**Fortifying E-Voting Privacy: A Triad of Blockchain, Convolutional Neural Networks, and Quantum Key Distribution for Unparalleled Security**

Bhaumik Tyagi

**ABSTRACT**

The advancement of blockchain has facilitated scholars to remodel e-voting systems for the future generations. Server-side attacks like SQL injection attacks and DOS attacks are the most common attacks nowadays where malicious codes are injected into the system through user input fields by illicit users which leads to data leakage in the worst scenarios. Besides, quantum attacks are also there which manipulate the transactional data. In order to deal with all the above-mentioned attacks, integration of Blockchain, CNN & Quantum Key Distribution is done in this very research. The utilization of blockchain technology in e-voting applications is not a novel concept. But privacy and security issues are still there in a public and private blockchain. To solve this, the use of hybrid blockchain is done in this research. This research proposed cryptographic signatures and blockchain algorithms to validate the origin and integrity of the votes The convolutional neural network (CNN), a normalized version of the multilayer perceptron is also applied in the system to analyze visual descriptions upon registration in direction to enhance the privacy voters and e-voting system. Quantum Key Distribution is being implemented in order to secure a blockchain based e-voting system from quantum attacks using quantum algorithms. Implementation of e-voting blockchain D-app and providing a proposed solution for the privacy of voters in e-voting using Blockchain, CNN & Quantum Key Distribution is done.

**KEYWORDS:** *Hybrid Blockchain, Secure e-voting system,**Convolutional Neural Networks, Quantum Key Distribution, One-time Pad.*

1. **INTRODUCTION**

E-Voting or Electronic voting refers to voting via the internet using any digital device. The vital challenge for e-voting is the major security risks it might cause as an electronic voting system essentially relies on the internet. Several countries are willing to begin electronic voting for a variety of purposes such as accessibility and decreased costs. Amid the increasing use of the internet, SQL injection attacks, DOS attacks, and Quantum Attacks are the common techniques to outbreak e-voting systems. In, SQL injection attacks, attackers inject some SQL codes into the original code in order to obtain or abolish delicate information. Timing attacks, Blind injection, union queries, illegal queries, and tautologies are some categories of SQL injection attacks. Blind SQL injection attacks are used by hackers in which the hacker requests true or false questions from the user. Using Tautology statements, attackers inject codes into the authentication phase, which says 1=1 is always true so that the injected query becomes valid even if the invalid password and username are entered. In order to join the original query with an injected query, the union operator is used while injecting codes. In recent times, the major apprehension about the e-voting process is its security fallouts. To counter these security issues and attacks, numerous studies have been done but not collectively and completely. Encrypting and decrypting plain-text messages while sending electronic data is the most popular application of cryptography. This technique encrypts data using a symmetric ("secret key") approach. The recipient is then provided with both the secret key and the encrypted message to be decrypted. In the event that a third party obtains the message, they have all the necessary tools to read and decode it. To solve this issue, "public key systems "(asymmetric) are formulated by cryptologists as a solution. In an asymmetric system, each user has two keys: one private & one public. The communication is encrypted and sent after the sender requests the public key. At the time of message arrival, only the recipient’s private key will decode it. To achieve a precise conclusion, implementation and optimization are required in existing e-voting research. The proposed solutions do not guarantee the security or privacy of voters from various aforementioned attacks. The integration of blockchain technology with convolutional neural networks (CNNs) & quantum key distribution (QKD) could provide secured e-voting solutions. The convolutional neural network is a regularized version of the multilayer perceptron in the system, it is employed to analyze visual imagery upon registration. The goal of this research is to modernize cutting-edge blockchain-based electronic voting systems. Quantum attacks can be implemented on blockchain-based systems using algorithms like Grover’s technique. Despite of being a secure technology, security of blockchain systems can be compromised by imposing quantum-led attacks.

Using such algorithms, the blocks can be easily substituted. Also, the ample record of chains can be refabricated once again by speeding nonce’s creation and quickly alters the hashes, jeopardizing the Blockchain's integrity. This research provides solution for privacy and security from attacks like Quantum attacks, SQL Injection attacks, DoS attacks, computational over-head, Stale block, Blockchain forks.

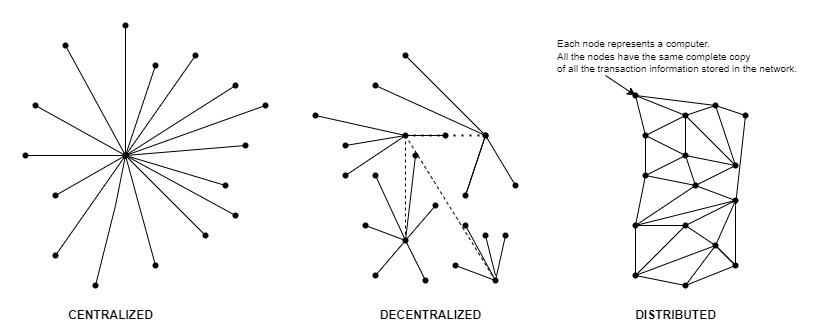
The most significant standards in order to create a secure voting system are as follows:

1. Unauthorized voters should not participate.
2. Authorized voters should not be allowed for casting two votes.
3. Voting is confidential.
4. Voter’s choice should not be public.
5. Verifiability: Voters must be able to verify that their votes are correctly calculated [1].
6. Privacy of voters should be at first priority while creating any voting system. Identity of voters should not be disclosed for whom they voted for.
7. Once the Vote is cast and verified then it must not be directed or controlled in any means.

Traditional voting systems are not preferred over e-voting systems in today’s era. Reason behind this is execution time in traditional voting systems is slow as compared to e-voting systems. E-voting systems are immutable whereas traditional voting system are mutable. Manual workflow is required in traditional voting system which in turn makes it expensive. E-voting systems are cheaper option because of automated workflow. Traditional voting systems are prone to error whereas e-voting systems are error free. Traditional voting system’s major drawback is they are less secure and purely centralized whereas e-voting systems are secure and decentralized.

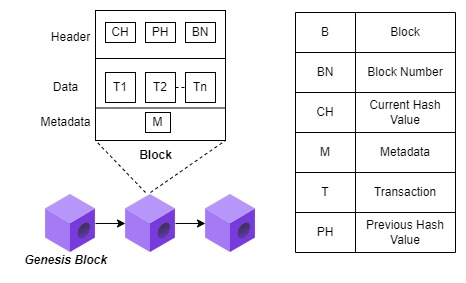
1. **BLOCKCHAIN**

Blockchain (BL) is an open distributed ledger. Open; Transactional information is public and available to all and a local copy of every transaction is available to every block. Blockchain is decentralized and fault-tolerant technology. No Central authority or third party is there to control the entire BL process. BL is efficient in terms of security, speed, and scalability. Verification of valid information is done before moving to further transactions. Every information that is stored once in the blockchain after validation is immutable or permanent. Hackers cannot hack the whole blockchain as in order to hack a single block, they should be able to hack another block as well.



**Fig 1: Centralized vs Decentralized vs Distributed**

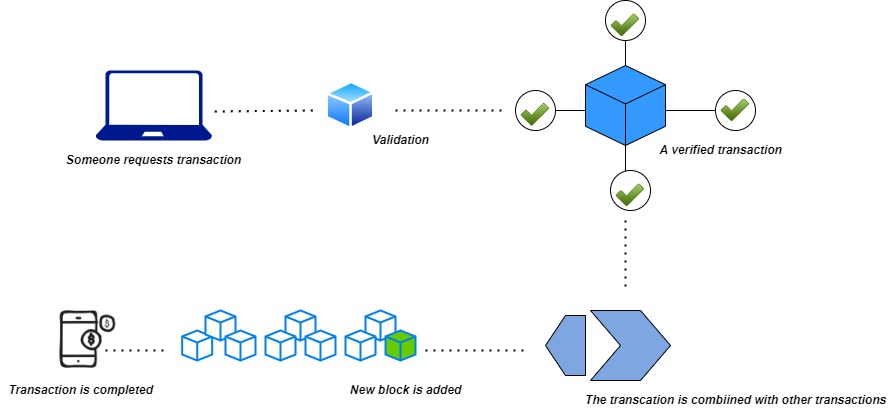
Centralized processes or networks are those in which power is given to central authority. For example: Banks, Traditional Voting systems etc. Decentralized processes or networks are those in which there is no involvement of central authority or any third party. Distributed Network or P2P(Peer-to-Peer) Network is a network among different Blockchains. All these blockchains will verify each other’s block data one by one & in case of any difference in any of the blockchain, the majority blockchain will decide what will be the data of the corrupted Blockchain. So, real data will get copied again in a corrupted Blockchain. Other BLs will discard the data of the corrupted blockchain. All this verification is repeated in a very short interval of time, so it is nearly impossible for a hacker to corrupt the blockchain or any block in such a short interval of time.



**Fig 2: Structure of a block**

In every blockchain, the first block is always known as Genesis Block. Block(B) contains Nounce, Header, Data, and Metadata(M). Whereas, Header contains Block Number (BN), Current Hash Value (CH), and Previous Hash Value (PH). A Block number is simply an identity number of a block. Before proceeding toward the Current hash value and previous hash value, the Hash should be discussed. So, Hash is a unique value that makes every block unique. The hash value is made up by using 64 hexadecimal character values [0-9] [A-F]. Hashing is done using SHA 256 algorithm (Secure Hash Algorithm). On the basis of [Block No., Nounce, Data and Previous Hash value, Hash value for each block is generated. Suppose, Block 1 has PH *“000acdehk32”* and CH *“453ack9dpe0*” so, the PH of Block 2 would be *“453ack9dpe0”*. Now, is there any way to change the value of hash? Yes, by changing the value of Nounce we can change the value of hash. Nounce stands for (No. used only once). We can change the nounce’s value in order to get the targeted value of Hash. We check if the hash value is available or not by again & again changing the value of Nounce. Metadata(M) contains information regarding transaction details, no. of nodes, and no. of participants.

Blockchain (BC) is a repository that stores an ever-growing collection of information archives known as "blocks." This decentralized system ensures secure data transmission without the interference of any central authority or third party. Each block consists of a collection of transactions that were committed to the blockchain by contributors. An exchange could include technical metadata (timestamp, transaction id). All parties must come to an agreement, known as BC consensus, to notify the chain of the most recent block.



**Fig 3: Blockchain Working**

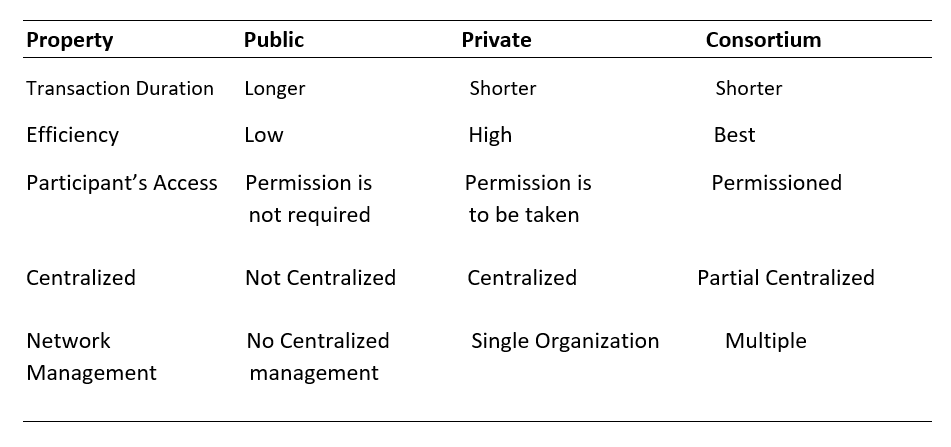
When a new block is approved and added to the chain after all affiliates have reached agreement on it. Decentralization and single point of failure (SPOF) are two traits of Blockchain. Information is maintained in a peer-to-peer environment because each member of the chain is linked to a crucial replica of the ledger. Blockchain is designed to be immutable implies as soon as "block" is augmented, even a tiny modification in the block would stay exceedingly hard. Digital signs, timestamps, and hash functions are a few of the technologies that blockchain is leading.

When astute user wishes to extrude in a block, this will also cause the hash to change, which means that user wishes to achieve perfect agreement for the present one and other blocks that follow the same. Due to BC's scattered and open nature, the ledger cannot be controlled or managed by a central entity that claims it has no "Single point of failure." The three basic classifications of blockchain networks are consortium, private, and public BC. Public BC is a permissionless blockchain in which any person can join a network.

A confirmed transaction can include "cryptocurrency, records, structures, or other information" and is authorised by the network of nodes (any kind). When a user requests a transaction including any kind of meta data, the requested transaction is sent to a P2P network made up of computers called "nodes." As soon as it is confirmed, the novel block is then permanently and irrevocably added to the active block. The transaction is finished once all of these steps have been taken.

Timestamp, user ID, transaction ID, user sign, Markle root, nonce, and prior hash are a few of the metadata that are contained in a block.

**Table 1: Blockchain Network Types:**

****

Blockchain is mainly of 4 types: Public BL, Private BL, Consortium BL, and Hybrid BL.

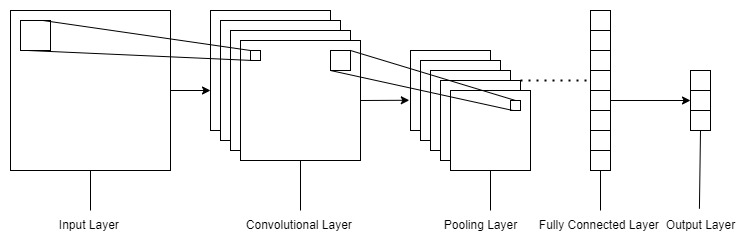
Public BL is open to all, permissionless, transparent, and accessible to all. It is fully decentralized. No permission is required for participant’s access as Public BL is permissionless. Though, Efficiency is low and transaction duration is longer. There is no centralized network management in Public BL. Examples of Public BC are Ethereum, Bitcoin, and Litecoin.

Private BL is private to some organizations which means it is not accessible to all. Private BL is partially decentralized not fully. Permission is required for the participant’s access. Not more than a single organization can participate. Transaction duration is shorter and efficiency is good. Example of Private BL are Ripple, Multichain, and Hyperledger.

In, Consortium BL or Federated BL group of organizations manages the BL. It is more decentralized as compared to Private BL. Permission is required for participant’s access. Transaction duration is shorter and efficiency is best.

Hybrid BL provides features and benefits of Public, Private, and Consortium BL. Hybrid BL is flexible as it works according to the situation, it carries out advantages from all the types of blockchains.

1. **CNN (Convolutional Neural Network)**



**Fig 4: CNN Layers**

Although ANN exists, there are good reasons why it shouldn't be applied to picture categorization. The following are some drawbacks of ANNs for picture classification: a) Excessive computation. b) Aware of where an object is located inside an image. c) Treats nearby pixels the same as distant pixels. As a result, using CNN for image classification is preferred because it offers techniques like ReLU and Pooling.

A convolutional neural network is made up of three layers: an input layer, a hidden layer, and an output layer. The middle layers of any feed-forward neural network are known as hidden layers because their inputs and outputs are hidden by the activation function and final convolution.

Convolutional layers are found in the hidden layers of a convolutional neural network. Typically, this entails adding a layer that performs a dot product of the layer's input matrix and the convolution kernel. As the convolution kernel moves across the input matrix for the layer, adding to the input of the following layer, a feature map is created. Following this are additional layers such as normalization, pooling, and fully connected layers.

* Input Layer
* Convolutional layer: Connection sparsity reduces overfitting; Convolution + Pooling provides location-invariant feature detection; and parameter sharing.
* Pooling Layer: Reduces dimensions & computation.
* Fully Connected Layer
* Output Layer
* Re-Lu Layer: Introduces nonlinearity, which speeds up training and makes computation faster.

CNN is unconcerned with rotation and scale. So, one needs to have rotated, scaled samples in the training dataset. If you don't have these samples, you can create new rotated or scaled training samples from current training samples using data augmentation techniques.

1. **QKD (Quantum Key Distribution)**

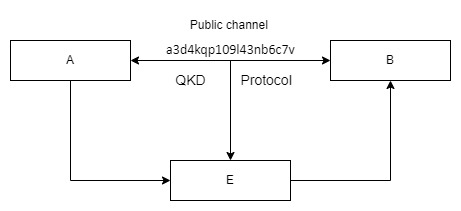
Nowadays, QKD (Quantum Key Distribution) is the emergent technology in the field of quantum cryptography. Quantum Kye Distribution is a well-known and sophisticated method to come out of quantum cryptography. Using the QKD protocol, a random bit stream can be created between two parties. Then, using an OTP (One-time pad) technique, a secret message is encrypted using this random message. QKD allows us to generate a secure key so that secret information can be sent securely from one to another. The main aim of using QKD is security as it guarantees security. This protocol is used to generate random bitstreams for two points of communication.

QKD is used to generate a secret key, which can then be used in the one-time pad scheme for sending messages securely. Quantum-generated keys are securely derived in two places at the same time using QKD. Quantum Keys are more unpredictable & secure than algorithm-based encryption keys. The secret message is then encrypted using the randomly generated bitstream as the one-time pad. Quantum cryptography is displacing established distribution techniques like Diffie-Hellman. The QKD protocol's produced key is used to encrypt the message.

BB84 (Bennet Brassard protocol) protocol is being used for generating a secured shared key. Two sections make up this procedure. Qubits are used to encode classical information in the first section, and the second section is post-processing, which uses no quantum processes at all. In contrast to other mathematically sophisticated cryptographic key distribution techniques, the security of this approach of distributing a secret shared key is provided by the very principles of quantum physics (e.g. Diffie– Hellman). The Heisenberg uncertainty principle is the foundation of the Quantum No-cloning Theorem, which states that it is impossible to duplicate a signal made up of individual quantum particles without introducing detectable flaws.

The goal of the quantum key distribution is for A and B to have a shared random key that is kept a secret from E while initially revealing no private information. Then, whenever necessary, this shared secret key can be utilised to convey critical secret messages using one-time pad (OTP) encryption.

B & A communicate publicly via a traditional public channel while also using a quantum channel to send polarised photons. E is free to conduct one-at-a-time measurements on the quantum channel photons. Additionally, E learns the contents of every message delivered over that public channel without having the power to obstruct such transmissions. B and A discussed and compared signals sent over the quantum channel while testing them for eavesdropping [13]. They performed by using the public channel. Figuring this out is Figure:



**Fig 5: Basic QKD Structure**

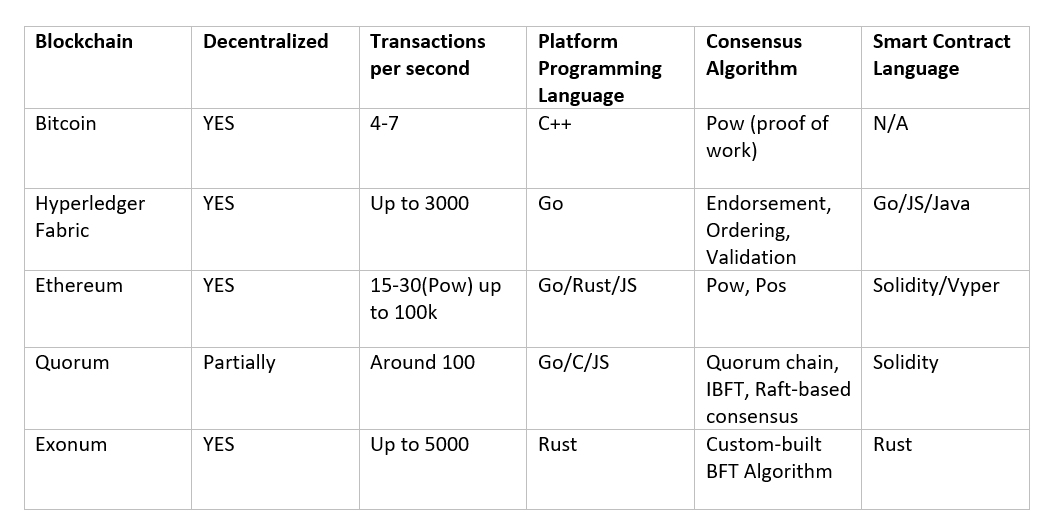
Error removal, Estimating Eve's information, Initialization, Privacy amplification, Quantum transmission are the five processes that make up the QKD protocol [13]. Keep in mind that this description does not address the public channel's authentication problem in order to keep things simple. More elaboration on this point can be found elsewhere [14].

The key is created and distributed via QKD; message data is not transmitted. The message can be encrypted (and decrypted) using the key and any chosen encryption algorithm, after which it can be sent over a regular communication channel [13].

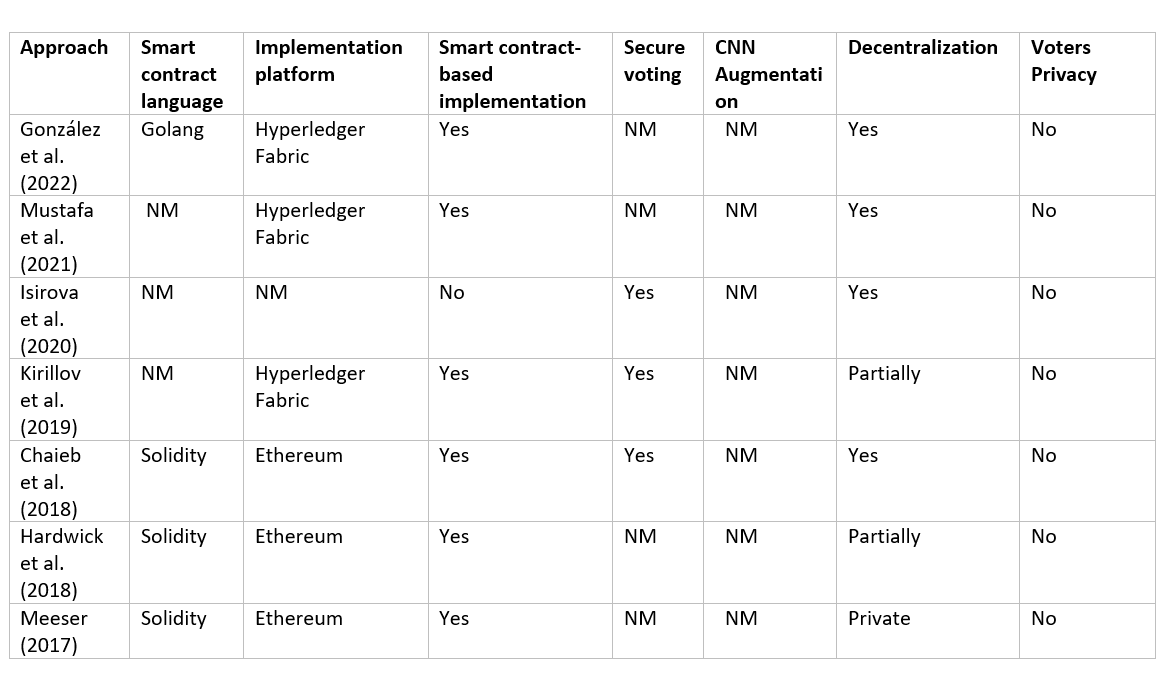
1. **LITERATURE REVIEW**

**Comparative Analysis:**

**Table 2: Blockchain Platform Comparison**

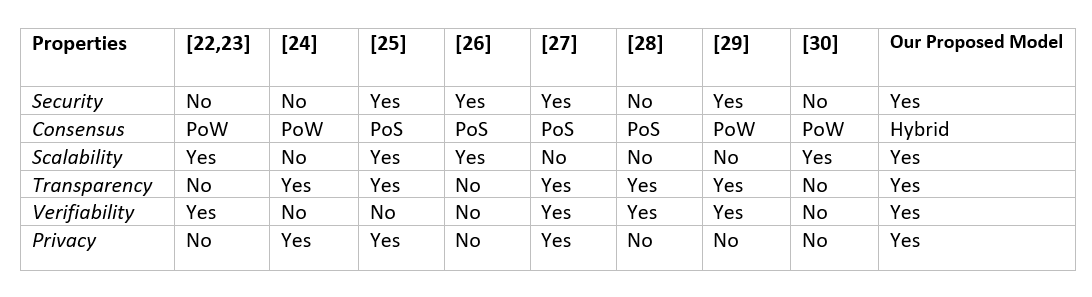


**Table 3: Functionality-based comparison between existing voting approaches**

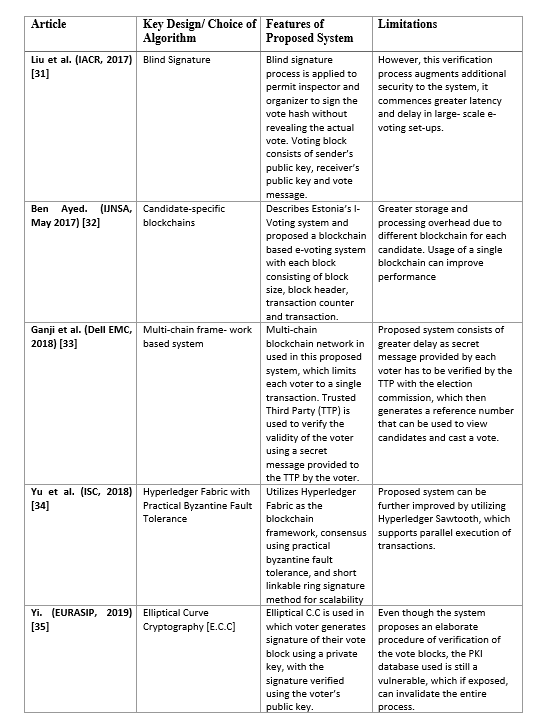


*Here, NM refers to Not Mentioned.*

**Table 4 compares earlier related work with the proposed blockchain-based electronic voting system.**

****

**Table 5: A comparison of current blockchain-based e-voting methods.**

****

Existing Approaches of Gonzalez et.al (2022), and Mustafa et.al (2021) use the implementation platform Hyperledger fabric which is purely private and leads to some sought of centralization which directly hampers voters’ privacy and secure voting. There is no CNN Augmentation and No QKD implementation. Whereas the approach of Isakova et.al (2020) provides secure voting and decentralization but there is no voter privacy. Approaches proposed by Messer et.al (2017), Hardwick et.al (2018), and Chaise et.al (2018) used Solidity as a smart contract language and Ethereum as an Implementation platform but all above-mentioned approaches are partially decentralized not purely which directly hampers Voter privacy and secure voting.

C. Bennett and G. Brassard proposed the original and best-known QKD technique in 1984 [15]. As a result, it is known as the ***“BB84 protocol”.*** In 1989 [16], the foremost experimental BB84 protocol implementation system was constructed. This protocol permits the production of secret keys through a public channel between two parties.

As per the foremost constructs of quantum mechanics, information cannot be gained from intercepted qubits by an eavesdropper[E] without any disturbance to their state.

In the meantime, several modifications, mathematical proofs & enhancements emerged in the works for QKD protocols. Certainly, several researchers used quantum simulations in order to study various aspects of various QKD protocols. Authenticity is one of the crucial issues between the participants intricated in QKD. As a result, the verification stances of QKD had been thoroughly examined.

The researchers in [17] thought on lowering the price of unconditionally-secure authentication. They proposed two major authentication modes, "partial" authentication and "complete" authentication modes, with regard to the stages that make up each QKD session. It was discovered that the complete mode, albeit having the highest level of security, can have an effect on the QKD system's efficacy.

Other earlier studies looked into how QKD (or quantum cryptography protocols generally) might be integrated with existing security and networking standards and protocols. A proposal that integrates QKD into IPSec, for instance, was given some thought by certain researchers [18]. Additionally, a method described in [19] incorporates QKD into the *“802.11i”* security procedures for the dissemination of encoding keys. A modification to the popular SSL/TLS web protocol that greatly simplifies the inclusion of QKD was introduced in [20]. Additionally, [21] advocated merging QKD to improve the transmission reassurance of a control system.

She consequently suggested a few enhancements pertaining to safety measures and verifiability concerns. These enhancements are anticipated to be mostly accomplished by adding some quantum cryptography techniques to the original protocol, such as QKD, quantum bit commitment, and multi-party computation [2].

However, QKD is the only quantum cryptography protocol that has been proven to be secure to date.

Table 4 contrasts our research with earlier studies.

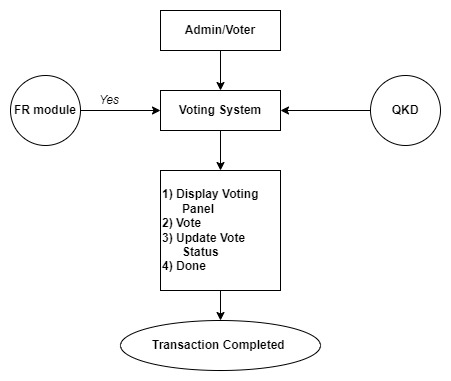
Former efforts to design procedures for blockchain-based e-voting systems developed schemes for cryptocurrencies. Cruz and Kaji [22] suggested a procedure for blockchain based e-voting and explored some of the aspects of security for electronic voting in response to recent developments [22–23]. Votes are tallied by adding up each candidate's tokens in the Bitcoin blockchain under the End-to-End (E2E) voting system that Bistarelli and others [23] proposed.

Using Bitcoin, Zhao and Chan [24] created a framework that is comparable to those previously discussed [22,23]. However, due to the time- and resource-intensive nature of the Bitcoin consensus mechanism, all three of the aforementioned protocols have limitations in terms of the scalability of the blockchain-based e-voting system [23].

A voting mechanism for electronic ballots based on the Ethereum blockchain, was introduced by other researchers [25–28]. All of these studies centred on exploiting the security characteristics provided by the Ethereum blockchain contract to increase the security of the electronic voting system. These studies, however, made no mention of the blockchain's scalability or performance. The efficiency and scalability issues for blockchain-based electronic voting systems were thoroughly discussed [29,30].

Khan [29] specifically experimented with permissionless and permissioned blockchain settings grounded on a variety of circumstances, such as the quantity of production rate, block, block size, transaction speed, voters, and. Zhang and others [30] asserted have demonstrated the shortcomings of blockchain-based electronic voting schemes. They do not, though, explicitly mention important security characteristics like uniqueness and ballot reception in their document.

1. **METHODOLOGY**

****

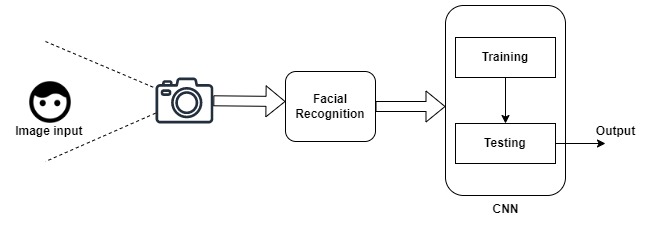
**Fig 6: System Architecture**

In order to achieve a fully secure voting system, the methodology that has been used is integration of Blockchain with Quantum Key Distribution and Convolutional Neural Network. Blockchain is being used to have a decentralized e-voting system in which the authority is not given to any third party means there is no centralization. Also, Blockchain is used to prevent the e-voting system from SQL Injection Quantum Key Distribution is being used to protect the e-voting system from Quantum Attacks.

Fig 6. depicts the system architecture of proposed methodology in order to create e-voting system.

Below is a high-level overview of the real-time face recognition process used in the registration process of the users on the proposed system.

**USER REGISTRATION PROCESS USING FACIAL RECOGNITION MODULE:**

****

**Fig 7: High-level overview of the real-time facial recognition process**

The data collected is pushed to the CNN architecture. This architecture comprises of fully connected layers that are followed by repetitions of a stack of numerous convolutional layers and a pooling layer.

**Eigen Face Algorithm:**

* For face recognition and image classification this algorithm follows the appearance-based approach. Variations in a pool of facial imageries are acquired by using this algorithm. The specific descriptions of individual faces are then encoded using the gathered data. The collection of facial photos is then generally compared to the encoded photographs.

Steps:

* Get the training set, use PCA projections & calculate Eigen Faces which describe Eigenspace.
* In “Principal Component Analysis (PCA)” method, cluster is formed in image space by face space.
* Calculate weight when a novel face is encountered.
* Verify if novel image matches image given in input faces.
* If it matches, categorize the weight pattern as matched/unmatched or known/unknown.
* In order to avoid fake votes during the voting process, the facial verification system is significantly worthwhile in recognizing fraudulent electorates or fake bots.

The facial photos will be divided up using this process, which is grounded on *information* *theory*. Then, a small group of distinctive features in images known as "Eigen faces" develop. This is nothing more than the main structural elements of the training set's face photos. The Eigen face method is among the most effective and straightforward techniques for face identification.

The distance between pairs of photographs is measured in the Eigen face approach. It is a recognised face if the space is at a lesser extent than a specified threshold value; else, it is an unidentified face. There are two sets of picture blocks in the algorithm: a training set image block and a test set image block. The trained image, or the Eigen face of the image in the database, is first obtained in the training set image block.

The weight W1 is then determined using the training set and the Eigen face. *Img* I is the unknown input image, which is simply the captured image, in the testing set image block. The input image and then Eigen’s face are used to determine the weight We 2. By calculating the average distance between We 1 and We 2, the worth of D is determined. The face is recognised if the D value is smaller than 0. The X and W2 values of the input image are then saved. The face is not recognised if the D value is higher than 0.

Step 1: Data preparation: The training set's faces have been removed for processing.

Step 2: Subtracting the mean: In this step, the average matrix must first be calculated before being deducted from the novel appearances. The outcome of it is then saved in a variable.

Step 3: Finding the covariance matrix is step three.

Step 4: Finding the covariance matrix's eigenvalues and eigenvectors Equation 2's covariance matrix is N2\*N2 dimensional, meaning it contains N2 eigenfaces and N2 eigenvalues. For instance, if we use 256\*256 photos, we must compute a total of 65,536\*65,536 matrices before determining a total of 65,536 Eigen faces. This makes it extremely difficult and impossible. Therefore, to characterise a limited group of vectors in a vast dimensional space, we utilise PCA.

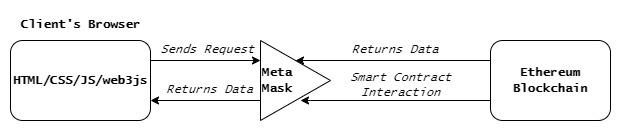
PCA provides an appropriate representation for calculating Eigen faces.

1. Determine the regular appearance: v.
2. Compile the variance amid the imageries(training) and the typical appearance in matrix A (M by N), where M denotes the number of pixels and N denotes the number of images.
3. The Eigen faces are given by the eigenvectors of the covariance matrix C (M by M). Because M is generally large, this procedure would take a long period.
4. AA \* Calculation of C Eigenvectors If the total no. of data points is less than (N\*N).

**Blockchain Implementation tools:**

* **Truffle Framework:** This framework is used to build a decentralized illustration on the Ethereum Blockchain. It helps in testing and deploying our smart contracts to the blockchain. With a help of Solidity and C++ programming language, it allows writing smart contracts by providing a suite of tools. Truffle provides a better environment that is entirely based on the Ethereum. Truffle is proficient in migrating and compiling contracts [*Ethereum*]. It is a worldwide sophistication enhancement tool that is experimenting with useful pipeline for blockchains and a framework based on the Ethereum Virtual Machine (EVM). A framework for deployment and migrations that is scriptable and extensible. - Network control for deploying to a variety of public and private networks. Truffle is a collaborating platform that allows for straight conclusion transmission.
* **Ganache:** It is used for live demonstration purposes. For testing purposes, 100 fake ether was preloaded with each account. It is a private blockchain dedicated to the development of instant Ethereum and Corda software. Used to build a personal Ethereum blockchain that resembles the real-world blockchain and allows you to execute commands and run tests. It is available in two varieties: UI (User Interface) and CLI (Command Line Interface).
* **Meta-mask:** In order to use the Ethereum blockchain, a special browser extension “Meta-mask” is used. We can interact with our smart contract by connecting our local Ethereum blockchain with an account. Meta-mask is an Ethereum Blockchain wallet that allows users to interact with it. Users can store and manage their account keys with meta-mask, as well as exchange Ethereum-based cryptocurrencies and tokens.

Using a web browser or the built-in browser app on a mobile device, users can connect to decentralised networks. Developers can start connections between a decentralised network and Meta-mask using the JavaScript plugin Web3js. For transactions between meta-mask and smart contracts, ethers are employed as the gas.



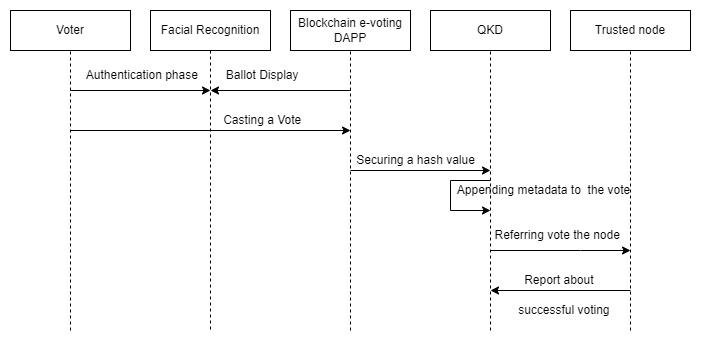
**Fig 8: Meta-mask Working**

Blockchain users may control their wallets thanks to Meta-mask. Users can utilise the wallet and complete transactions over the web by using the browser extension. A meta-mask that appears when a transaction is completed asks the user to confirm the activity. Users can manage their keys and accounts in various ways thanks to meta-mask. Figure 8 shows how Meta-Mask operates. User requests for data are sent from the client side to the Ethereum blockchain through a mediator named meta-mask. In response, the Ethereum blockchain and meta-smart mask's contracts interact, and meta-mask then sends data to the client side.

* **Ethereum**: Ethereum is a decentralised blockchain platform that allows smart contracts to be executed. Smart contracts are applications that operate without the risk of fraud, censorship, or third-party intervention. Also, Ethereum owns EVM which is designed to assist as an execution environment for Ethereum-based smart contracts.
* **Solidity:** Solidity is a contract-oriented programming language that is statically typed and designed to direct the Ethereum Virtual Machine (EVM). Its internals are heavily influenced by Python, JavaScript, and C++. Solidity can be used for purposes like multi-signature wallets and voting as well.
* **MongoDB:** A document-oriented, cross-platform database that delivers elevated performance availability and scalability.
* **Node.js:** Node.js is a server-side platform that never expects an API to return data.

1. **WORKFLOW**

**Sequence Diagram**:

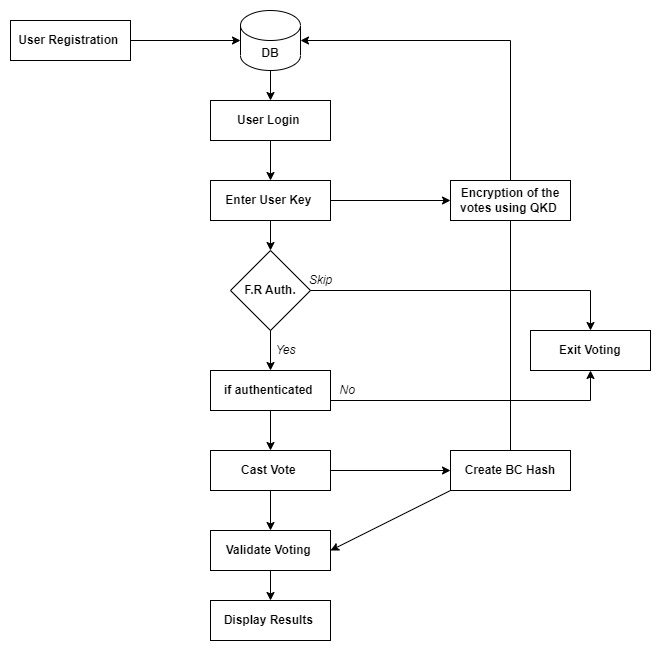


**Fig 9: Workflow of the proposed e-voting procedure metholdogy**

Sequence diagram for proposed solution is depicted in figure 8. Voter will first pass through the authentication phase which is done by using facial recognition. After proper login and authentication, the voter will redirect to ballot display page. Voter will choose candidate and cast a vote accordingly. After casting a vote, all the blockchain hashes of current transaction will get secured by using quantum key distribution. The main aim of quantum key distribution is securing a current hash value of novel transaction which is just get executed in decentralized application. There is one trusted hybrid node in which metadata of blocks are appended and votes are referred to the node. This will report about termination of one cycle by displaying successful voting on display page. Voters complete an application in the initial step of the registration procedure. The application will then begin its procedure in the next phase. The voter's face is first photographed using a camera or a webcam. The current database is then provided that image. Here, information on voters with images is taken from an existing database or centralised database. The system then compares the sent photographs to those that have already been recorded in the main database. The server transmits the (One-Time Pad )OTP QKD scheme to the voters' registered email address or mobile number after the picture has been effectively recognised or matched with the database's existing images. This results in creation of a new secure hash value that cannot be cracked even by quantum attacks. Once the voter has input the OTP, the database will once more verify it, and if it is accurate, the voter is a legitimate user.

The list of candidates will then emerge. The candidate's name appears on this list, and a button labelled "Vote" is available next to that name. Only after the voter presses that button shall the voting process be considered complete. Security is also required in this case to protect the system. For this, a session-level security mechanism is used. The client must first request a valid session id from the server. After client validation, the server will reply with the session id. Handshakes are used to communicate session id. The session id will be sent and validated for each request. When a client logs out of the system, the session id is destroyed. Additionally, the password is MD5-encoded, making it impossible for anyone to view the password content.

1. **PROPOSED WORK**

****

**Fig 10: Proposed Architecture**

* Setup: After generating the public and private (asymmetric pair of keys), the input parameters are decrypted and encrypted.
* Registration: Use of IDs in order to generate public or private key as output.
* Vote: Vote parameter is created by the electors and then signature and consequent text is computed.
* Validation: The server chooses the input as a vote and then performs further validation to verify the validity of the vote.
* Append: Cryptographic text is updated.
* Publish: Voting results are published widely.

**After the registration process using a facial recognition module,** each voter is coupled with a “private and a public key” (Asymmetric Key Pair). Senders send private keys to blockchain Peer-to-Peer network in order to sign messages. In this state, for a point of time these messages are stored and accumulated in a block. Receivers acquire this message recorded on the blockchain and distributed over the network.

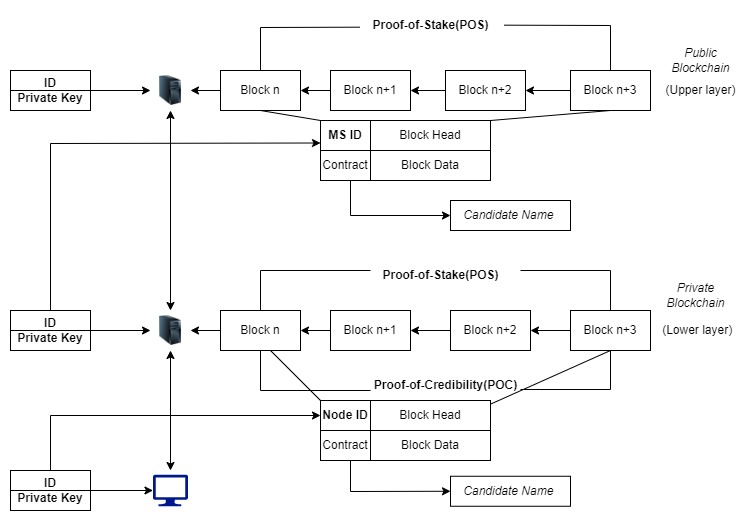
**System Execution in Phases:**

1. **The Registration Phase**
2. **The Removal Phase**
3. **The Voting Phase**
4. **The Vote Counting Phase**

**Blockchain Workflow:**

To perform core implementation, hybrid blockchain is being used. In Hybrid BC, a selected portion of the data in a blockchain can be made public while keeping the rest private. Usually verified by private network, but a transaction can also be distributed in the public network for verification. Hybrid Blockchain provides best security among all other types of blockchain. Hybrid BC works according to situation, it carries out advantages from all the types of Blockchains.

In every blockchain, the first block is always known as Genesis Block. Block(B) contains Nounce, Header, Data, and Metadata(M). Whereas, Header contains Block Number (BN), Current Hash Value (CH), and Previous Hash Value (PH). A Block number is simply an identity number of a block. Before proceeding toward the Current hash value and previous hash value, the Hash should be discussed. So, Hash is a unique value that makes every block unique. The hash value is made up by using 64 hexadecimal character values [0-9] [A-F]. Hashing is done using SHA 256 algorithm (Secure Hash Algorithm). On the basis of [Block No., Nounce, Data and Previous Hash value, Hash value for each block is generated. Suppose, Block 1 has PH *“000acdehk32”* and CH *“453ack9dpe0*” so, the PH of Block 2 would be *“453ack9dpe0”*. Now, is there any way to change the value of hash? Yes, by changing the value of Nounce we can change the value of hash. Nounce stands for (No. used only once). We can change the nounce’s value in order to get the targeted value of Hash. We check if the hash value is available or not by again & again changing the value of Nounce. Metadata(M) contains information regarding transaction details, no. of nodes, and no. of participants.

****

**Fig 11: Hybrid Blockchain Consensus Model for e-voting**

This model combines Proof of Credibility (PoC) and Proof of Stake (PoS).

Proof of Work (PoW) has a negative influence on transaction throughput and latency, which has a direct impact on scalability and performance. Proof of Stake (PoS) has been employed as a result to get near the problems with PoW. In the PoS consensus method, a group of validators take turns proposing and voting on the next block, and the weight(w) of the vote is determined by the value of stake of tokens. PoW is not a computationally difficult system, though. In PoW, multiple nodes & multiple miners compete with each other in order to achieve targeted hash value or can say Complex Mathematical Puzzle. After achieving targeted value, other nodes verify that hash value & post verification that hash is assigned to a block which is (Block Hash). This Block hash represents the work done by the miner which is known as PoW(Proof of Work). Drawback of PoW is that it is expensive.

PoS, a replacement for PoW that is more effective and scalable than PoW, is now available. In PoS, a group of nodes are set up at a stake in order to be candidates, validators, and transaction verifiers. In return, they receive a stake amount (S).

Utilizing PoW or PoS can lessen the possibility of a plot or attack.

The hybrid consensus model has drawn a lot of interest as a potential solution to the problems with the PoS & PoW consensus approach. PoC is used to improve block generation & increase the capacity of trusted nodes.

The functions of the e-voting system's components are discussed, along with the design of the suggested scheme architecture:

Server management/Manager Server (MS): The purpose of (MS) is to supply nodes with certificates, store node information in the lower blockchain network, and broadcast it to the higher blockchain network. This enables node authentication and contains user credentials for system login.

Blockchain network: The proposed e-voting blockchain network is made up of multiple blockchains that work in tandem. This structure allows for parallel execution, which improves overall system performance and scalability. Lower-chains (private-chains) serve to store node information and the voter identity register, as each node in the private-chains has a local blockchain containing privacy-sensitive data. After certain voters successfully agree on the transactions, a process known as proof-of-stake consensus, and concurrently process transactions, the upper-chain (public blockchain such as Ethereum) serves to store separate blockchain states across all voters. The transactions recorded in the upper-chain (public blockchain) are trusted and irreversible.

Users (voters): Users are voters who also serve on the election committee; they can authenticate and access their wallets using their identity ID. Voters are given a digital token that allows them to cast their ballot. As a result, smart contracts are deployed in the blockchain's top layer (Ethereum blockchain).

Smart contract (blockchain contract): In the anticipated decentralised scheme, smart contracts are self-executing snippets of code. The functions embedded in smart contracts define the contract agreements that allow transactions in the blockchain network's top layer to be tracked.

Every node in the blockchain network can run the smart contract independently to reach a consensus, resulting in the development of a flexible cryptosystem for e-voting systems.

Figure 9 illustrates how we envisioned the election organising committee using hybrid two-tier consensus approach as a public blockchain, possibly with a separate node. This implies that the government does not have to put all of its faith in a public blockchain network to decide the future of the entire democratic process. Since the technology is still relatively new and so untapped. As an alternative, we are adding an additional layer of external validation to the process, which will be carried out by a large, unbiased community of validators who are unconnected to the election itself. As a result, the process control remains in the desired state while the observation remains constant & external. Additionally, in that it can be constructed from the ground up, our system offers a holistic approach. The entire procedure is validated using a hybrid consensus model. Following are the settings made available by the hybrid blockchain to facilitate the multi-party confidence consensus system:

* Transactions from an upper-chain unamed authenticated node can be verified by the miners (voters) (Ethereum blockchain).
* The PoS consensus mechanism is the only one that the upper-chain network can use to reach a consensus on the transactions. A digital, blockchain-based electronic voting system may also be a solution when considering future elections. In this case, voters would be identified by biometrics and given a digital token that would enable them to cast a ballot while maintaining their anonymity. Private blockchains on the PSC-Bchain Network with a hybrid consensus method (PoS and PoC) would process these transactions, and concerned parties might include political parties, block producers, observers, & governmental organizations. The electorates might transfer the saved transactions in order to replicate to the private blockchain network and momentarily store them in virtual blocks.

This setting discusses the following additional competencies of the hybrid blockchain:

* In real-time, the voter machine directs delicate communications to the public blockchain & protects the communications in the implicit blocks.
* The voter device could likewise accumulate & save recorded delicate communications in implicit blocks.
* A voter transmits its implicit blocks to a private blockchain for replication of data. By enforcing, carrying out, and governing the blockchain network, validators of blocks would verify that whether the block creators are not engaging in fraud.

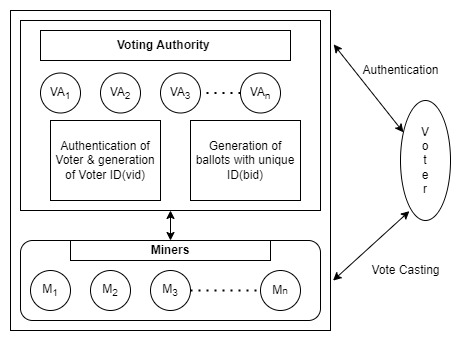
**Process for validating the reliability of a node:**

There is a specific process for validating the reliability of a node. Basically, a node includes Private Key and ID for nodes, where the private key is used as a flag for node reliability authentication & asymmetric encryption whereas to identify one another, a specific node identifier “the ID” is used. The Manager Server (MS) is in charge of each node management that obtains and issues the private key. In (MS), ‘*Private* *Key’*, ‘*ID’,* and ‘*block* *contents’* are considered as surplus information in the MS. Amongst these, the sole MS identifier is the ID. Manager Server is similarly a sort of (system node), it is the same as the other nodes exclusive of its storage capacity & computational ability. Consequently, the node IDs and MS should have the equivalent description & MS must have an equivalent attribute ID as other nodes.

As per blockchain technology, every block includes block data and blockhead. The edifice of the block log includes MS, Node ID, flag for deleting or adding data of the entities handled by the Merkle root, Timestamp, Contract, MS, and Transaction count. The objective of the block header is to accumulate existing block info. with hash values & block numbers. The Public Key must be formed from the Private Key in one block record.

**QKD Workflow**

In this workflow, permission is granted to every user in order to read the information in the blockchain, but only specific entities are permissible in order to become miners.



**Fig 12: QKD Workflow**

**Voting Authority (VA)**: Voting authority (𝑉𝐴) has the inclusive duty to operate the election process. The key responsibility of the VA is to take consideration of the voter’s registration by verifying their credentials and create a set of voters (V), who are fit to cast their votes in the elections. Voter set (V) provides a random unique id to every voter, and to begin the voting process this information is recorded by VA for verification. For adding votes into the blockchain 𝑉𝐴 have the accountability to appoint the authorities to the miners. Likewise For the intent of authentication of the votes, 𝑉𝐴 will interact with the miners. Lastly, after achieving the inspection of the votes, 𝑉𝐴 will announce the outcome of the elections.

**Voters:** Voters are vital to any election process. Before the casting of votes, every voter has to get registered with 𝑉𝐴 in order to become a part of the eligible set of voters (V). 𝑉𝐴 will set rules for voters in order to cast vote by broadcasting their votes to the miners.

**Miners(m):** Miners (m) have the inclusive duty to maintain the blockchain. With the help of the VA, miners will verify the votes which they will collect from the voters. By using “DPoS” protocol vote is lastly appended to the blockchain after verification by miners(m). In this protocol, VA will select ′r ′ miners by considering the particular depictions from the voter set(V). All the miners will add the block to the voting blockchain after validating and verifying the votes. If the miner is not available at any point of time, mining power will be shifted to the subsequent accessible miner. Also, when the block metadata is validated by the miners in the majority, only then a block will be taken into consideration as a valid block.

Cryptography helps in transmitting secret messages, and “this secret message is secure until the key for encoding that message is secure”.

The challenge of secure communication can be ruminated as the difficulty for secure distribution of the secret keys.

By resolving this challenge, public key cryptography postulated a uprising in this domain, in which the message is encoded and decoded by the public key & private key of the receiver respectively. Acquiring the private key is computationally unattainable for any hacker or external entity. Public key cryptography is thus computationally secure.

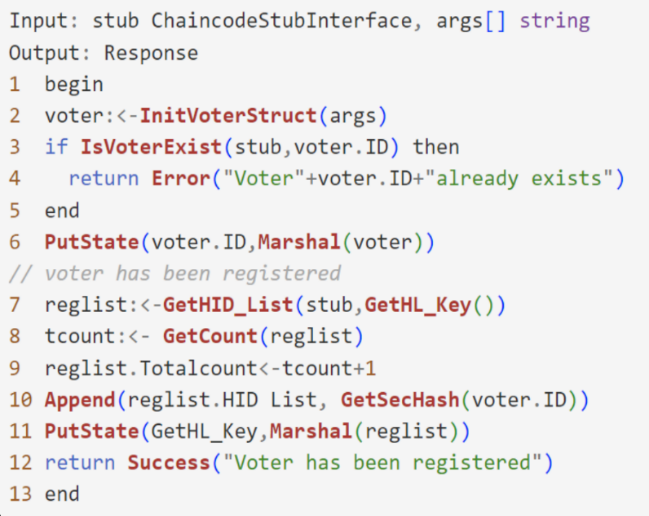
Though, Shor demonstrated in 1994 that the RSA algorithm can be broken if a quantum computer is used. This sparked renewed interest in developing unconditionally secure key distribution. The keys generated by quantum cryptography are unconditionally secure in comparison to the computational security provided by classical cryptography. This field of unconditionally secure quantum cryptography began in 1984 with Bennett and Brassard's first QKD protocol, now known as the BB84 protocol.

1. **IMPLEMENTATION**

Implementation is divided into two phases: One is Initial Implementation and the Second is Core Implementation. This is the initial implementation which comprises of Registration Phase, Vote Counting Phase & Removal Phase.

1. **REGISTRATION PHASE:**

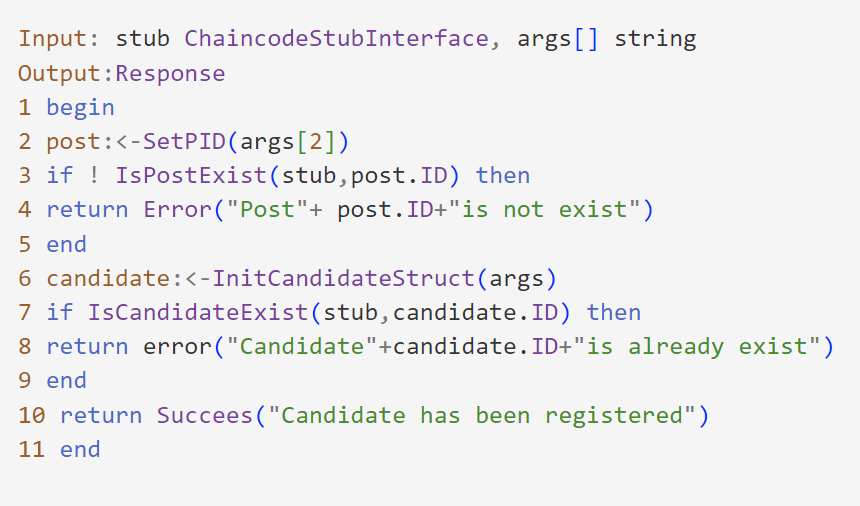
Algorithm 1: Registration contract for voter



Algorithm 1 defines the voter registration contract that takes i/p as an argument and stub and returns a response as an o/p. Steps involved in this contract are:

1. The *“Marshal(voter)”* value is stored by using *“PutState”* function. Using *“Marshal()”* , the object named voter is converted to JSON format.
2. To return the reg. list *“GetHID\_List”* function is used. Then, to return the no. of entries in the list *“GetCount”* function is used. After this, *“tcount”* is updated into the *“reglist”* after getting incremented by one.
3. *“GetSecHash()”* hashes the newly reg. voter id. And further appended it into the *“reglist”.*
4. *“GetHL\_Key”* function generates the key which is then stored with *“reglist”*.
5. Lastly, it gives a response with the voter registration message.

Algorithm 2: Registration contract for candidate



Algorithm 2 defines the candidate registration contract that takes arguments & stubs as an i/p and returns a response as an o/p. It checks whether the candidate is registered and returns an error if they are already registered. Steps involved in this contract are:

1. To store the *“Marshal(post)”* value the *“PutState”* function is used.
2. To return the candidate registration list *“GetCandList”* function is used. Then, to return the number of entries in the list the *“GetCount”* function is used. After this, *“tcount”* is updated into the *“creglist”* after getting incremented by one. In *“creglist”*, newly registered candidate is appended.
3. *“GetCL\_Key”* function generates the key which is then stored with *“creglist”*.
4. Lastly, it gives a response with the candidate registration message.
5. **REMOVAL PHASE:**

Algorithm 3: Withdrawal Cont. for removing voter



Algo. 3 defines the voter withdrawal contract which takes input as arguments and stubs and returns a response as an o/p. Stages intricated in this algorithm are:

1. Voter id is initialized with *“[cid]”* in order to get the voter object with an input *“args[0]”.*
2. To remove the entry of the voter “DelState” function is used.
3. To return the registration list *“GetHID\_List”* function is used.
4. *“GetHID\_List”* is to check the existence of the voter in the *“reglist”* and return an error in case the voter does not exist.
5. *“GetHL\_Key”* function generates the key which is then stored with *“reglist”*.
6. Lastly, it gives a response with the voter withdrawal message.

Algorithm 4: Contract for removing the candidate



Algorithm 4 states the contract for removing the candidate in which arguments and stub are taken as input. The steps intricated in this contract are:

1. Candidate id is initialized as *“[cid]”* in order to get the candidate object with an input *“args[0]”*.
2. To remove the entry of the candidate *“DelState”* function is used.
3. To return the registration list *“GetCandList”* function is used.
4. To check the existence of the candidate in the *“reglist GetHID\_List”* function is used.
5. *“GetCL\_Key”* function generates the key which is then stored with *“reglist”*.
6. Lastly, it gives a response with the voter withdrawal message.

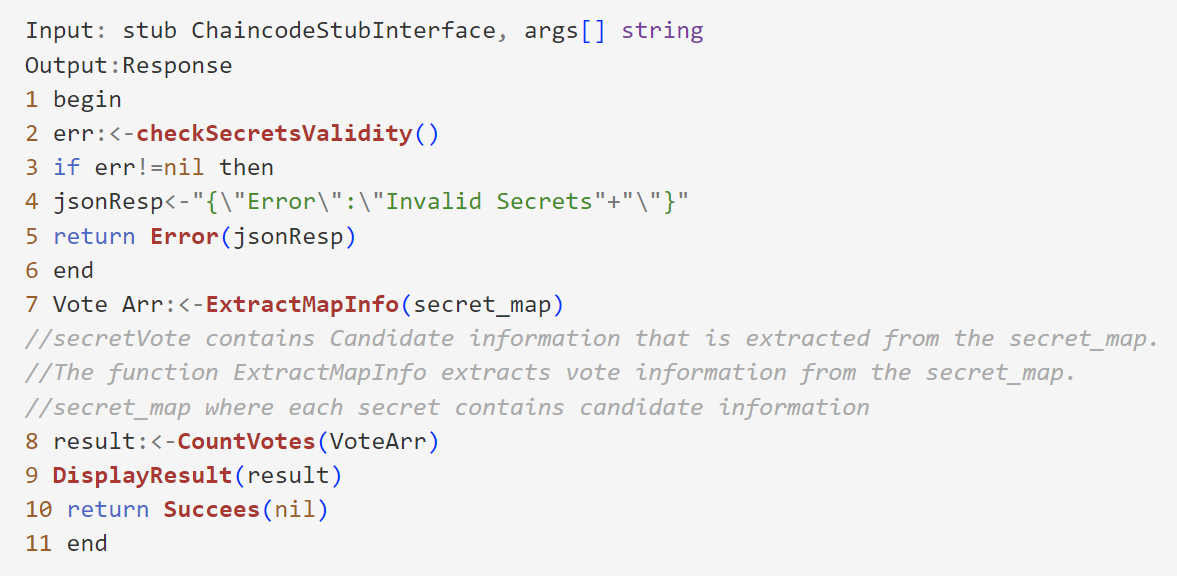
Algorithm 5: Contract for disable voter



Algorithm 5 states the contract to disable a voter, that takes input as arguments and stub & further gives voter object as a response. Once the voter is disabled, he/she would not be able to vote in the voting process again. The steps intricated in this contract are:

1. *“[vid]VoterId”* is initialized with an input *“args[0]”.*
2. Corresponding to the vid it gets the voter object.
3. If tokens<=0, *“Voter.tokens”* returns the voter object.
4. Else, it will return with an error message *“errors.New”*.
5. **VOTE COUNTING PHASE:**

Algorithm 6: Contract for vote counting

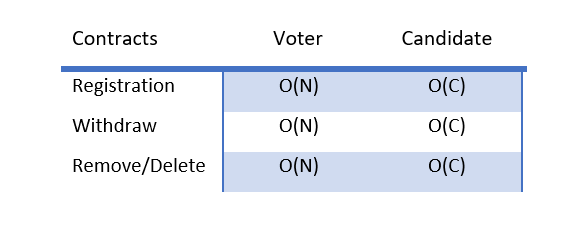


Algorithm 6 defines counting contract, which takes input as arguments and stub and proceeds with a conforming response. Steps involved in this contract are:

1. In case of misrepresentation identified, function named *“checkSecretsValidity()”* gives and error as response. It proceeds further if there is no error reported.
2. In order to accumulate row-wise secrets, it generates a *“secret\_map”.*
3. By calling Combine function, candidate information is extracted.
4. Information like *“VotesReceived”*, *“CID”* is extracted by function named *“ExtractMapsInfo”* from *“secret\_map”.*
5. *“CountVotes”* function is then called in order to generate the whole voting fallouts.
6. Lastly, a successful response is returned.

1. **RESULT & DISCUSSION**

**Table 6: Time Complexities of Smart Contract Algorithms:**

****

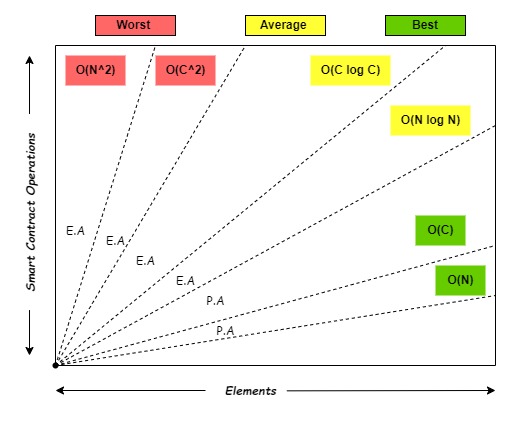
***There are “C” total number of voters and “N” voters.***

**The operational intricacies of the proposed smart contracts are depicted in Table 6.**

In, afore-mentioned algorithms, k posts, n voters, t candidates, m authorities, involved in an election process.

At voter registration, all formerly recorded entries are required to get access for a particular registration operation in order to authenticate the presence of the specific individual. So, for voter registration the time required is O(n). Also, time required for the authority is O(m), time required for the registration of candidate is O(t) & time required for the post is “O(k)”. Furthermore, before execution both Remove and Disable contracts need the precise list in order to check the presence of record. Thus, these processes take “O(m)” time in case of authority, “O(t)” time in case of candidate, “O(n)” time in case of voter, and “O(k)” time in case of post.

***E.A stands for Existing Approaches & P.A stands for Proposed Approach.***



**Fig 12: Smart Contracts Complexity Graph**

Time complexity for Voter Regis. contract is O(N) and Candidate Registration contract is O(C). Time complexity of Voter Withdraw and Candidate Withdraw are O(N) & O(C) respectively. Time complexity for Voter-Candidate Remove/Delete contract is O(N) & O(C) respectively. This concludes that proposed smart contracts algorithms are best complexities in terms of time and space. However, smart contract algorithms for existing approaches are somewhere lies at O(N^2) and O(N log N) which are horrible and worst complexities in terms of time and space. Fig 12 depicts smart contracts complexity graph by dividing approaches as existing and proposed. Algorithms used for creating smart contracts in previous approaches are not optimized as compare to proposed approach because time complexity of existing approaches lies somewhere between worst O(N^2) to average O(N log N). This shows that our proposed algorithms are optimized and ready to be executed.

1. **CONCLUSION & FUTURE DIRECTION**

Regardless of being a reliable technology, blockchain occupies Quantum key distribution order to counter the risk of quantum attacks. The proposed e-voting system is decentralized and does not need to rely on a third party or centralized authority. Authorized and eligible voters can vote using their e-devices connected to the internet. In this system, each voter can validate individual voting actions. This system defends the voting process from fraudulent activities and attacks. Time complexities of smart contracts states that these contracts are best suited in terms of time and space complexities. The proposed system is divided into two implementations: initial implementation and core implementation. Initial implementation is completed till now. Core implementation includes syncing all the technologies on practical level. By performing initial implementation, choices of algorithms and contracts are well-defined to proceed to the next stage of implementation. The study aims to build a “blockchain-cnn-qkd” grounded elective system which accomplishes all necessities of an voting process by upholding security, privacy and a scale of decentralization. The proposed work secures the system by reducing the chances of attacks like DOS, SQL Injection, Stale block, Quantum attacks, and Blockchain forks. A blockchain fork occurs when a single chain of blocks gets fragmented into two or more chains that are valid. There are mainly three types of forks: Soft fork, Hard fork, and Temporary fork. Quantum attacks can easily Blocks that are no longer part of the current best blockchain because they are overridden by a longer chain are Stale blocks. Quantum attacks can manipulate the transactional information between the blocks which will lead further leads to fraudulent activities. Integration of Blockchain, CNN & Quantum Key Distribution leads to a triple-layered security scheme. In blockchain all blocks are linked cryptographically using quantum key distribution protocol. Sooner, for applied high security communications, QKD will certainly become a standard.

1. **REFERENCES**

[1] M. Gibbins, S. McCracken, S.E. Salterio, The auditor’s strategy selection for negotiation with management: flexibility of initial accounting position and nature of the relationship, Organ. Soc. Accounting (2010).

[2] Sabino M., 2016, “Quantum Cryptography applied to Electronic-Voting Protocols,” M.Sc. Thesis/ Mathematics and Applications, IST, Tecnico, Lisboa.

[3] Ronald L Rivest, Adi Shamir, and Leonard Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (1978), 120–126.

[4] Peter W Shor. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science. Ieee, 124–134

[5] Charles H Bennett and G Brassard. 1984. Quantum cryptography: Public key distribution and coin tossing. In International Conference on Computer System and Signal Processing, IEEE, 1984. 175–179.

[6] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. 2002. Quantum cryptography. Reviews of Modern Physics 74, 1 (2002), 145–195

[7] Akshata Shenoy-Hejamadi, Anirban Pathak, and Srikanth Radhakrishna. 2017. Quantum cryptography: Key distribution and beyond. Quanta 6, 1 (2017), 1–47

[8] Artur K Ekert. 1991. Quantum cryptography based on Bell’s theorem. Physical Review Letters 67, 6 (1991), 661–663.

[9] Umesh Vazirani and Thomas Vidick. 2019. Fully device independent quantum key distribution. Commun. ACM 62, 4 (2019), 133–133

[10] Oleksandr Korchenko, Yevhen Vasiliu, and Sergiy Gnatyuk. 2010. Modern quantum technologies of information security against cyber-terrorist attacks. Aviation 14, 2 (2010), 58–69

[11] Feihu Xu, Xiongfeng Ma, Qiang Zhang, Hoi-Kwong Lo, and Jian-Wei Pan. 2020. Secure quantum key distribution with realistic devices. Reviews of Modern Physics 92, 2 (2020), 025002.

[12] Martin E Hellman. 2002. An overview of public key cryptography. IEEE Communications Magazine 40, 5 (2002), 42–49.

[13] Al-Janabi S., 1999, "Quantum Cryptographic Key Distribution in Optical Communication Networks," Ph.D. Thesis, College of Engineering, Al-Nahrain University, Baghdad.

[14] Al-Janabi S., and Jasim O., 2011, "Reducing the Authentication Cost in Quantum Cryptography," The 12th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet2011), UK, 363-368.

[15] Bennett C., and Brassard G., 1984, “Quantum cryptography: public key distribution and coin tossing,” International Conference on Computers, Systems and Signal Processing, India, 175-179.

[16] Bennett C., Bessette F., Brassard G., Salvail L., and Smolin J., 1991, “Experimental quantum cryptography,” J. of Cryptology, 5, 3-28.

[17] Al-Janabi S., and Jasim O., 2011, "Reducing the Authentication Cost in Quantum Cryptography," The 12th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet2011), UK, 363-368.

[18] Sfaxi M., Ghernaouti-Helie S., and Ribordy G., 2005, "Using quantum key distribution within IPSec to secure MAN communications", Proceedings of the IFIP-MAN 2005 Conference on Metropolitan Area Networks, Vietnam, 255-260.

[19] Nguyen T., Sfaxi M., and Ghernaouti-Helie S., 2006, "802.11i encryption key distribution using quantum cryptography," Journal of Networks, 1(5), 33-40.

[20] Al-Janabi S., 2008, “A novel extension of SSL/TLS based on quantum key distribution”, Proceedings of the International Conference on Computer and Communication Engineering (ICCCE08), Malaysia, I, 919-922.

[21] Zhou J., Lu L., Lei Y., and Chen X., 2014, "Research on improving security of protection for power system secondary system by quantum key technology," Power Syst. Technol., 38, 1518-1522.

[22] J. P. Cruz and Y. Kaji, E-voting system based on the bitcoin protocol and blind signatures, IPSJ Trans. Math. Model. Appl. 10 (2017), 14–22.

[23] S. Bistarelli et al., An end-to-end voting-system based on bitcoin, in Proc. Symp. Appl. Comput. (Marrakech, Morocco), 2017, pp. 1836–1841.

[24] Z. Zhao and T.-H. Hubert Chan, How to vote privately using bitcoin, in Proc. Int. Conf. Inf. Commun.Security (Beijing, China), 2015, pp. 82–96.

[25] E. Yavuz et al., Towards secure e-voting using ethereum blockchain, in Proc. Int. Symp. Digital Forensic Security (Antalya, Turkey), 2018, pp. 1–7.

[26] G. G. Dagher et al., Secure voting system using ethereum’s blockchain, in Proc. Int. Conf. Inf. Syst, Security Privacy (Madeira, Portugal), 2018, pp. 96–107.

[27] S. Shukla et al., Online voting application using ethereum blockchain, in Proc. Int. Conf. Advances Comput., Commun. Inform. (Bangalore, India), 2018, pp. 873–880.

[28] S. Pareek et al., E-voting using ethereum blockchain, IJRTI. 3 (2018), 2456–3315.

[29] K. M. Khan, J. Arshad, and M. M. Khan, Investigating performance constraints for blockchain based secure e-voting system, Future Gener. Comput. Syst. 105 (2020), 13–26.

[30] S. Zhang, L. Wang, and H. Xiong, Chaintegrity: Blockchainenabled large-scale e-voting system with robustness and universal verifiability, Int. J. Inf. Secur. 19 (2019), 1–19.

[31] Liu, Yi, and Qi Wang. ”An E-voting Protocol Based on Blockchain.” IACR Cryptology ePrint Archive 2017 (2017)

[32] Ayed, Ahmed Ben. ”A conceptual secure blockchainbased electronic voting system.” International Journal of Network Security & Its Appli- cations 9, no. 3 (2017)

[33] Ganji, Raghavendra, and B. N. Yatish. ”ELECTRONIC VOTING SYS- TEM USING BLOCKCHAIN.” (2018).

[34] Yu, Bin, Joseph K. Liu, Amin Sakzad, Surya Nepal, Ron Steinfeld, Paul Rimba, and Man Ho Au.” Platform-independent secure blockchain- based voting system.” In International Conference on Information Secu- rity, pp. 369-386. Springer, Cham, 2018.

[35] Yi, Haibo. ”Securing e-voting based on blockchain in P2P network.” EURASIP Journal on Wireless Communications and Networking 2019, no. 1 (2019): 137