iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VIII **Month of publication:** Aug 2023

DOI: https://doi.org/10.22214/ijraset.2023.55394

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VIII Aug 2023- Available at www.ijraset.com

E-Voting System on Blockchain

Anmol Panwar
Goa Institute of Management

I. INTRODUCTION

When we hear the word 'democracy', the concepts like – *right to choose, respect for individual rights, freedom of speech, regular and fair elections* etc are the most common ones that arise in one's mind. The constitutional laws have been designed in such a way that the features of democracy stand preserved. But, there always is a slight deviation between what's written on the paper and when it's implemented in the real world. Etymologically, democracy means a system of government where common people hold the political power. Yet the way the democratic practices are administered, still carries a huge burden of accusations and objections, and the term 'democracy' tops the list of the most indiscriminately abused terms in history. Although its structure and methodologies have greatly transformed since the beginning but the remarkable indicia of democracy – the elections, are still frowned upon and for no bad reasons.

A free and fair election is one of the most fundamental pillars that supports the democratic system. The foundation of the democracy lies in the fact that common people have the right to choose their candidate and rule through the elected representative. And this foundation, up until today with so much advancement in technology, fails to mark its presence. We have built buildings taller than mountains, we've made computers that work faster than a picosecond and when it comes to develop a flawless and a reliable system for elections, what are we up to – electorate manipulation, bought votes, ballot stuffing, sabotaged EVMs, ridged elections?

The process of elections is the heart of every democracy and thus demand a great deal of security and privacy to preserve their fairness. In fact, the election security is a matter of national concern. But, in spite of all the mindfulness and cautions, we still fail to conduct secure and fair elections and not only in India but across the globe. From the dawn of the democratic elections, the process of conducting elections has been practiced via paper ballots. Ballot is a system of elections in which the voter marks the candidate he/she wishes to choose on a piece of paper, known as a ballot paper. These ballot papers are then collected and the votes for each candidate are tallied up manually to form the result. This system had some major drawbacks because counting the votes manually does not account for human error and also, it can be easily tampered with. They were heavily criticised because of various news of fraudulent votes and captured booths. Another major drawback it suffered was the 'Spoilt votes' – where the voter's vote cannot be counted due to improper marking on the paper or when there was a case of multiple marks by the voter. So, the system was fault tolerant neither from the voter's side, nor from the perspective of the administration.

It was critical to replace this system with something else to regulate fraud and to develop a truly robust system of elections. Fortunately, moving on to a new approach for conducting elections via a device called Electronic Voting Machine (EVM) gave some relief. It wasn't any longer until came to light the cases of faulty and corrupted EVMs. And the allegations thrown at the EVMs cannot be claimed as false because after all, the system is centralised and anyone, to whom the EVM is physically accessible, can easily sabotage the EVM and the voters' privacy and right to choose are jeopardised. And we circled right back to the starting point. The cases of tainted EVMs pushed several countries back to the ballot system. Countries like Netherlands, Finland, Germany etc believe that there are various security issues with the Electronic Voting Machines and claim that the risks of EVMs outweigh its benefits.

The bitter truth about democracy is – 'Security is the biggest lie in a democracy'. And the big questions still remain unanswered:
How can we secure our elections?
How can we design a truly secure and trust-less voting system without giving a central authority, full control over the process? The answer is 'Blockchain'.

A. What is a Blockchain?

In I.T. systems, a lot of data is generated and it must be stored somewhere in an organised way. These organised collections of data are known as databases. In sectors like financial systems, healthcare systems or E-commerce websites etc where data belonging to various users is generated every second, and the confidentiality of data is a matter of concern, then the following questions arise:

1) Where the database must be located? If at an allegedly-claimed-as-secure central location, then why?
2) Who should decide the rules for writing the data into the database?

©IJRASET: All Rights are Reserved

3) Who must be allowed to write the data in the database?

To answer these questions, we let everybody (every participant in the system, i.e. users) have their own copy of this database. Now the database is distributed among the peers and is decentralised over the whole network.

Then we introduce a few concepts of cryptography and distributed computing while imposing a consensus protocol among the network. And the Blockchain technology is born.

Blockchain is a decentralised distributed immutable ledger which holds any kind of data and is not available to or controlled by a single party rather by all the participants of the system.

Blockchain, just like a database holds an ever increasing record of structured data and is secured by various cryptographic techniques like Public Key Encryption, Digital Signatures and hashing algorithms like SHA family. But, what separates a Blockchain from an ordinary database is the fact that Blockchain is not controlled by or available at a central location/organisation.

The development of a blockchain database involves these major steps:

A peer to peer network of nodes that participate in a system (like a transaction system, but without a bank and rather a peer-to-peer system)

The transaction (or any data) generated among the nodes is signed digital by the one who generated it and must be broadcasted among the whole network. This maintains the authenticity of the data.

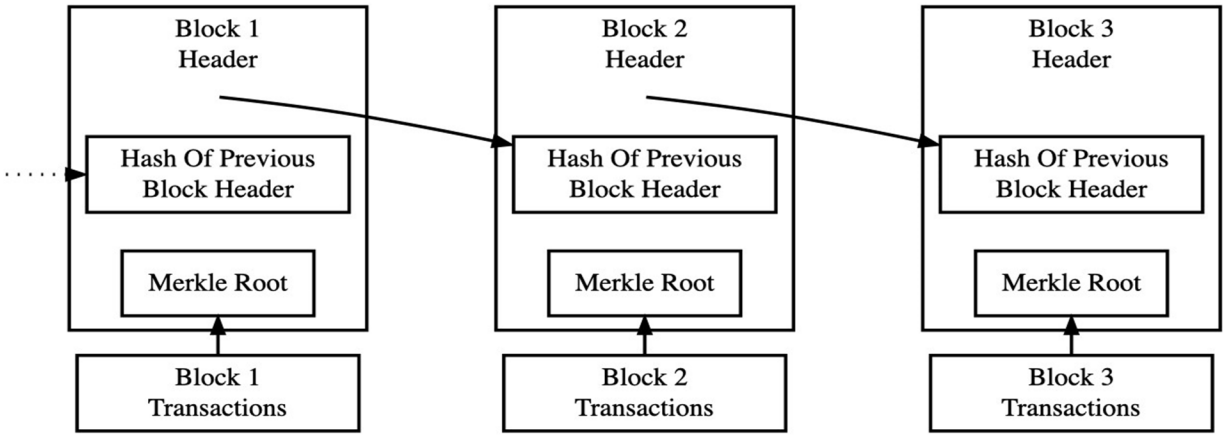
The other nodes when receive the data, store it in a temporary pool of data/transactions. The data is not yet written on to the blockchain.

Now to ensure that all the nodes in a network are in consistent state (or in sync) with each other, we divide the data into small files called *'blocks'* and to define an order among those blocks we *'chain'* the blocks together such that the data of a block contains the hash of the data of the previous block and so on.

The blocks are connected to each other in a chain of hashes, thus it is known as Blockchain.

In order to get over the problem of fraudulent data injection among the nodes by an attacker, we propose a 'Proof-of-Work' protocol, which states that to add a new block to the blockchain and broadcast it to the network, the node must solve a cryptographic puzzle which requires a heavy computational work to be done before adding the block.

This makes the blockchain immutable because the data in any block would result in changing the hash of that block along with hashes of all the subsequent blocks and would changing require an infeasible amount of work to be done to recalculate the hashes of those blocks in the blockchain.



In this paper, we explore the potential of Blockchain to develop the democratic voting process more secure and reliable and making it a trust-less system by having a simple truth as the foundation of our knowledge, i.e. *'unless the honest participants outnumber the corrupt ones, the integrity and security of blockchain ecosystem is preserved'*.

II. PROPOSED WORK

An election is nothing but a political competition to gain political power and rule the state/nation. And without any deeper look into the process, one can almost certainly say that the result of such political competition lies in the very hands of the participants of this competition itself. Thus, the system can be undeniably rated as absurd.

Thus, in order to secure the purpose of our democracy, we need to revise the fundamental concepts of our elections. To achieve this, the first and foremost idea that must be implemented is that the results must not be controlled and located under the authority of participants (political parties) themselves, rather the control must be given back to the people as the democracy dictates.
In this paper, we propose a system of Electronic Voting done via Distributed Ledger Technology, that not only allows the people to cast votes, but also to keep the process decentralised and distributed among the people themselves and keeping it away from any potential threat.

A. Election Design Principles

The objective of democratic elections is to enable common people to harness their right to choose their leader without disclosing it to anyone. To ensure the security of elections while maintaining voter's privacy we must fulfil the essential requirements in order to effectively conduct fair elections in the name of democracy. These requirements are as follows -
Only eligible beings must be allowed to cast vote, i.e. in India, above the age of 18
Voting under the influence of any kind of coercion must be restrained
Any vote in the election must not be traceable back to the voter in any case
The election system must be able to prove to the voter that their vote has indeed been counted and has not been tampered with
The integrity of the election data must be preserved and it must be tamper-resistant in every way

B. Methodology

1) Voter Eligibility And Authentication

We propose a system to securely authenticate the eligible voter.
Voter Eligibility
First of all, we need to make sure that the voter is eligible to vote and that can be done by checking the voter ID from the government voter ID database. For that we develop a form on the home page of the E-voting portal and match the voter ID through the database. If found eligible, the voter shall proceed to choose his/her candidate, else not.
Voter authentication
Then, we need to verify whether the voter is authenticate or not, i.e. by checking the following two conditions:
Whether the voter has already voted in the election at some other time
Whether the voter is voting from the same district that he is supposed to vote from If one (or both) of the conditions fail to be verified, the voter will not be allowed to vote for the elections.

2) Securing The Voter's Privacy And Vote Data

Our system should function in such a way that the votes cannot be traced back to the voter and the voter's identity shall remain anonymous. To achieve this we implement a hashing algorithm to hash the voter IDs and store them on the Blockchain with the corresponding vote data. Hashes are infeasible to compute in the reverse direction.
That means, given a particular hash 'H' of some data 'x', we cannot determine the value of x such that:

$$h(x) = H$$

(where h is the hashing algorithm, by which the hash H was obtained)

So, hashing the voter ID and storing that in the Blockchain is probably safe because the hash cannot yield the information back. Also, running all possible inputs to guess-and-check the output hash against the given hash will definitely not solve the problem as discussed above.
But, this gives rise to another interesting problem.
When the data that is to be hashed has a limited (and feasible) number of values, the security offered by hashes might be compromised. Because an attacker might simply run the hashes of all possible permutations of data values. So implement a technique known as *Zero Knowledge Proof (ZKP).*
Zero Knowledge Proof
In cryptography, a Zero Knowledge Proof (or Zero Knowledge Protocol) is a technique in which one proves to another, that they have some particular information 'X', without disclosing the information to them at all.
This can be implemented by appending a value (a 4-6 digit pin) after the voter ID and then hashing and storing the value in the Blockchain. In such scenario, the attacker need both the voter ID to match the hash in the Blockchain and a 6 digit pin will increase the difficulty of finding out the data associated with that hash by 10,000 times.

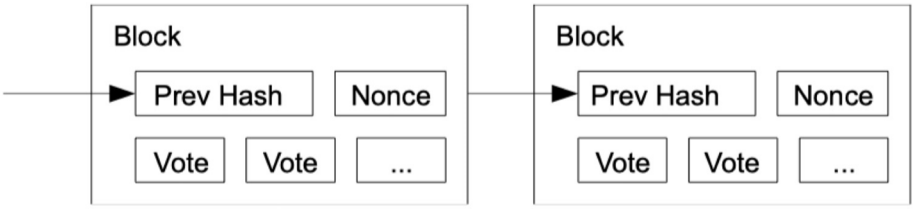
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VIII Aug 2023- Available at www.ijraset.com

3) Implementing the consensus algorithm – 'Proof-of-Work'

This is the consensus algorithm that is the heart of the Blockchain. The Proof-of-Work consensus protocol dictates that in order to verify an information, we should not trust the peer nodes on the network, but rather we must trust the computational work. In other words, when a node hears a transaction 'Tx' from a peer on the network, it must not trust this transaction by verifying it from the peer nodes but when a considerable amount of computational work has been put into it.

In our blockchain, when the votes are broadcasted to the network, everyone (including the sender) stores it in a temporary pool of unconfirmed votes and after some time when the previous block has been mined, every node must put it into a block and find a number such that the hash of the block contains a particular number of zeros in the beginning as dictated by the protocol.



a) Daisy chaining in computing the Proof-of-Work

The confidentiality and authenticity of data is maintained by the help of cryptographic methods like digital signatures, public key encryption etc. But, the integrity of the blockchain is maintained by the power of computational work done on the blocks. This is done by finding the correct *nonce* for each block in the blockchain.

In public blockchains, this process is done independently by the nodes in the network. However, we realised that in a private or permissioned blockchains, this is not really necessary. If the participants in the network already know and trust each other, like in our private blockchain network, we can increase the difficulty of finding the proof of work by some factor and in order to compensate that difficulty, we can divide the solution space in which the brute forcing to find the nonce is to happen.

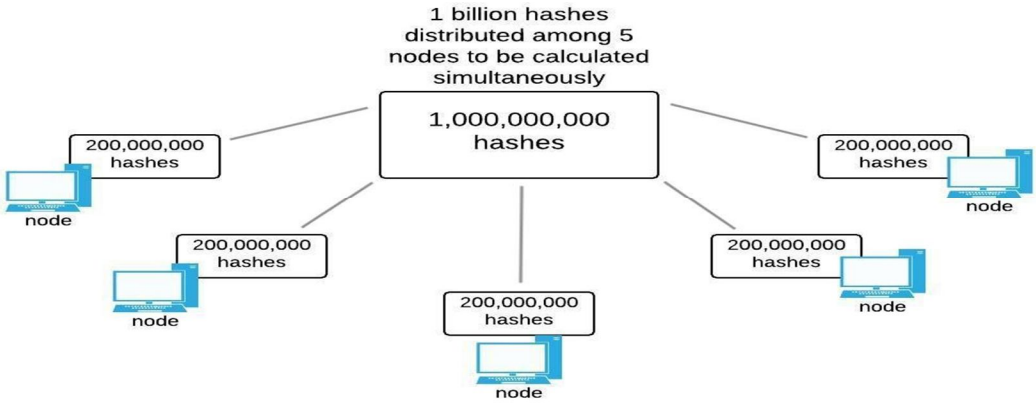
b) Implementation

Let's assume that the current cryptographic difficulty is 30 zeroes. Means that the hash of every block must start with 30 zeroes in 256 bit representation.

The total possible nonce we have check is 230 ≈ 1 billion.

In this example, suppose we have 5 nodes in the network.

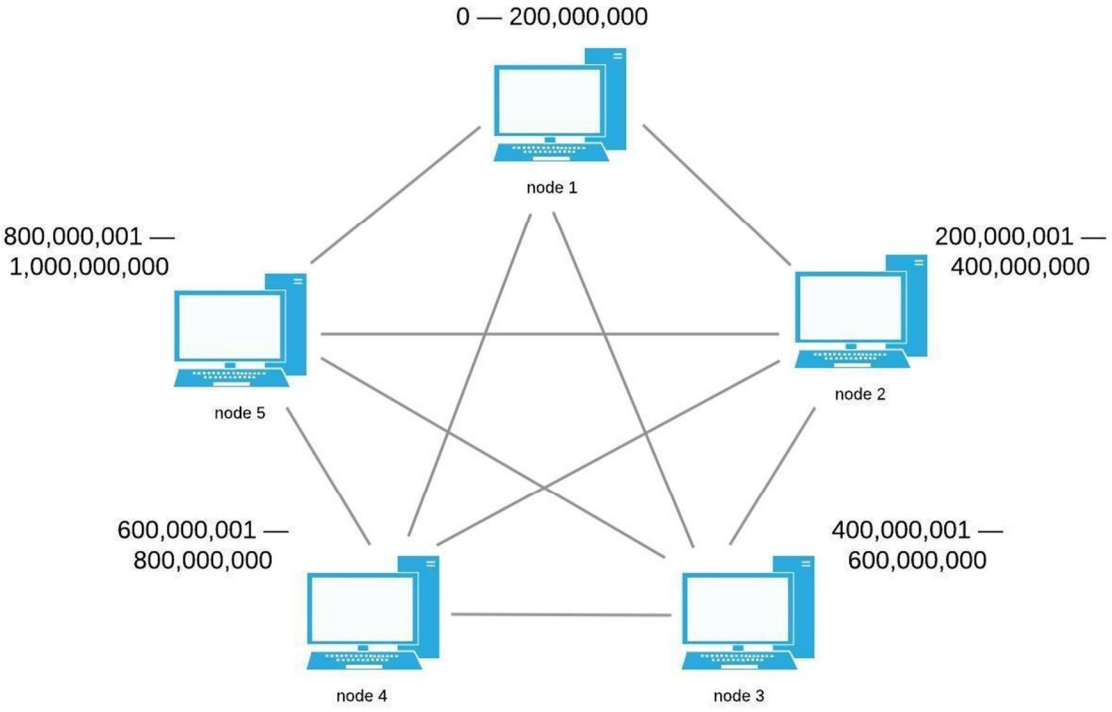
So we calculate the total number of possible hashes, we would need to run and we split them among the nodes in the network. The more the devices are there in our network, the more computing power we achieve.



After splitting the proof of work among by the number of nodes, we can allot the range in which each device must calculate the nonce. This reduces the redundancy in computing the proof of work, and unlike the public blockchains, this is not an independent race to find the nonce, rather an organised teamwork.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VIII Aug 2023- Available at www.ijraset.com



4) Broadcasting the information on the Network

Every information generated on a node will be broadcasted to all the other nodes on the network. When a vote is generated on a system, that information will be broadcasted to the network and everyone will save that vote in an unconfirmed pool of votes. Then, after the current block has been mined, the nodes (peers) on the network will pack the unconfirmed votes into a block and start working on the proof-of-work for the current block. When a node finds the right proof-ofwork, that block will be broadcasted on the network again, and the rest of the nodes will verify that block by hashing the block header and matching with the broadcasted proof-of-work.

5) Verifying the votes

In case, a voter feels insecure that whether his/her vote has been counted or not, or whether it has not been tampered with, he/she can indeed verify that by signing in to the portal using his/her voter ID and the algorithm will show that the information is indeed safely written on the blockchain.

If by any means, the vote data has been tampered with, then the resultant hash of the data will not match the signed hash attached with the data. Even if someone decrypts the hash using the voter's public key, even then it cannot be encrypted back by the private key of the voter and thus, the algorithm would simply deny any such change.

Take your Private key with you

The program generates a private/public key pair for the voter and the public key is uploaded in the block and written to the Blockchain safely along with the *'hiddenVoterID'* field that has the hashed data of VoterID and the PIN/Password chosen by the voter.

Also, there is an option to take the information of the voter's PIN and his/her private key with themselves safely either by scanning or printing a QR Code generated by the program.

Irrespective of what the voter may choose, (whether to scan/print the code or skip the option and leave), the QR code and its information will be immediately deleted from the voting device and the voter shall not be able to recover the QR code by any means. Thus, it is better to 'take your Private key with you' and in any case if one wants to verify their vote, they can check that with the help of their login credentials and the private key.

Thus, the voter can rest assured that the vote is signed by his own digital signature and is tamperproof.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

6) Observe Results

After the elections are declared over, the administrator node will tally up the votes looping through the blockchain, one block at a time and first verifying the data and the signature with the help of the public key and then decrypting the data by his own private key and counting the vote.

So, the process involves these tasks to be performed repetitively:

a) Unlocking/Decrypting the AES key by the private key of the Administrator.

b) Decrypting the data from the obtained AES key.

c) Verifying the digital signature by the voter, using the voter's public key provided with the data.

d) Obtaining the information about the chosen candidate by the voter and counting up the vote.

This result can then be displayed later as per the schedule.

III. RESULT AND ANALYSIS

In this paper, we proposed a trust-less decentralised distributed voting system with an intent to conduct free and fair elections and give the control back to the common people. But, to ensure that the system really functions as advertised and solves the aforementioned problems, then we need to carefully assess the various aspects of the proposed system.

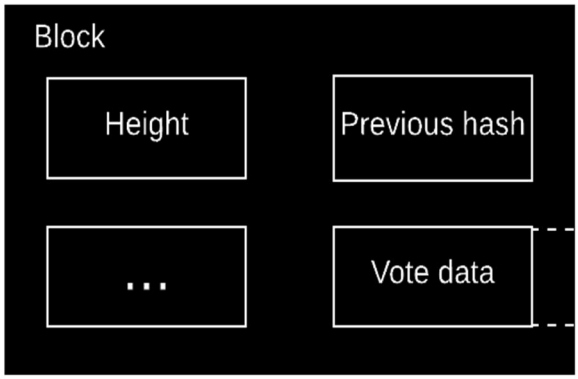
In this section, we will analyse the those aspects related to the practicality and effectiveness of the proposed system. Also, we will try to observe the vulnerabilities and any potential threats and we will gauge the severity of the impacts posed by those threats on our system.

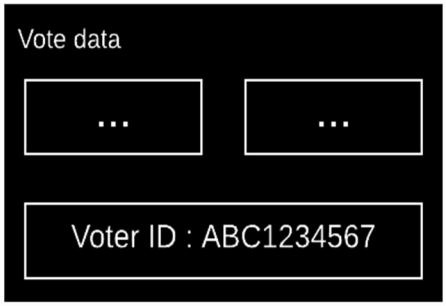
A. Voter Traceability

One of the most important features about an ideal election system is that any vote must not be traceable back to the voter. It must not reveal any information about the voter whilst being stored on the publicly accessible (in the private network) blockchain database.

Every vote data packet that is being transmitted on the network and stored on the blockchain contains majorly 3 things — Voter's public key, AES encrypted vote data (time, candidate chosen and hidden voter ID) and RSA encrypted AES key for decrypting the vote data.

In order to enable an election system which can prove to the voter, that their vote is indeed safe and has been counted and counted correctly, the voter ID must be stored on the blockchain along with the corresponding vote data. But, this makes the voter ID vulnerable to passive attacks.





For example, someone might be monitoring the data transmission over the network via Traffic analysis and might be able to decrypt the message, which is highly unlikely though, as the RSA encryption is considered a secure way to encrypt data and thus the modern networking technology hugely depends on it. But, if in the worst case scenario that does happen somehow, then the voterID linked to that vote is not confidential anymore.

To achieve this, we apply a technique called *Zero Knowledge Proof (ZKP).*

According to this methodology, one party can actually prove to other party that they have an information 'X' without revealing any part of that information to the other party.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VIII Aug 2023- Available at www.ijraset.com

For example —
A party ‘A’ has to give an information ‘r’ to other party ‘B’ without disclosing it. Then, if the possible value spectrum of ‘r’ is big enough or it has a high minimum entropy, that is infeasible to be computed by brute forcing, then the proving party ‘A' can just hash the information ‘r’ and give that hash to the latter one.

$$r \longrightarrow h(r)$$

But, if the value ‘r’ has a low minimum entropy, like ‘r’ is a birth month of the party A (having a small possible values) and party ‘B’ knows that ‘r’ is a month, then hashing all months and comparing to the information given by ‘A’ won’t work.
So, what ‘A’ can do is, take a random number ‘X’ and append it with ‘r’ and then hash the data.

$$h(r \,||\, X)$$

Now, without knowing the value of ‘X’, party ‘B’ can never find out the information. And if ‘B’ wants to verify, ‘A’ can provide the values of ‘r’ and ‘X’ and it checks out.

1) Implementation



To hide this voter ID using the concept of ZKP, we ask the voter for a 4-8 digit PIN/Password while the sign up. Then we append the password with the voter ID and hash the resultant string together using a separator, a simple string to separate both the data. Separator may be as simple as ‘ *** ’ or ‘———’ ; any string that will separate both the values.
By this method, we add the voter ID to the blockchain, without revealing it. In other words, in any case of discrepancy, if the voter wants to be assured that his vote has not been tampered with, he/she can identify his/her vote by proving the corresponding voter ID to the administration, along with their password. Until then, no one can trace back the vote to its respective voter.

2) Analysis
This is the implementation of Zero Knowledge Proof to hide the voter ID even if it is present right in the blockchain ledger. But to find out really how well this system works, we need to analyse what it takes to fool this system.
We already know hashes cannot be reverse engineered and the only way someone can find out the data, that was hashed, is by a *Brute Force method. Time consumption analysis in Brute forcing technique*

Object	Length and format	Total permutations
Voter ID	3 digit state and district code, 7 digit EPIC number	7.2 x 107 (≈ 7.2 billion)
PIN / Password	8 digit alphanumeric	21.8 x 1013
Appended (Voter ID ⊕ PIN)	21 characters = voter ID + PIN + 3 byte separator	15.7 x 1021

Hash rate of an average computer GPU = below 1GH/s (Giga Hashes per second)
For a margin let us consider a highly motivated individual, in order to take down the system gathers a computing power of about 10 GH/s.
Time consumed in brute forcing ONE voter ID:

$$= (15.7 \times 1021) / 1010$$
$$\approx 49784 \text{ years}$$

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VIII Aug 2023- Available at www.ijraset.com

Moore's Law — This law states that the number of transistors in a processor chip doubles every two years and the cost of its production is halved.

Taking into account the Moore's law, this number — 46784 years, is still an unrealistic amount of time frame to achieve the objective within.

B. QR code handling

After the voter has casted his vote, the voter is presented with a mandatory QR code to either scan by their smartphone or to carry a print out of the QR code with them. This QR code contains the private key generated for the voter and their password.

This QR code does not affect the vote already casted by the voter and its loss will only affect the opportunity to verify their vote's integrity. However, the data of this QR code if stored in a phone, is safe but for those people who chose to carry a print out copy of that QR code, are susceptible to the probability of this print out being lost somewhere which may pose a problem in the optional vote integrity verification.

C. Daisy chaining in Proof-of-Work analysis

Theoretically, the daisy chaining seems to work well enough. Examining the practicality of this technique is very crucial because the more the computing power is achieved, the more the integrity of our blockchain will be. Thus, more difficult it will be to gather the computational power that would suffice the purpose of changing the data of the blockchain.

Running the results for the time taken in computing the proof of work independently (represented by t) versus time taken when the daisy chaining is in action (represented by t_{dc}) gives us the following results in 6 different cases:

where — n = Nodes in the network

d = Difficulty of cryptographic puzzle (number of zeroes in 256 bit representation)

given that —

Network latency (ping) =< 5 ms **n = 3; d = 8**

Case = 1	t = 56.64	t_{dc} = 15.44
Case = 2	t = 47.93	t_{dc} = 19.76
Case = 3	t = 57.21	t_{dc} = 9.37
Case = 4	t = 51.56	t_{dc} = 16.26
Case = 5	t = 63.32	t_{dc} = 13.03
Case = 6	t = 38.14	t_{dc} = 10.71
Case = 7	t = 49.84	t_{dc} = 12.28
Case = 8	t = 69.29	t_{dc} = 16.33

Adding a node in the network and increasing significant amount of difficulty showed no significant increase in average block mining time. The result are as follows:

(For comparison, going from d = 8 to d = 10 must increase the difficulty 22 times, i.e. 4 times. But, that proportion is not observed in the rising of block mining time.) **n = 4; d = 10**

Case = 1	t = 188.23	t_{dc} = 26.32
Case = 2	t = 221.47	t_{dc} = 33.94
Case = 3	t = 297.52	t_{dc} = 41.05
Case = 4	t = 191.08	t_{dc} = 35.68
Case = 5	t = 203.98	t_{dc} = 46.99
Case = 6	t = 224.59	t_{dc} = 31.82
Case = 7	t = 173.38	t_{dc} = 39.72
Case = 8	t = 241.66	t_{dc} = 42.77

A noticeable fact observed from the results is that the gap between the average values of time taken in case of independent computation and daisy chaining keeps on increasing as we increase the difficulty. And thus, it can be concluded that as the difficulty increases and so do the number of nodes in the network, the independent proof of work computation method will never be able to catch up with the daisy chaining method in proof of work computation.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

D. DDoS attack

Distributed Denial of Service is an attack performed on centralised databases, like a website hosting server or any database. It works by flooding the bandwidth of the system with a large number of requests and slowing down the response or even blocking the genuine requests. Blockchains being decentralised are prone to this attack. In our private blockchain the nodes would simply deny any such request and would only respond when the peer node in the network requests something.

Also, if in the worst case scenario, if the connection of the node is disrupted, the node would simply carry on to store the votes and will broadcast all the data to the network, soon as the internet is restored. Also, it will update the blockchain data to synchronise with the other nodes.

E. Sybil attack

Sybil attack is an attack commonly executed on peer to peer networks. It involves creation of several identities in the network, much like one can create many different accounts on a social media platform.

This is done to achieve a fair amount of authority in the system, but in blockchains the number of identities does not matter. This is easily eradicated by the proof of work consensus protocol which dictates to trust computational work, not other nodes in the network.

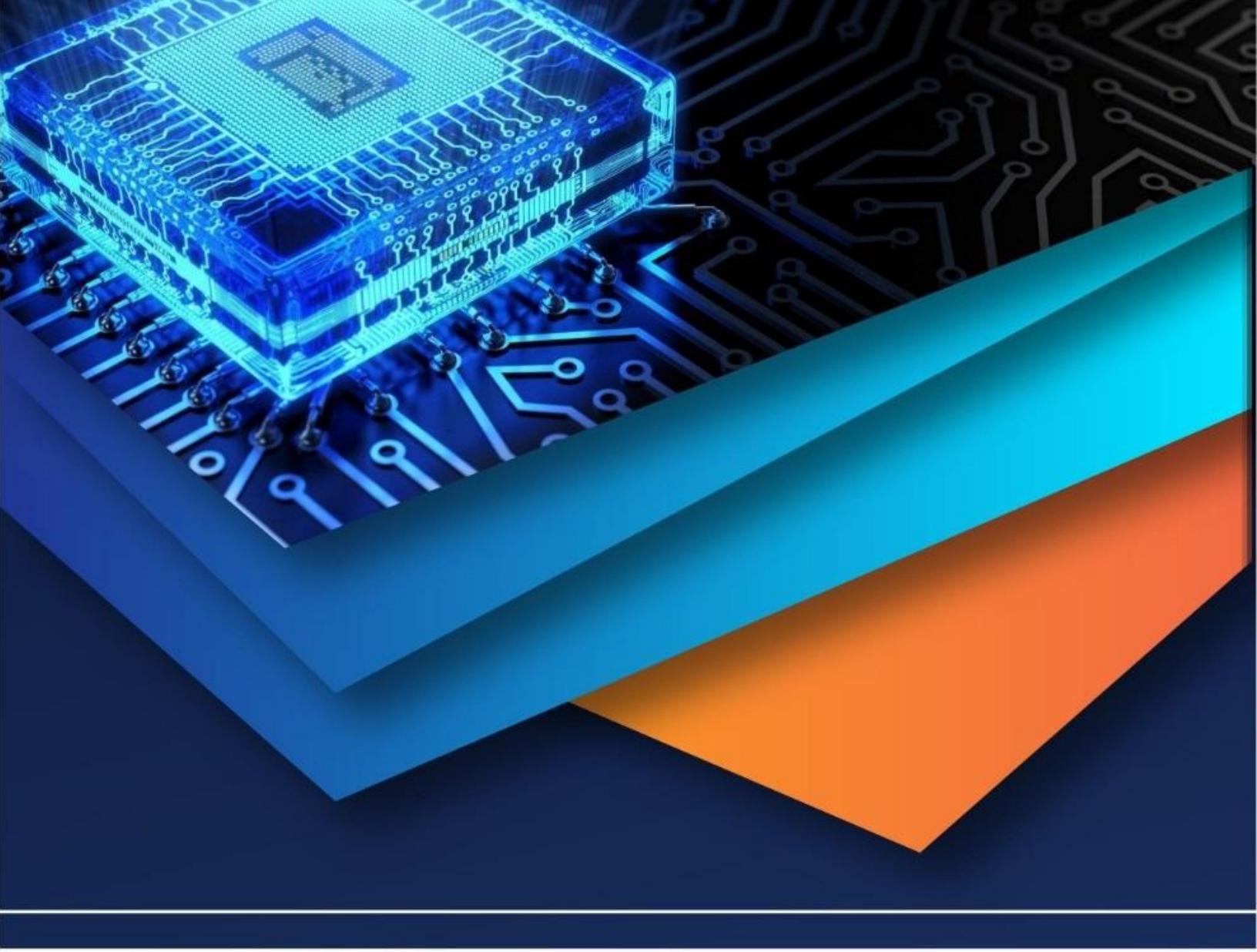
Potential vulnerabilities and their damage severity evaluation

Potential vulnerabilities	Nature of vulnerability	Cost to execute	Probability of success	Damage severity
Voter ID traceability	Manually executed	Very high	Infinitesimal	High
QR code mishandling	Natural	N.A.	Medium	Very low
Sybil attacks	Manually executed	Very high	Very low	Very high
51% attack	Manually executed	Very very high	Infinitesimal	Very high
DDoS attack	Manually executed	High	Very low	Medium
Internet connectivity	Natural / Manually executed	N.A. / Medium	Medium	Low

REFERENCES

[1] Satoshi Nakamoto (2009) – Bitcoin: A Peer-to-Peer Electronic Cash System. Available at – https://bitcoin.org/bitcoin.pdf
[2] Skemman (2012) – Blockchain-Based E-Voting System. Available at – https://skemman.is/ bitstream/1946/31161/1/Research-Paper-BBEVS.pdf
[3] Sos.ca.gov (2007) – Top-to-Bottom Review | California Secretary of State. Available at – http://www.sos.ca.gov/elections/voting-systems/oversight/top-bottomreview/
[4] Nicholas Weaver (2016) – Secure the Vote today. Available at – https://www.lawfareblog.com/ secure-vote-today
[5] Vitalik Buterin (2015) – Ethereum White paper. Available at – https://github.com/ethereum/wiki/ wiki/White-Paper
[6] Anmol Panwar (2020) – Here's why 'Blockchain' is not yet ready to gain traction. Available at – https://medium.com/@anmolpanwar8/heres-why-blockchain-is-not-yet-ready-to-gaintraction-ac92c323d84d
[7] Andrew Barnes, Christopher Brake and Thomas Perry (2016) – Digital Voting with the use of Blockchain Technology. Available at – https://www.economist.com/sites/default/files/ plymouth.pdf
[8] Rumeysa Bulut, Safa Keskin (2019) – Blockchain-Based Electronic Voting System for Elections in Turkey. Available at – https://arxiv.org/pdf/1911.09903.pdf
[9] Vincent Gramoli (2016) – On the Danger of Private Blockchains (When PoW can be Harmful to Applications with Termination Requirements). Available at – https://www.zurich.ibm.com/dccl/ papers/gramoli_dccl.pdf

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)