Block Chain Voting System

SAGAR SHAH

QAISH KANCHWALA

HUAIQIAN MI

FROM NORTHEASTERN UNIVERSITY

Table of Contents

Introduction Proposed System
Pronosed System.
Architecture
Parts of the system
User
Authentication Server(AS)
Arbitration Server(AR)
Blockchain system
Processes in the system
Registering to vote
Voting
Verifying the vote
Counting votes
Recounting the cotes
Economic advantages of the system
Cost of current voting system
Cost of proposed system
Cost benefit analysis
Social benefits1
Conclusion1

Executive Summary

Large sections of society today do not trust their government [1]. This makes the election a very important event in a modern democracy. The issue with the current ballot system is that it can be easily manipulated by power hungry organizations [2]. The proposed system looks to eliminate the aspect of trust from an election to make it more secure and transparent. The system uses existing technology such as a client server architecture integrated with a blockchain system to ensure aspects such as transparency, security and auditability are achieved without sacrificing privacy for voters. The cost of building the system is substantially less as compared to the cost of running a ballot based system. There are substantial social benefits to using the system as well such an easier and quicker voting process which will lead to higher voter turnout. This system can be implemented for a larger number of countries as the internet penetration in the world increases. We might definitely see a future where every country has implemented a system similar to ours.

Introduction

In today's world, widespread mistrust towards the government and interference in countries' processes by external actors have made the democratic process of voting more critical than ever. Democratic countries have been experiencing dictatorial regimes which have introduced widespread terror among their people. People have had their human rights violated and their fundamental freedoms provided by their constitution taken away. In such an atmosphere, having a fair and transparent election is something that is paramount for the freedom most people enjoy today.

The pitfalls of the current system of ballot voting are being taken advantage of by people or organizations looking to gain power. In the African countries of Uganda and Kenya there has been widespread controversy over their elections in recent years. The election of 1946 in Romania was heavily rigged. The communists took over Romania and abolished the multi-party system to gain complete control of the country [2].

These instances of controversial elections could all have been avoided if the counting process was fair, transparent and verifiable. The current ballot system does offer anonymity to the voter but the counting process is not transparent. People are supposed to trust the result which is provided by an Election commission or a government body. This makes the process of counting, a major vulnerability in the current process. There are also other major electoral scams such as voter fraud, ballot stuffing and booth capturing. All these make it very difficult for organizers of an election to distinguish between the actual votes and votes added without authorization.

The system that is being proposed solves most of the issues mentioned above and can be implemented in the current world environment.

Proposed System

The proposed system involves a client server architecture integrated with a block chain system. The minimum requirements needed by a voter is a smartphone or a computer with a webcam and an internet connection. If these are not met alternate arrangements such as pop up cyber cafes and computers at public buildings must provide access to disadvantaged voters.

Architecture

Shown below is the network architecture of the proposed system.

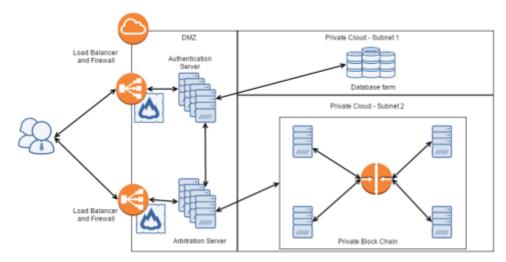


Figure 1: Network architecture of proposed system

The above diagram shows how the user interacts with the different parts of the system. The system will be explained in two parts — explaining the functionality of each part and the processes associated with the system.

Parts of the system

There are four main parts to the system. Each of them will be explained in detail below.

User

The user must have a smartphone, laptop or any device with a browser and a front facing camera. The user must also have an internet connection to register and vote as well. If the user does not have a computer or an internet connection, he/she could go to a public building such as a library or a school which does have computers to register to vote. These could be kept open all day during voting registration and voting days to ensure people with low sources of income do not get left out.

Authentication Server(AS)

The Authentication Server is a traditional centralized web server. It has a backend database connected to it which has the information of all the citizens in the country. This system is used by people to register to vote for their elections. People create login accounts when they register. It also creates accounts on the blockchain system for the users when they vote. The blockchain account is used by the Arbitration Server to vote for a candidate of the users' choice. The AS also authenticates the token provided to the Authorization Server by the user while voting.

Arbitration Server(AR)

The Arbitration Server acts as an intermediary between a user and the Blockchain voting system. It verifies the user while voting using the Authentication Server. The AR is a blockchain thin client that sends the users' vote to a blockchain node [3]. It also sends the user the key to encrypt their vote. The AR sends the users' vote to the appropriate node to be added to the blockchain. The user can verify their vote using the AR as an intermediary.

Blockchain system

The blockchain system is the system on which the actual voting takes place. The users' vote is sent to the one of the nodes on the system depending on the load on each node. The node then adds the transaction to the blockchain depending on the smart contracts that exist on each node. The smart contracts are the rules that the nodes follow to not only verify but also add the vote in the system. Each node follows the smart contract to verify the vote. The blockchain is a private system and is not accessible to the public directly. The system will currently have node server in each state to ensure a distributed network traffic on the system. The

Processes in the system

The process of voting has two steps in the old process, i.e. registering to vote and the process of voting itself. The proposed system will have an extra step, i.e. the ability for the user to verify their vote. This is an important step where the user can get a confirmation of their vote. The other steps include counting the votes by the organizer and recounting in case of any discrepancies.

Registering to vote

The process of registering to vote begins with the user interacting with the Authentication Server via a website. The AS contains information about voters in a database. The user enters his/her Personally Identifiable Information(PII) and scans supporting documentation to upload into the system along with an email address. The users' picture is also taken for verification. If the information is verified and is correct, the user is allowed to create an account. The user enters a username of their choice and a password to log in. This information is stored separately and not linked to the users' PII. This ensures **privacy** and **anonymity** while voting. Also an entry is made next to the users' database entry storing whether he/she has registered to vote. If the users' information cannot be verified, he/she is not allowed to create an account.

All the information between the user and the AS is sent using TLS v1.2 protocol to ensure it is all secure. The code for registration process is placed in Appendix B and Appendix C of this document.

Voting

The process of voting is a multi-step process. It involves verifying your identity with the AS and then voting using the AR. On the day of voting each candidate is given an account on the blockchain system so they can get votes. There is also a "Abstain account" for abstained votes to be sent to.

During voting day, the user logs in to the authentication server using the username and password created in the previous step. An image of the user is taken to ensure that the user is the owner of the account. This image is compared with the image taken during registration.

A small video of the user is taken before they log in and is sent to the Authentication Server. Using the Affectiva API the AS can identify user emotions based on machine learning technology [4]. If the system detects fear, the users' session is stopped and told to retry after 5 minutes. While logging in for the second time if the system detects fear again, the user must go to their local polling center such as a library to vote. This system would reduce the impact of the issue related to **voting under duress**.

Once the user logs in, their system would create a public key which they would send to the Authentication Server. The AS would add associate the key with the username. The key would be used to create an account for the user on the blockchain system to vote. A specific amount of ether (currency the user can use to vote) is added to the users' account which enables them to vote.

The AS would then send a session token back to the user.

The user would be redirected to the AR. The user would provide the AR with the session token, would verify it with AS. The verification and generation of token between the AR, user and AS is done using the Modified Needham-Schroeder protocol. This protects the system from impersonation and man in the middle attacks.

The AR would send a verification message to the user along with the public key of the blockchain node to which his/her vote would be sent.

The user would encrypt their vote with the public key and send it to AR. This would ensure that the AR cannot read the users' vote and hence the vote would remain a secret. The AR would send the encrypted vote to the appropriate node. The node would decrypt the message with their private key and send a specific amount of ether from the users' account to the candidates' (or to the Abstain account if they would like to forfeit their vote) blockchain account. Each node would verify the transaction according to the smart contracts. These contracts would verify a particular transaction was a duplicate one or no and check its validity. After this process the node would pass this transaction to other nodes in the blockchain system.

The code for the smart contracts used in the Blockchain system is placed in Appendix A.

Verifying the vote

The process of verifying the vote depends on the type of election it is. Some elections allow for interim results and some do not. In either case the voter must get a confirmation that his/her transaction has been approved by the blockchain system.

In case of the election that allows interim results, one of the nodes of the blockchain could be made publicly accessible. It would have a website similar to https://blockchain.info where a user could enter their public key to verify whether their vote was counted. This node would not have the ability to add any transactions to the system. This will be implemented through smart contracts. It will only be a reader of the transactions. This will reduce the attack surface of the system.

If the organizers of an election want to keep the interim results a secret, he/she could only get a binary verification via the AR. Since the AR is a thin client it would act as an intermediary to verify the transaction [3]. At the end of the election in above case, the user will be able to check the result for the election.

Counting votes

The process of counting votes of a candidate can be very simple. Each voter has a fixed amount of ether or currency value that they use to vote for a candidate of their choice. The candidate with the highest amount of ether in their account wins the election.

For users who abstained from voting, their ether will be sent to an Abstain Account. This ensures their vote does not get misused.

Recounting the cotes

There are instances of disputes in the results of an election. These can be resolved in the proposed system easily.

The entire tree associated with a single account root can be made public for people to verify if their vote has been tallied or no [5]. This makes the system transparent for users. Since no one knows which user is associated with which account, it protects the users' anonymity in voting. The public keys for each transaction in the blockchain system can be mapped to accounts in the AS. The list of public keys generated gives the list of people who voted. By mapping this list with the public keys associated with each transaction, the election can be verified.

Economic advantages of the system

Cost benefit analysis of proposed system

Cost of labor per person = \$80/hour

Requirement of 25 people over 12 months to build and test the system

One time Cost to customer = \$4,000,000

Cost of hardware and maintenance for 1^{st} election = \$100,000,000 for a voter base of 100 million (based on EC2 calculator)

(Includes data center costs, network equipment and bandwidth)

For subsequent election cycles cost = \$50,000,000

Cost of running a ballot based election = \$2 per person [6]

= \$200,000,000 for 100 million voters

The cost of running the election for the 1^{st} year using proposed system = \$104,000,000

Cost for a country

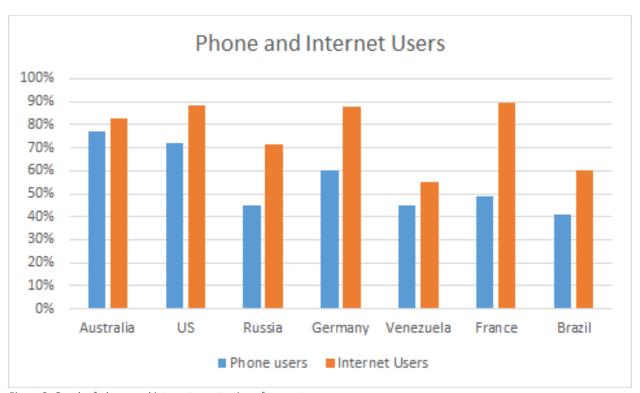


Figure 2: Graph of phone and internet penetration of a country

The graph above shows the internet and smartphone penetration in a country. These are the two minimum requirements of a user to vote. The internet penetration is given a higher importance since people can still go to government building such as libraries to vote. The

system requires internet penetration of a country to be greater than or equals to 90% to make this project feasible. Anything below that will require the government to invest into too much infrastructure to set up an internet connection for users. These are current statistics of countries in the world. Large number or people in the world are gaining access to the internet every day. As this number grows the proposed system will become feasible for many more countries in the world than it is now.

Social benefits

The system of remote blockchain voting will impact society in a very positive way.

The system will increase convenience for voters [7]. It will make it very easy for people with disabilities or who have trouble moving around to vote. It is very quick and private way to vote [8]. This will increase the number of voters since the process does not take up too much of their time of the day. It will help increase the trust of the people in the government since it is more transparent than the current ballot system. The system is better for the environment as compared to the paper voting system. It eliminates the need for paper voting and the carbon emitted by the logistics of those ballots. Hence this system has a much smaller carbon footprint.

Conclusion

The current ballot system is shown to have large number of issues which can lead to widespread political unrest in a country. It is vital for a democracy to have a transparent voting system that must have the least number of obstacles for a voter to vote. The proposed system not only handles voter privacy and auditability but also provides a transparent system for verification of the election. The proposed system is shown to be highly cost efficient as compared to other countries and can be implemented with existing infrastructure owned by a nation. Keeping all these factors in mind the proposed system is a comprehensive solution that satisfies all the requirements requested by the client.

Bibliography

- [1] Gallup, "Trust in Government," Gallup, 30 September 2015. [Online]. Available: http://www.gallup.com/poll/5392/trust-government.aspx. [Accessed 28 September 2016].
- [2] Wikipedia, "List of controversial elections," 20 September 2016. [Online]. Available: https://en.wikipedia.org/wiki/List_of_controversial_elections. [Accessed 27 September 2016].
- [3] R. Skudnov, "Bitcoin Clients," Turku University of Applied Sciences, Turku, 2012.
- [4] Affectiva, "Affective Product Overview," 15 January 2016. [Online]. Available: http://www.affectiva.com/wp-content/uploads/2014/11/AffectivaProductOverview.pdf. [Accessed 28 September 2016].
- [5] P. Noizat, "Blockchain Electronic Vote," in *handbook of digital Currency*, Paris, Elsevier Inc., 2015, pp. 453-461.
- [6] The electoral knowledge network, "Cost of Registration and Elections," ACE Project, 15 Jan 2016. [Online]. Available: http://aceproject.org/ace-en/focus/core/crb/crb03. [Accessed 28 September 2016].
- [7] N. Uribe, "10 Benefits of Electronic Voting," 01 August 2016. [Online]. Available: http://www.fobsoftware.com/blog/10-benefits-of-electronic-voting-for-home-owner-associations. [Accessed 28 September 2016].
- [8] G. Schryen, "Security Aspects of Internet Voting," in IEEE, Hawaii, 2004.
- [9] Wikipedia, "List of countries by number of Internet users," 23 September 2016. [Online]. Available: https://en.wikipedia.org/wiki/List_of_countries_by_number_of_Internet_users. [Accessed 29 September 2016].
- [10 Wikipedia, "List of countries by smartphone penetration," 18 September 2016. [Online].
- Available: https://en.wikipedia.org/wiki/List_of_countries_by_smartphone_penetration#2016_rankings. [Accessed 29 September 2016].

Appendix A: Smart Contracts for the blockchain system

```
contract NewPoll {
 //defines the poll
 struct Poll {
  address owner;
  string title;
  uint votelimit;
  string options;
  uint deadline;
  bool status;
  uint num Votes;
 // event tracking of all votes
 event NewVote(string votechoice);
 // declare a public poll called p
 Poll public p;
 //initiator function that stores the necessary poll information
 function NewPoll(string _options, string _title, uint _votelimit, uint _deadline) {
  p.owner = msg.sender;
  p.options = _options;
  p.title = _title;
  p.votelimit = _votelimit;
  p.deadline = _deadline;
  p.status = true;
  p.numVotes = 0;
 //function for user vote. input is a string choice
 function vote(string choice) returns (bool) {
  if (msg.sender!= p.owner || p.status!= true) {
   return false;
  p.numVotes += 1;
  NewVote(choice);
  // if votelimit reached, end poll
  if (p.votelimit > 0) {
   if (p.numVotes >= p.votelimit) {
     endPoll();
  return true;
```

```
//when time or vote limit is reached, set the poll status to false
function endPoll() returns (bool) {
  if (msg.sender!= p.owner) {
    return false;
  }
  p.status = false;
  return true;
}
```

APPENDIX B: Controller for Registration System in Authentication Server

```
def link(): return response.download(request,db,attachment=False)
def index():
       image form = FORM(
               INPUT(_name='image_title',_type='text'),
               INPUT(_name='image_file',_type='file')
       if image_form.accepts(reguest.vars,formname='image_form'):
               image = db.image.file.store(image_form.vars.image_file.file,
image_form.vars.image_file.filename)
               id = db.image.insert(file=image,title=image_form.vars.image_title)
       images = db().select(db.image.ALL)
       return dict(images=images)
def user():
  exposes:
  http://.../[app]/default/user/login
  http://..../[app]/default/user/logout
  http://..../[app]/default/user/register
  http://.../[app]/default/user/profile
  http://..../[app]/default/user/retrieve_password
  http://..../[app]/default/user/change_password
  http://..../[app]/default/user/bulk_register
  use @auth.requires_login()
     @auth.requires_membership('group name')
     @auth.requires_permission('read', 'table name', record_id)
  to decorate functions that need access control
  also notice there is http://..../[app]/appadmin/manage/auth to allow administrator to manage
users
  return dict(form=auth())
@cache.action()
def download():
  allows downloading of uploaded files
  http://..../[app]/default/download/[filename]
```

111111

return response.download(request, db)

```
def call():
```

exposes services. for example: http://..../[app]/default/call/jsonrpc decorate with @services.jsonrpc the functions to expose supports xml, json, xmlrpc, jsonrpc, amfrpc, rss, csv

return service()

APPENDIX C: Model for Registration System in Authentication Server

```
db = DAL("sqlite://storage.sqlite")
db.define_table('image',
          Field('title', unique=True),
          Field('file', 'upload'),
          format = '%(title)s')
db.define_table('post',
          Field('image_id', 'reference image'),
          Field('author'),
          Field('email'),
          Field('body', 'text'))
db.image.title.requires = IS_NOT_IN_DB(db, db.image.title)
db.post.image_id.requires = IS_IN_DB(db, db.image.id, '%(title)s')
db.post.author.requires = IS_NOT_EMPTY()
db.post.email.requires = IS_EMAIL()
db.post.body.requires = IS_NOT_EMPTY()
db.post.image_id.writable = db.post.image_id.readable = False
from gluon.contrib.appconfig import AppConfig
myconf = AppConfig(reload=True)
```

APPENDIX D: Table for graph

Country	Phone users	Internet Users
Australia	77%	83%
US	72%	88.22%
Russia	45%	71.30%
Germany	60%	88%
Venezuela	45%	54.90%
France	49%	89.46%
Brazil	41%	60.10%

Table for the graph of internet and phone penetration statistics [9] [10].