues are summarized as follows.

- *Educating voters:* Adopting a blockchain-based electronic voting system requires voters to have some basic knowledge of technologies. Hence, voters whose familiar only with a ballot-based voting system must be well educated before a blockchain-based voting system can be used practically. A trial mock-up poll might need to be held first such that voters can have some time to learn and adopt to the new voting system;
- *Social Engineering/Human Error:* The proposed system can ensure that all the voting data can be kept non-volatile after votes are cast. However, the proposed system, along with many other voting systems cannot prevent errors that might be occurred by mistake from voters (e.g., voting the wrong candidate). The system also cannot prevent a social engineering case, where voters can be tricked.
- *Laws:* In government-organized elections, such as national elections, regional elections, or referendum polls, using an electronic system are usually not approved legally. There are many legal considerations that have been be fulfilled, before wider adoption could be examined. In the example of a blockchain-based approach, there are additional challenges due to the fact that the voting authorities may not approve of such approaches due to the decentralized nature of the approach. Laws and regulations of a given government might need to be reconsidered before a blockchain-based electronic voting system can be used.

6.3 Comparing the Voting Approaches

This subsection compares between the ballot based system, electronic voting systems, online voting systems, and the proposed blockchain systems between many different aspects. The summary of the discussions is displayed in Table 1.

6.3.1 Accountability and Fraud Prevention

Accountability and fraud prevention are considered as important aspects of the voting system. Voting is expected to be fair and reflect the mandate of the voting population. However, it is not possible to remove all instances of voting fraud, as some of the fraud are not the fault of the system but are beyond the scope of the voting system such as vote-buying instances. However, the discussion in this section explores the mechanisms of the voting system, and how they can potentially prevent instances of fraud and support accountability.

Ballot-Based systems are manual in nature. Due to that, there is potential for fraud. However, to protect against fraud, stakeholders can monitor the election and double-check the vote tallying to improve accountability and prevent fraud in the election. However, this requires manning of vote tallying stations

Table 1: *Comparison between Different Voting Systems and Important Factors*

Factors	**Type of Voting System**			
	Traditional Electronic Voting Systems			**BlockVOTE**
	Ballot Based Systems	**Electronic Voting Machines**	**Online Voting Systems**	**Proposed Block Chain Approach**
Accountability Fraud Prevention	Average	Low-Average	Low	Good
Validity of Ballot	Low	Good	Good	Good
Vote Tallying Time Factor	Extremely Slow	Extremely Slow to Average (Depending on Architecture)	Fast	Fast
Cost Factor	Expensive	Extremely Expensive (Less on the Long Run)	Expensive (Less on the Long Run)	Average (Less on the Long Run)
Accessibility	Low	Low	Low-Average	Average
Ease for Voter Turnout	Low	Low	Low to High	High
Implementation Difficulty	Low	Average	Extremely Hard	Hard
Training Required	Low	Average	High	High
Scalability	Low	Low	Average	High
Adoption Rates	High	Average	Low	Negligible

and monitors in all processes of the voting. This can be expensive and time-consuming.

Electronic voting machines range in accountability and fraud prevention. In examples of direct voting in the machines [26], it is possible to easily modify the tabular records to skew the election results directly. However, some machines only print out ballots, which is like the ballot-based systems. Some other systems provide a direct voting system and prints out the ballot in which would be used to verify the results of the voting, though similar to ballot-based systems.

Online voting systems are considered the least secure system due to the centralized nature of the voting server, and the potential to commit fraud anywhere along the chain of the system. Fraud that is committed is also difficult to detect. From a security standpoint, online systems have high risk.

The proposed blockchain decentralized system (i.e., our BlockVOTE) would be considered the most accountable system compared to the previous approaches. The system itself has mechanisms for fraud detection built within the blockchain. By doing so, voting data is now stored among the decentralized blockchain nodes. Tampering with voting results in one of the nodes is considered extremely difficult because the tampered result will not be accepted by other nodes through the consensus handling mechanism of blockchain. It is possible to have a self-auditing system that could not be easily modified due to the nature of the distributed blockchain.

6.3.2 Validity of Ballot

One factor that affects voting is the validity of the ballot, whether the voter has submitted a valid ballot that follows the voting regulations and could be tallied. In the example of ballot-based systems, it is noticed that there is a minority of ballots that have been discarded due to issues of being marked incorrectly. In some cases, the mismarks can be intentional, but the majority are usually considered to be unintentional [27]. Traditional systems depend upon the voters to cast ballots that are valid.

For other systems, due to the electronic nature of the system, it considered by default that all the ballots that are submitted by the voter to the system to be considered valid.

6.3.3 Voting Tallying Time Factor

One of the issues of voting systems is that it takes significant time for the voting tallying to finish. In traditional ballot-based systems, the amount of time required is extensive as the tally is done manually. Electronic voting machines vary in the time requirement. For systems that allow direct voting into the machine, it is possible to quickly tally the votes, though it is considered dangerous due to the lack of accountability. Other systems that provide a paper ballot printout, the results are similar to the ballot-based systems due to the need to manually tally the results. Online voting systems are considered the fastest in tallying the voting results due to the centralized nature of the system. The proposed blockchain approach could also accomplish the tallying process in comparability quick time with centralized online voting systems, though just marginally slower due to the multi-authority server authentication requirements.

6.3.4 Cost Factor

The ballot-based system is expensive to conduct due to the intensive manual undertaking of the voting. This is also coupled with the requirements of preparing the ballots before-hand and the logistics required for the voting, in which little could be reused in the next voting. For electronic voting systems, the initial investment is considered extremely expensive. This is due to the high cost of electronic voting machines that needed to be invested upon. However, the electronic voting systems can potentially be reused, saving the cost required for the paper printouts of the ballots on subsequent elections. Online voting systems may seem cheap to implement, but due to the security issues that plague centralized online systems, it is important to invest significantly in security and testing before deployment. This makes the cost of robust elections systems considered expensive. However, this can be offset that future uses of the system would require less investment and could be beneficial in the long run. The proposed blockchain system, in theory, is likely to cost the least. Due to the nature of the blockchain technology, the security issues and the distributed and decentralized nature of the blockchain solves many of the outstanding issues that must be addressed on online voting systems. Due to that, the cost of development of a national level voting should potentially cost less than online voting systems. In addition, there is potential for reuse, making the system cheaper to utilize in the long run, making the approach very attractive on the cost factor.

6.3.5 Accessibility

Accessibility to voting is considered another important factor. When considering accessibility, it is possible to examine the accessibility of the person to the voting station, or the ability to provide accessibility options to people who have disabilities to allow them to vote. In examples of ballot-based systems and electronic voting machines, there is little flexibility in both factors. The voting has to be conducted on-site, and there are little options to help people with disabilities in voting on the system. Online voting systems may provide more options in the accessibility aspect. Due to the online nature, it is possible to allow voters to vote online as opposed to predefined voting stations. In addition, the client application used is likely to be more customizable providing greater accessibility options. This is not always guaranteed, as some online voting systems may be based on EVM that require users to cast their vote in voting stations. The decentralized blockchain approach may provide the best accessibility options. Due to the distributed nature, it is not required that votes must be cast on the voting stations. Also, the client could be developed for accessibility options providing the best performance.

6.3.6 Ease for Voter Turnout

Voter turnout is considered an important issue in voting. It has been reported in the previous literature that there is a myriad of factors that can affect voter turnout [28]. One such factor that has been reported is that many people who did not go to vote stated that the voting station was far and was the primary reason against going to vote. Proximity to the voting stations would then be considered one important factor in improving potential voter turnout. Increasing voting stations is considered an expensive proposition, so the issue is not covered. However, some of the voting systems have the potential to allow remote voting, which eradicates the proximity to the voting station from the equation. Online voting systems and the proposed blockchain approach can offer an advantage.

6.3.7 Implementation Difficulty

Traditional ballot-based systems are considered the easiest to organize due to the simplicity in the sys-tem and the years of experience in organizing voting. Electronic voting systems require the implementation of electronic voting machines which require additional implementation overhead. However, the requirement for electronic voting machines are generally less when compared to online voting systems. Online systems need to be secure, so that the risk of vote tampering or hacking during the voting would not occur, and should be resilient that the system could handle large numbers of concurrent users in adequate time. Those requirements are considered very difficult to implement properly and there had been many cases of online systems failing under high loads especially in larger election scenarios. In the example of blockchain systems, the implementation of the system is considered to be quite difficult. Luckily the underlining mechanisms of the blockchain systems could help mitigate many of the technical challenges present.

6.3.8 Training Required

Ballot based systems are common, and usually, the general population can cast their ballots without much difficulty. However, when moving to electronic-based approaches, there are some segments of the voting population that may not be well-versed with technology especially in the population with little IT literacy. There are some requirements for training for usage with Electronic Voting Machines, but due to the limited input possibilities, the training is considered trivial. However, for online voting or blockchain-based approaches, there are requirements to educate the population on how the system works extensively to avoid such problems during the lead up to the voting. In addition to that factor, there has been a little study on human-computer interaction on blockchain applications [29], which makes it harder to developer

intuitive applications for people to use.

6.3.9 Scalability

Scalability is another issue that needs to be addressed. For ballot based systems, the scalability is low. Once the number of voters have been decided, the number of ballots would have to be prepared according to the number of voters. Any changes would usually require the reprinting of ballots, in which the process can take weeks to months to prepare for larger elections. For electronic voting machines, the scalability is also quite low. After the EVM have been procured, additional procurement of machines would not be easy, and requires significant investment, which factors heavily against the scalability. However the EVM approach does not require reprints of the ballots, and could scale better on that regard.

For centralized online voting systems, the approach scales better with lower number of voters. However when expanding to higher number voters that would use the system in short window of time, multiple issues will have to be addressed. To handle larger loads, a centralized system could not be a monolithic architecture. Increasing the processing and I/O of the server is not adequate for those purposes, and multiple approaches could be attempted. Load-balancing systems could have to be applied with the centralized system to help account for server loads. However in elections, load-balancing would not be adequate and other data approaches would be utilized. Dedicated data service could be applied which uses concepts of range, hash, or splitting to improve the centralized system performance. Additional optimization in lower level approaches in read and write level could be done to improve performance. The issues would cascade causing scalability to be exponentially more difficult with higher loads.

The scalability issues are less of an issue with decentralized blockchain system. With decentralized systems, scaling is less of an issue as the system was initially designed for scalability. Expanding the scale of the system would not be a huge undertaking in decentralized blockchain system, due to the system design. However the issue with the increased scale does not come from handling the users directly, but from the issue of consensus which could be time-consuming for larger scale of operations. One issue that can help alleviate the issue is that the proposed system does not utilize the Proof of Work, consensus mechanism. The Proof of Work consensus mechanism is considered to be slow, and the Proof of Authority consensus mechanism was adopted due to the better fit in the framework of voting, and due to the faster consensus handling performance, which would offer overall better scalability options.

6.3.10 Adoption Rates

Ballot based systems are the most common approaches in voting and are widely adopted in many elections. Due to the slow speed in tallying and inaccuracies in ballot based systems, electronic voting systems have been utilized. There had been a number of national-level elections such in United States of America [6], India [10], and many other countries that have utilized electronic voting systems with varying degrees of success. Due to the security challenges of online voting systems, online voting has not been widely adopted and mostly has been used on a trial basis. Experiments include trail voting in USA [9] to test the feasibility and wider adoption has been reported in Switzerland where online voting has been used on trial basis on canton level voting and the program has gradually been expanded [11]. For blockchain-based voting systems that are decentralized, there has been negligible adoption. Due to the decentralized nature of the system, many voting authorities have not approved the approach due to many reasons ranging from legal consideration, technical issues, and management issues. One of the largest adoption of the decentralized blockchain-based voting system was recorded at the Moscow City Duma election of 2019[12] where a total of 3 districts with approximately 450,000 voters and 6% of the city residents were voting with the blockchain system. Though the voting results were considered a success, external security experts have discovered weakness in the implemented encryption scheme that was too weak, and could be cracked. Though the blockchain system was cracked, this was not due to the weakness in the blockchain system itself, but from the weak encryption scheme implemented.

7. CONCLUSION AND FUTURE WORK

Though traditional centralized voting systems have been used since the invention of democracy, there is always some potential to improve the voting systems. In the area of centralized voting systems, there had been the exploration of usage of online voting systems which offer a compelling improvement from ballot systems and electronic voting systems. However online voting systems are complex and there is potential for abuse in the system, without mechanisms to double-check the results, that would have severe ramifications to the results of the elections.

An alternative approach in the voting system is to explore other approaches. Blockchain technology which is an open, distributed and self-auditing ledger

EVM in India:
https://eci.gov.in/faqs/evm/general-qa/electronic-voting-machine-r2/
Online Voting Details at Switzerland
https://www.ch.ch/en/demokratie/voting-online
Moscow City Duma Election 2019 - Blockchain Experiment
https://www.mos.ru/en/city/projects/blockchain-vybory/

that can record transactions between parties in an efficient, permanent, and verifiable manner is one technology that could be adopted for use in voting systems. As opposed to centralized voting systems, blockchain could provide a decentralized system that is robust and fulfill many of the needs required in voting systems. The research has proposed *BlockVOTE: An Architecture of a Blockchain-based Electronic Voting System*. The system was implemented to test the feasibility of implementing a decentralized voting system. The architecture, data models and all related formal definitions are proposed in this study. The architecture was validated by means of prototyping and experimenting using two different implementation frameworks which are Ethereum and HyperLedger. The result shows that the proposed system, in both of our implementations, can be used for conducting a poll, keeping the result secured, while keeping the result tallying time to be minimized.

Though the proposed voting system may provide many advantages, there are numerous issues that need to be addressed before the adoption of the technology could be used in voting that could have national level ramifications. The issue of voting authority, voter registry, and trust are huge issues. For national-level elections, voting authorities hold huge power over the organization of voting, which is consistent with the centralized nature of the voting systems. However, when changing to decentralized systems, there is less requirement of a central powerful authority in voting systems that could cause conflict due to limitation of power. To allow blockchain decentralized elections to work, the voter registry must be provided to the blockchain. This is likely to face resistance as voter registry is a sensitive national data. Another issue is the issue of trust. Though blockchain, in theory, provides self-auditing, which makes the results fairer, the demotion of the central voting authority may cause issues with trust, as no central voting authority would be responsible for the validation of the voting results. In a decentralized blockchain, it would be changed to several authority figures based on the design. As there is no central voting authority, the declaration of results can be difficult to comprehend as the results depend on the self-auditing process in the blockchain that has been verified by selected authentication figures.

However, if the issues addressed earlier have been resolved, decentralized voting systems utilizing blockchain technology can provide many advantages over old centralized voting systems. Decentralized blockchain, in theory, allows greater accountability, accuracy and fraud prevention measures compared to existing centralized voting systems and at a cheaper cost. In the ballot casting process, there is potential to improve the accessibility and voter turnout with the framework. In addition to that, the voting tallying process can be improved, and voting results could be validated soon after elections.

For future work, there are several topics that the research team is tackling. One of the areas is to examine another blockchain-based application framework, or proposing our own framework that is more suitable to be used in an electronic voting system than Ethereum or HyperLedger. Another area is to expand the test voting scenario, and a larger voting scenario could be explored and examined in further details. Another area would be exploring the security of the system, by attempting systematic attacks on the system to examine the resilience of the system. Another future work that would be useful is to propagate the usefulness of blockchain technology in elections to a wider audience so that the voting authorities and governmental organizations would examine the technology and may adopt it for future voting.

ACKNOWLEDGEMENTS

This research is supported by the Faculty of Science Research Fund (2019), Faculty of Science, Prince of Songkla University. The researchers would like to acknowledge and thank all the support provided by Prince of Songkla University during the period of this research.

References

[1] R. Dugger, "Annals of democracy: Counting votes," *Annals of democracy: Counting votes*, vol. 64, no. 38, p. 40108, 1988.

[2] S. Nakamoto, "Bitcoin: A peerto-peer electronic cash system, `http://bitcoin.org/bitcoin.pdf`," 2009.

[3] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trustless medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 1475714767, 2017.

[4] R. Casado-Vara, A. González-Briones, J. Prieto, and J. M. Corchado, "Smart contract for monitoring and control of logistics activities: Pharmaceutical utilities case study, in *International Joint Conference SOCO'18-CISIS'18ICEUTE'18*, M. Grana, J. M. López-Guede, O. Etxaniz, Á. Herrero, J. A. Sáez, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2019, pp. 509517.

[5] T. Kohno, A. Stubbleeld, A. D. Rubin, and D. S. Wallach, "Analysis of an electronic voting system," in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, May 2004, pp. 2740.

[6] A. Hern, "Kids at hacking conference show how easily us elections could be sabotaged hacking," *The Guardian*, Aug 2018.

[7] S. Robinson, "Did your vote count? new coded

ballots may prove it did," *New York Times*, Mar 2004.

[8] A. D. Smith and J. S. Clark, "Revolutionising the voting process through online strategies," *Online Information Review*, vol. 29, no. 5, pp. 513530, 2005.

[9] B. Simons and D. W. Jones, "Internet voting in the us," *Communications of the ACM*, vol. 55, no. 10, pp. 6877, 2012.

[10] A. Gervais, G. O. Karame, K. Wust, 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*, ser. CCS'16. New York, NY, USA: ACM, 2016, pp. 316. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978341

[11] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2016, pp. 467468.

[12] A. Roehrs, C. A. da Costa, and R. da Rosa Righi, "Omniphr: A distributed architecture model to integrate personal health records," *Journal of Biomedical Informatics*, vol. 71, pp. 70 81, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1532046417301089

[13] D. Tse, B. Zhang, Y. Yang, C. Cheng, and H. Mu, "Blockchain application in food supply information security," in *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Dec 2017, pp. 13571361.

[14] C. Angsuchotmetee and P. Kaewkandee, "Vodcoin: A cryptocurrency-based architecture for a decentralized-based video-on-demand service," in *Proceedings of the 10th International Conference on Management of Digital EcoSystems*, ser. MEDES '18. New York, NY, USA: ACM, 2018, pp. 100105. [Online]. Available: http://doi.acm.org/10.1145/3281375.3281392

[15] J.-H. Huh and K. Seo, "Blockchain-based mobile ngerprint verication and automatic log-in platform for future computing," *The Journal of Supercomputing*, vol. 75, no. 6, pp. 31233139, Jun 2019. [Online]. Available: https://doi.org/10.1007/s11227-018-2496-1

[16] P. Noizat, "Chapter 22 - blockchain electronic vote," in *Handbook of Digital Currency*, D. L. K. Chuen], Ed. San Diego: Academic Press, 2015, pp. 453 461. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128021170000229

[17] N. Kshetri and J. Voas, "Blockchain-enabled evoting," *IEEE Software*, vol. 35, no. 4, pp. 9599, 2018.

[18] F. . Hjlmarsson, G. K. Hreiarsson, M. Hamdaqa, and G. Hjlmtsson, "Blockchain-based e-voting system," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2018, pp. 983986.

[19] E. Yavuz, A. K. Ko, U. C. abuk, and G. Dalkl, "Towards secure e-voting using ethereum blockchain," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, 2018, pp. 17.

[20] S. Shukla, A. N. Thasmiya, D. O. Shashank, and H. R. Mamatha, "Online voting application using ethereum blockchain," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 873880.

[21] F. Sheer Hardwick, A. Gioulis, R. Naeem Akram, and K. Markantonakis, "E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy," in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018, pp.1561 1567.

[22] D. Khoury, E. F. Kfoury, A. Kassem, and H. Harb, "Decentralized voting platform based on ethereum blockchain," in *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, 2018, pp.16.

[23] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, J. GarciaAlfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomartí, Eds. Cham: Springer International Publishing, 2017, pp. 297315.

[24] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: ACM, 2018, pp. 30:130:15. [Online]. Available: http://doi.acm.org/10.1145/3190508.3190538

[25] L. Ismail, H. Hameed, M. AlShamsi, M. AlHammadi, and N. AlDhanhani, "Towards a blockchain deployment at uae university: Performance evaluation and blockchain taxonomy," in *Proceedings of the 2019 International Conference*

on Blockchain Technology, ser. ICBCT 2019. New York, NY, USA: ACM, 2019, pp. 3038. [Online]. Available: http://doi.acm.org/10.1145/3320154.3320156

[26] J. Bannet, D. W. Price, A. Rudys, J. Singer, and D. S. Wallach, "Hack-a-vote: Security issues with electronic voting systems," *IEEE Security & Privacy*, vol. 2, no. 1, pp. 3237, 2004.

[27] F. Uggla, "Incompetence, alienation, or calculation? explaining levels of invalid ballots and extra-parliamentary votes," *Comparative Political Studies*, vol. 41, no. 8, pp. 11411164, 2008.

[28] B. Geys, "Explaining voter turnout: A review of aggregate-level research," *Electoral studies*, vol. 25, no. 4, pp. 637663, 2006.

[29] C. Elsden, A. Manohar, J. Briggs, M. Harding, C. Speed, and J. Vines, "Making sense of blockchain applications: A typology for hci," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 458.

Chinnapong Angsuchotmetee is currently the Deputy Head of the Department of Computer Science, Faculty of Science, Prince of Songkla University. He has received his Bachelor degree in Computer Science, and Master degree in Information Technology from King Mongkut's University of Technology Thonburi. He has received his PhD degree in Computer Science from Universite de Pau et des Pays de l'Adour, France. His research interests include IP Telecommunication, Next Generation Networking, Ubiquitous Computing, Internet of Things, and blockchain technology.

Pisal Setthawong is currently the Deputy Chairperson of the Department of Management Information Systems at Assumption University. He has received his Bachelor and Master degree in Computer Science from Assumption University and his PhD degree in Computer Science from King Mongkut's University of Technology Thonburi. His research interests includes image processing, computer graphics, multimedia technologies, IoT applications, and blockchain technology.