Blockchain Based E-Voting System With Homomorphic Encryption and Threshold Signature

by

Mushfique Nasir Probor
19301227
Mursalin Ahmed
19301228
Sharika Bintey Kabir
19101135
Md. Muhtasim Fuad
19301236
Tasnim Bushra
19301060

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
May 2023

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mushfique Nasir Probor
19301227

Ahmed

Mursalin Ahmed
19301228

Sharika

Sharika Bintey Kabir
19101135

Fuad

Md. Muhtasim Fuad
19301236

Bushra

Tasnim Bushra
19301060

Approval

The thesis/project titled "Blockchain Based E-Voting System With Homomorphic Encryption and Threshold Signature" submitted by

1. Mushfique Nasir Probor (19301227)
2. Mursalin Ahmed (19301228)
3. Sharika Bintey Kabir (19101135)
4. Md. Muhtasim Fuad (19301236)
5. Tasnim Bushra (19301060)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 22, 2023.

Examining Committee:

Supervisor:
(Member)

Iqbal

Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Security issues have been at large as mankind is becoming more comfortable with technologies. The rapid growth of technology has not given us enough time to understand technologies in whole. Thus this revolution in technological industries has opened the door for more chances of information breach, malicious attack and technical vulnerability. A smart voting system is very important for running smart cities. But the conventional voting system has problems on its own, like vote manipulation, forging outcomes, or personal threats, etc. But incorporating different technologies to move away from normal voting systems to e-voting systems makes us weak to vulnerabilities mentioned above. In this paper, we are proposing an ethereum based electoral voting system with homomorphic encryption and threshold signature. Firstly, we deployed our system on the Ethereum blockchain network which gives us an open view to contracts. Secondly, homomorphically encrypted votes from the voters lets us encrypt the votes and work on the encrypted data. So, the votes are never disclosed while counting. Finally, threshold signature is used to ensure multi layers of security by engaging multiple signers to build a single signature which will be used to retrieve the desired result maintaining all protocols.

Keywords: Blockchain; Ethereum; Threshold Signature; Homomorphic Encryption; Paillier Algorithm; E-Voting System.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our advisor Muhammad Iqbal Hossain, PhD sir for his kind support and advice in our work. He helped us whenever we needed help.

Thirdly, to Muhammad Nur Yanhaona, PhD for his invaluable guidance throughout the completion of this thesis.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

λ	Lambda
μ	Mu
φ	Phi
pK	Public Key
PoS	Proof of Stake
PoW	Proof of Work
sK	Secret Key / Private key

Chapter 1

Introduction

1.1 Overview

In the modern era that is gradually forging ahead toward the zenith of technological advancement and creating smart cities, one of the most cardinal aspects hindering such progress is the need for an unprejudiced and transparent voting system. The voting system allows each and every adult citizen of a sovereign country to partake in the process of electing their preferred representative. Such a fundamental approach must be strengthened, decentralized, and impartial. Voting ballots are supposed to be such that only the voters have the authority to mark them in private booths without anyone knowing who they are voting for. However, the conventional voting system is centralized and does not provide the security that is required to warrant a proper and fair election. Given the technological progress that countries across the globe have achieved, corrupting the centralized traditional voting process by adopting illegal means such as- casting fake votes, manipulating the voters to vote multiple times , parties of the electoral candidates threatening and offering money to the voting centers, controlling the polling booths and ballots etc, have become much easier. To address such issues with the existing election process, the electronic voting system was first introduced by David Shaum in the early 1980s, which employed public key cryptography to protect the anonymity of the voters [21]. The blind signature theorem was used to ensure that the voters and the ballots were not associated [21]. The first country to support the electronic voting system was Estonia for its national elections [25]. But it had several issues with security and verifiability as it lacked a proper approach to solving the problems. Again in 2011, Norway employed an e- voting system for its council elections, which was much like that of the Estonian voting system [21]. However, one of the vital problems with their system was that the votes were at risk of going public and the identities of the voters being exposed due to cyberattacks. As a repercussion, the urgency for a much more guaranteed, decentralized system came into the picture. In order to eradicate these problems, Satoshi Nakamoto, in the year 2008, came up with the exquisite idea of a blockchain based e-voting system [35]. Blockchain, since its inception, has been well received by the modern era, because of its decentralized notion. Its distributed structure offers high security and flexibility due to having a decentralized nature. It can provide the legitimacy and transparency that are much needed in a fair election process. Some of the fundamental properties of blockchain, such as - decentralized system, higher security, verifiability, and availability, distributed ledgers, strict transparency, etc., meet the criteria to avoid illegal approaches during the election process. Especially in developing countries like Bangladesh, where faulty elections and corruption have been on the run for years, the use of this system can bring about revolutionary changes in the political as well as economic sectors by corroborating legal and transparent elections and money transactions [40]. Blockchain technology is no longer an implausible concept, as many countries have already adopted this system into their voting processes. Sierra Leone was the first country to employ a blockchain based electronic voting system in 2019. Countries like Russia and the United States have also adopted the use of blockchain to some extent in their election processes. The first application of the blockchain was Bitcoin, which supports cryptographic transactions [21]. Nevertheless, it was regarded as onerous and slow because of having a strict authentication procedure. The electronic voting system has been made much more secure and affordable by applying the Ethereum blockchain providing smart contracts which has yielded greater authenticity and transparency. It helps process the information provided by the voters [30]. The Ethereum blockchain is one of the most popular systems due to it being consistent and ability to provide smart contracts [30]. This blockchain would help preserve the vote and ballot records. But it is not completely anonymous, as the miners can access the votes and ballots. Anonymity, security, and verifiability are very important in a secure electronic voting system. Hence, anonymity and security can be achieved by Paillier's homomorphic encryption. Paillier in 1999 proposed Paillier's cryptosystem, which is additive homomorphic encryption [23]. It has gained popularity due to maintaining the discretion of the voters' information. On the other hand, threshold signatures and trustee shares are used to deploy and decrypt the votes based on the maximum number of threshold signatures and shares of the election commission members. We have adopted this system in our research as it can greatly contribute to establishing a fair, impartial, and transparent election process.

1.2 Research Problem Statement

In modern society, the prerequisites of an explicit and impartial election are strong security, the legitimacy of the individual's information, and anonymity. The conventional voting system has prevailed for years as it has gained widespread trust and popularity to vote for candidates or representatives by assisting in the standardization of the election procedure. However, it is no longer preferred, as this field has become one of the biggest areas for corruption and unfairness. One of the radical problems with the existing traditional voting system is that it is not unambiguous when it comes to tallying the votes. It includes numerous forms of voting fraud, polling booth theft, phony voters adding illegitimate votes, etc. This system cannot provide complete anonymity or fairness since many agencies and hackers can easily access the voters' information or tamper with their votes. Taking these instances into consideration, the online voting system, also known as the e-voting system, has been introduced. E-voting has gained much popularity in recent years due to its being much more secure, efficient, and flexible compared to the customary voting system. Not to mention the amount of budget it saves as the voters can cast their votes online from any location without having to go to the voting centers, thus saving time and effort. Nonetheless, establishing a solid e-voting system that renders legitimacy and security in contrast to the flawed prevalent voting system has been rather strenuous, as there are particular security and privacy shortcomings in the electronic voting system that still allow fraud and poll rigging. We have to ensure that the individual information of the voters is secure and inaccessible to others in order to establish an online voting process. However, the gradual increase in internet security and communication security issues has been recently addressed and is especially focused on. Encryption alone cannot facilitate anonymity in online voting systems. Because it is still possible for an intruder to trace back to the voters' confidential information through the votes that they have cast. Hence, more emphasis is being put on constructing a more reliable and secure online voting system. The blockchain approach can solve all these issues with its decentralized system, as the centralized system is way too faulty with the risks of illegal involvement. Although at first its use was limited to transactions in cryptocurrency, blockchain is now gaining popularity at an astounding pace as it can fulfill the requirements for security and transparency, as well as ensure the anonymity of the voters, elevate ballot security, and enhance the authenticity of data.

The primary strategy of this research concerns the use of an electronic voting system using blockchain to enable voters to shift to online voting, which would preserve their anonymity and enhance security by replacing the conventional voting system. The blockchain-based security method maintains two distinct blockchains, where one is used for conserving the voters while the other is used to save confidential information about the voters to improve security. Such use of separate blockchains allows the voters to vote autonomously, ensuring the safety of their personal information through the authorization of a PIN confirmation prior to tallying the votes. Another blockchain is constructed to trace the tallies of the votes by assigning votes as transactions, which would help the voters tally their own votes themselves and confirm that no votes were swapped, nullified, or added. Thus, when the results are published, everyone can settle on the ultimate count. In short, this system can guarantee fair elections free of fraud and rigging by manifesting a legitimate purpose where individuals can vote remotely and have authority over their own votes.

1.3 Research Questions

As this research deals with the concept of a whole new technology, that is, blockchain technology, which is used in order to solve the prevalent issues with the existing voting system and ensure a fair election, there might arise some questions regarding the blockchain based voting system, its types, applications, methodologies, and so on. This research provides a detailed explanation along with necessary diagrams, comparisons, tables, flowcharts and codes in answer to the following questions-

- What is blockchain?
- What are the types of blockchain?
- How does blockchain work in the voting system?
- Why is the e-voting system preferred over the conventional voting system?
- Which approach can be used to solve the existing issues in the current voting system?

- How to implement the blockchain based voting system?
- Is it possible to fully ensure a secure online voting system using the proposed system?

1.4 Objectives of the Study

The main objective of this research is to establish an electronic voting system where voters can vote remotely while preserving their anonymity and information security. Our research system will ensure that no form of voting fraud, rigging, or polling theft takes place and that voting is fair and transparent, with only legitimized voters participating in the election. This research aims to provide-

1. An understanding of the blockchain concept
2. A brief idea about the issues in the prevalent voting system
3. An understanding of the methodologies adopted to solve the issues of the e-voting system
4. An assessment of our findings
5. The proper approach to executing the blockchain based voting system
6. Limitations of the study

1.5 Analytical Framework of the study

An electronic voting system, or e-voting system, requires devices with an internet connection to conduct all the steps of the voting process that relate to the concept of cryptography, along with the basis of encryption and signature algorithms. Unlike the traditional voting system, it is a cost-effective and practical approach to ensuring high security for the large amount of data provided by the voters and representatives. The traditional voting system is being replaced with an online voting system with the purpose of enhancing voting security and protecting the confidential information of the voters, which can be attained by presenting the blockchain concept. With the objective of improving security and preserving the anonymity of the voters, we have used various techniques to utilize the blockchain to model an advanced and safe e-voting system. With the advancement of blockchain technology, the notion of decentralization has gained more acceptance. Utilizing a secured decentralized voting system, blockchain can prevail over the centralized conventional voting system by contributing its decentralized ledger technology. Blockchain can provide its outstanding properties when it comes to the security system, identity management systems, etc. Nonetheless, the Bitcoin blockchain is very secure yet has an extremely rigid validation process that is very inefficient. On the contrary, the Ethereum blockchain, which has a collaborative system among the miners, is not completely anonymous. Thus, taking these issues into consideration in our research, we have attempted to incorporate the properties of homomorphic encryption to ensure anonymity and security of the voters' information and threshold signatures, together with trustee shares, to publish and decrypt the vote results based on the majority number of threshold signatures and shares from the members of the election commission and Ethereum smart contracts to maintain transparency. This research has used homomorphic encryption together with threshold signatures and trustee shares in order to establish an optimized online voting system. Using homomorphic encryption, votes can be combined without imparting the real votes. In this case, Ethereum smart contracts provide the transparency, while threshold signatures along with homomorphic encryption are used to decrypt the votes on the basis of a fixed number of trustee shares. That is, only if a certain number of trustees, depending on the threshold, provide their signatures would the result of votes be decrypted and deployed. This system does not allow vote replacement or allow the same voter to vote multiple times, thus preventing fraudulent votes. Again, this system ensures the legitimacy of the voters from the updated election commission database. Hence, phony voters will not be able to cast votes illegally. This system can achieve the ideal benchmark of carrying out a fair election by providing high security to the voters and their information without any third party involvement.

Chapter 2

Background Study

2.1 Blockchain

Blockchain is a revolutionary technology and in recent days it has gained much popularity. Basically, blockchain is completely decentralized and it creates a platform to maintain a shared database of transactions without any central authority. This technology is getting popular day by day as it provides transparency, security, and immutability to digital transactions and data.

2.1.1 Characteristics of Blockchain

Decentralization

There exists a stark contrast between traditional centralized transaction system and blockchain or decentralized system. While the former asks for verification of each transaction by a trusted third party(for instance central bank) which ultimately leads to financial and operational bottlenecks at the main server, the latter uses consensus approach to preserve data consistency across distributed networks [28].

Anonymity

Users can communicate with the blockchain using a randomly created address that conceals their true identities. Be aware that blockchain cannot ensure complete Privacy preservation due to an inherent restriction [28].

Auditability

Using the Unspent Transaction Output (UTXO) architecture, the Bitcoin blockchain retains information on user balances: Every transaction must refer to some earlier unpaid transactions. The status of those referred to unspent transactions changes from unspent to spent once the current transaction is added to the blockchain. Consequently, transactions could be easily tracked and validated [28].

2.1.2 Types of blockchain

Public Blockchain

Public blockchain is one of the most common types of blockchain and it is open and decentralized. In this type of blockchain anyone interested can access the network to do transactions. Public blockchain utilizes proof of work or proof of stake models being used in public blockchain and the validation person earns the transaction incentives. Bitcoin and ethereum are examples of public blockchain [43].

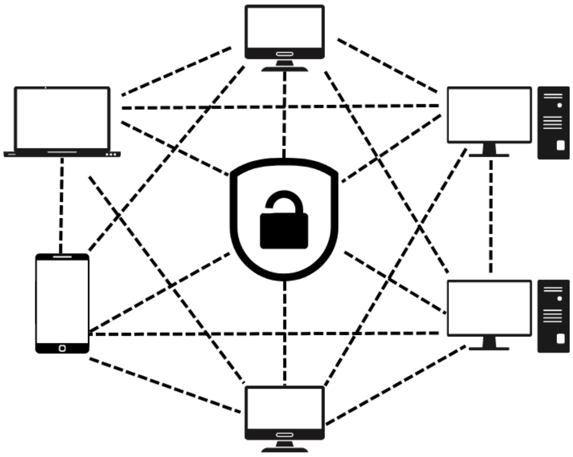


Figure 2.1: Public Blockchain

Features of public blockchain

- **Trustable**
  Unlike private blockchains, where members must consider authenticity, public blockchains are trusted. There is no transaction fraud because they don't need to be aware of other nodes in this kind of public blockchain. In this category, nodes can communicate without relying on any particular nodes [43].

- **Security**
  Connecting with other users and nodes within the same public network is possible using the public blockchain, enabling safe, extensive, and increased involvement. Due to this feature, it is challenging for attackers to access the systems, and every node will perform verifications and transactions by standards. Some experts claim that because intelligent cryptogenic encrypting techniques are applied here, it is significantly safer than the private blockchain [43].

- **Transparent**
  The openness of the public blockchain is also present, and data is transparent to all nodes in this system. One blockchain record is often accessible to all authorized nodes. As a result, there are no fraudulent transactions or informational secrets here, and all the nodes are open and transparent [43].

Disadvantages of Public Blockchain

- **Lower Transaction rate**

  Due to the vast network and numerous nodes in the public blockchain system, the transaction rate per second is also relatively low. Here, each node must verify the transaction and do time-consuming proof-of-work. Seven transactions occur in public systems each second, and the Ethereum network here has a roughly 15 TPS rate [43].

- **Scalability**

  The significant difficulty for public blockchains is scalability, which refers to obtaining consensus across distributed computer nodes in an efficient and scalable manner. Blockchain networks like Bitcoin and Ethereum need to expand the transaction throughput. Ethereum can support about 20 tps, compared to Bitcoin's 1MB block size, which allows for about seven tps. Increasing block sizes, expanding SegWit to encompass more transactions, experimenting with novel data structures like DAGs, and creating more effective consensus algorithms are among the solutions [37].

- **Sustainability**

  Public blockchains must be sustainable, which calls for effectiveness, efficiency, and decentralized governance. However, many blockchains, including some that are better than Bitcoin, becoming centralized as a result of corporate engagement [37].

Private Blockchain

Private blockchain networks, in contrast to public ones, are permissioned, meaning only those the network administrator has invited are allowed to join [35]. Access is restricted for both participants and validators without any invitation to join. Companies who wish to secure their data without giving up autonomy or running the danger of exposing it to the public internet utilize this type of blockchain network [32].

Advantages of the Private Blockchain

- **Speed**

  Private blockchains operate more quickly than public blockchains. Hence a higher TPC (transaction per second) rate can be seen here. Additionally, the speed is higher because fewer nodes are visible here. Here, all nodes are capable of processing verification, which makes it possible to add new transactions to a block at a rapid rate [43].

- **Scalability**

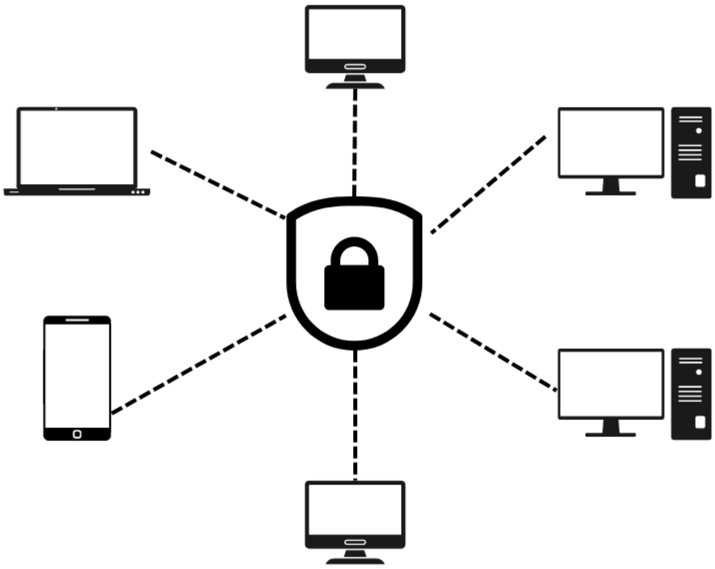


Figure 2.2: Private Blockchain

A private blockchain is faster than a public blockchain and offers more scalability. Here, adding nodes to already-existing ones is simple and quick. Private blockchains are hence more scalable and adaptable as a result. And here, adding or removing nodes has no impact on the functioning of the current systems [43].

Disadvantages of the Private Blockchain

- **Needs to Build Trust**

  As a type of open ledger, public blockchains are concerned with the security and legitimacy of each user, but private blockchains have fewer users, and so need to establish trust.

- **Reduced Security**

  A node can more easily hack the entire private blockchain system in this situation because the private blockchain is vulnerable when a third party acquires access to the central administration system [43].

Consortium Blockchain

The blockchain network is permissioned and semi-decentralized, similar to the private blockchain network consortium, except multiple businesses may each run a node on such a network rather than a single entity managing it [43].

2.1.3 Blockchain Architecture

Distributed Network

A blockchain network is made up of numerous nodes (computers) that are dispersed throughout it. These nodes participate in the consensus procedure and keep a copy of the blockchain ledger [47].

Blockchain Ledger

A chain of blocks holding transactions or other data is kept in the ledger, which is a decentralized database. A chain of blocks is created because each block carries a cryptographic hash of the one before it [47].

Consensus

Blockchains now include consensus techniques for transaction verification as a fault-tolerant approach. To maintain agreement among the network nodes, consensus is used. As the network grows, more nodes are added, making it more challenging to reach a consensus. Users must participate in a public blockchain to verify and authenticate the transactions. Blockchain involves adopting a secure method to maintain the integrity of the transactions, having participants agree on a consensus, as it is a dynamic, automated system. Different consensus techniques have been put forth, each with a unique set of underlying ideas and uses [44].

Smart Contracts

Self-executing contracts with established rules inscribed on the blockchain are known as smart contracts. Without the use of intermediaries, they automatically execute and enforce agreements. Ethereum is the most widely used blockchain for smart contracts [17].

Peer-to-Peer Network

Peer-to-peer (P2P) communication protocols are used in blockchain networks to connect nodes, communicate information, and spread transactions throughout the system. Examples include the Gossip protocol for Bitcoin and the Whisper protocol for Ethereum [47].

Decentralized Consensus Algorithm

Several algorithms are employed to reach decentralized consensus, ensuring agreement on the state of the blockchain without relying on a central authority. Examples include Ethereum's future transition to Proof of Stake (PoS) through the Ethereum 2.0 upgrade and Bitcoin's Proof of Work (PoW) system [47].

- **POS**

  PoS is a consensus algorithm where the next block's creator is selected based on the stake or ownership of a cryptocurrency. The likelihood that a participant will be chosen to validate and add new blocks to the blockchain is higher for those with a sizable amount of the coin. PoS is more energy-efficient than PoW since it requires less powerful computing resources. Several well-known PoS-based cryptocurrencies include Tezos, Cardano, and Ethereum (which is switching to PoS) [17].

- **PoW**

  PoW is a consensus algorithm that calls on users to solve challenging arithmetic problems to validate and add new blocks to the blockchain. The first miner to find the answer wins the right to add the block and the reward. Miners compete with one another to find the answer. Due to its resource-intensive nature, PoW has been frequently employed in cryptocurrencies like Bitcoin and offers security. Nevertheless, it uses a lot of electricity and computing resources [47].

2.2 Homomorphic Encryption

Ancient Greek regularly used the word homos to denote ”same,” while morphe was usually used to denote ”shape” [31]. As a result, the word homomorphism became popular and used in a number of fields. According to Malik et al. [6], a map that conserves all the algebraic relations between an algebraic set’s domain and range is known as homomorphism in the field of abstract algebra. Fundamentally, this map is a function, or an operation that accepts inputs from the set of domains and produces an element in the range, like addition or multiplication. Without explicit knowledge of m1 and m2, the encryption function E can be used to generate E(m1 + m2) using an additively homomorphic approach. This method is essential for protecting sensitive data privacy. However, until the data has been decoded, conventional encryption techniques cannot be used to protect it. In other words, individuals must give up their privacy in order to benefit from cloud-based teamwork, document sharing, and storage options. Furthermore, even after users stop using the services, unreliable servers, providers, and well-known cloud operators could still hold on to users’ physically identifying characteristics [15]. Users have serious privacy concerns about this. It would be very helpful to have a system that permits infinite operations on encrypted data without decryption. According to cryptology history, Rivest et al. (1978a) introduced the word homomorphism as a possible answer to the computation without decrypting problem in 1978. Several attempts by analysts worldwide to develop such a homomorphic scheme employing a limited set of operations were inspired by Rivest et al. (1978a). A homomorphic encryption method offers a way to directly compute encrypted data while yet maintaining privacy. A type of encryption known as homomorphic encryption enables calculations to be made on ciphertext, producing an output that is itself encrypted [2].

2.2.1 Types of Homomorphic Encryption

A cryptographic method called homomorphic encryption enables computation to be done on encrypted material without having to first decrypt it. Simply enabling addition and multiplication operations can be used to build an encryption method that allows the homomorphic evaluation of any function. This is because, compared to finite sets, addition and multiplication are functionally full sets. It's interesting to note that XOR (addition) and AND (multiplication) gates alone can describe any Boolean circuit. While homomorphic encryption can be programmed to use specific keys for these operations (asymmetric), it can also be used with symmetric key encryption and decryption. Rothblum (2011) presents a general approach that shows how symmetric and asymmetric homomorphic encryption methods can be transformed into one another [12]. A homomorphic encryption system is essentially composed of four fundamental operations namely KeyGeneration, Encryption, Decryption, and Evaluation. The asymmetric homomorphic encryption scheme generates a secret and public key pair through the KeyGeneration operation while in the symmetric system, it produces a single key. KeyGeneration, Encryption, and Decryption operations perform similar functions as their counterparts in traditional encryption methods. However, the HE-specific operation, Evaluation, differs in that it receives ciphertexts as input and produces a corresponding ciphertext for a functioned plaintext. Evaluation performs the function f() on the encrypted data (c1, c2) without inspecting the messages (m1, m2). Eval takes in ciphertexts (c1, c2) and returns evaluated ciphertexts. This process ensures that the function f() is executed over the ciphertexts without revealing the plaintext message. Consequently, the system ensures the confidentiality and integrity of the message while performing computation. The configuration of the ciphertexts should be retained after an evaluation method in order to enable an accurate decryption, which is the most important aspect of homomorphic encryption. Additionally, in order to permit an endless number of functions, the dimensions of the encrypted message must remain constant. In contrast, a larger ciphertext will use more resources and restrict the number of operations. Partially homomorphic encryption schemes are restricted to supporting the evaluation function for only addition or multiplication, whereas fully homomorphic encryption schemes can support the evaluation of any function for an infinite number of times over ciphertexts. This is true of all the homomorphic encryption techniques that are currently available and mentioned in the literature [24].

- **Partial Homomorphic Encryption (PHE)**

  A partially homomorphic cryptosystem displays either additive or multiplicative homomorphism, but not both [16]. It seems unlikely that the cryptosystem can conduct homomorphic computations on both the addition and multiplication of ciphertexts concurrently. Hence, the system is not wholly homomorphic in nature, as it lacks the ability to execute homomorphic calculations on both addition and multiplication operations concurrently. This notion of incomplete homomorphism is immensely important in the field of cryptography and has significant implications for the creation and execution of cryptographic systems. Some instances of cryptosystems that exhibit partial homomorphic properties can be observed.

  1. Paillier - additive homomorphism
  2. ElGamal - multiplicative homomorphism
  3. RSA - multiplicative homomorphism

1. **Paillier Encryption Scheme**

   The Paillier encryption methodology, which was initially presented to the scientific community by Pascal Paillier in 1999, is an exemplar of a partially homomorphic encryption scheme. This encryption system is highly intricate and is founded on the Decisional Composite Residuosity (DCR) problem, which is a computationally arduous issue, closely associated with the difficulty of factoring large integers. The Paillier encryption method's public key comprises a generator g and a composite number n, which is the product of two significant magnitude prime numbers. In contrast, the private key comprises the prime factors of 'n'. The Paillier encryption process operates by transforming the plain text message into an integer value, which is subsequently raised to the power of the public key 'n', multiplied by a random factor 'r'. The resulting ciphertext is obtained by executing a modular multiplication operation using 'g^{r}'. Decryption, on the other hand, is carried out by raising the ciphertext to the power of the private key modulo 'n^{2}' and then applying a decryption algorithm to recover the original plaintext [45].

2. **ElGamal Encryption Scheme**

   The technique used for encryption called ElGamal belongs to a category of cryptographic schemes that are public-key based and was developed by Taher ElGamal in 1985. The underlying principle of this technique is based on the computational complexity of the discrete logarithm problem in a finite cyclic group. The ElGamal encryption scheme is implemented by every user who creates a set of public-private keys to ensure secure communication. The public key comprises a generator g and a prime number p of substantial magnitude, whilst the private key is randomly selected as a secret exponent x. In the ElGamal encryption scheme, the transformation of the plaintext message into an element of the group is achieved by a process of encryption which is generally represented as an integer. To decrypt the ciphertext, the recipient utilizes their private key exponent x to compute the shared secret g raised to the power xy modulo p and subsequently applies modular inverse operations to retrieve the original plaintext [3].

3. **RSA Based Encryption Scheme**

   The RSA (Rivest-Shamir-Adleman) encryption algorithm is a highly widespread form of public-key encryption. It was first introduced to the world by a team of three brilliant individuals: Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. This encryption technique is based on the computational complexity involved in factoring large composite numbers into their prime factors. Semantic security, which ensures that malicious parties cannot extract any coherent intelligence about the plaintext from the ciphertext, forms the foundation of RSA encryption security. This quality makes RSA encryption suitable for a range of uses, including key exchange protocols, secure communication, and digital signatures. Large composite numbers are challenging to factor form the basis of RSA encryption security. The RSA encryption method becomes more secure when higher prime factors are used in modulus N. By shielding communications from interception and unwanted access, RSA encryption offers a dependable and secure mode of communication. The RSA encryption system continues to be secure since higher prime factors are used in the modulus N [1].

- **Fully Homomorphic Encryption (FHE)**
  A cryptosystem may be categorized as fully homomorphic when it shows both additive and multiplicative homomorphism characteristics [22]. The homomorphism mentioned above properties allow the cryptosystem to maintain mathematical operations without decryption. The extensive comprehension of these homomorphic characteristics will culminate in a cryptosystem that can execute any arbitrary computation on encrypted data without decryption while ensuring the data's confidentiality.

  1. Gentry's FHE scheme
  2. Recent Advancements in FHE Schemes

  1. **Gentry's FHE Scheme**

     Gentry's Fully Homomorphic Encryption (FHE) methodology, which allows for the computation of arbitrary calculations on encrypted data without prior decryption, was created by a computer program. Gentry created this groundbreaking cryptographic system and introduced it in his seminal paper published in 2009. Gentry's FHE approach is founded on bootstrapping, which enables computation on encrypted data while maintaining encryption. It uses symmetric encryption, asymmetric encryption, and lattice-based cryptography. The FHE method uses a public key to encrypt the input data, making it possible for anybody to do calculations on encrypted data. Numerous homomorphic operations, including addition and multiplication, can be performed on the encrypted data without disclosing any information about the original data. The result is obtained in an encrypted format [10].
  2. **Recent Advancements in FHE Schemes**

     Modern Fully Homomorphic Encryption (FHE) systems have undergone significant development, which has enhanced the effectiveness, security, and usability of Gentry's original approach. These developments have made it easier to apply FHE practically for various purposes. In this recent time, Fully Homomorphic Encryption has been advanced in several area such as Efficiency Improvement, Noise Reducing Algorithm, Verifiable Computation and Homomorphic Authentication, Hardware Optimization and many more.
     Except for secure data analytics, privacy-preserving machine learning,

and secure computation outsourcing, these recent developments have only opened the door for one field of practical applications of FHE [13].

2.3 Threshold Signature

Digital Signature

Digital signature has been a revolution for blockchain Technologies. Digital Signature allows us to validate and proof transactions in a chain. Throughout the advent of digital signature it has been complimented as of immense importance. Digital signature is the guarantee that a digital asset in a decentralized platform has a valid owner. Through many rigorous efforts many changes have taken place to conventional digital signature, its architecture and how it works. Different variations have come into being, each with its own benefits and problems. All this came into being because of one goal and that is to authenticate any transaction carried out in a blockchain. Without the authentication everything built on a blockchain comes to a deadend, even scarier it could cost bleach of valuable information and a lot of money [29].

Overview of Digital Signature (RSA, DSA)

There are different parts in a digital signature. For example, key generation, signing a transaction and validating the transaction. During key generation a pair of public (pK) and private key (sK) is generated. The private key is used to sign the contract, then the public key is used to verify the transaction. The private key is always kept secret whereas the public key is shareable. But there are vulnerabilities in maintaining the integrity of this type of signing system [36].
In RSA digital signature a signer signs the message with his private key and then another person could decode the message through signers public keys. This is the RSA digital signature overview. But there lies a problem that anyone with knowledge of the private key of the signer can access the message or anyone who could decrypt the private key can see the message. The shield here is minimal [8].

DSA has the same steps as RSA. But the encryption and decryption calculations are different. In DSA the message is hashed incorporating the public and private keys, then the total signature with the keys and hashed message is taken to the receiver where validity is ensured with complex computation with the public key. DSA is a better and more complex form of RSA, but with a quicker decryption process [4].

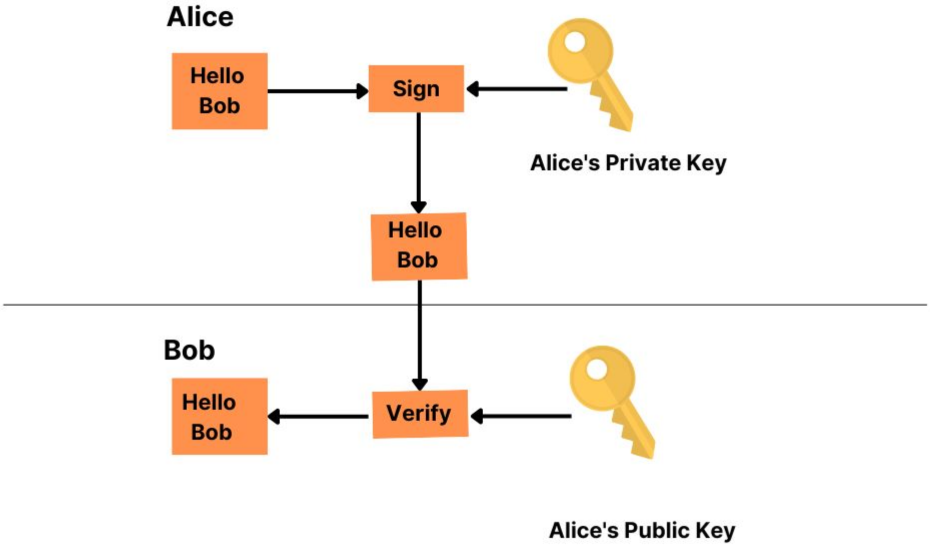


Figure 2.3: RSA Digital Signature

Intro to Threshold Signature

Having a single level signature formation does not give much protection to the signature schemes. For such security reasons came into being multilevel signatures. In multi-signature participation of multiple signers together make one single signature used to sign the message. On the other hand Threshold signature lets multiple signers build a single signature and decipher a message with the participation of a specific number of signers (subset of total signer). It does not need the collaboration of all the signers to decrypt a message like a multi-signature [48].
At the advent different threshold signatures came into being. First one came from Botd, where 2 out of l RSA is followed. Out of l signers 2 shareholders can sign a message. The shareholders themselves can be anonymous and share a common public key. One shareholder only needs to collaborate with another to form a signature. This signature is close to RSA, but which has a more secure algorithm is ambiguous. Then there is the t out of l scheme of Croft and Harris. This scheme has a backdraw, and that is less than the threshold number of signers can compromise the system. Lastly, Desmedt came up with another form where encryption is done with shares from 50% of the total signers. This scheme is not for individual use, rather between organizations [5].

Threshold Signature Types and Schemes

Different types of threshold schemes came into being. Few of them are given below. popular ones are noted here.

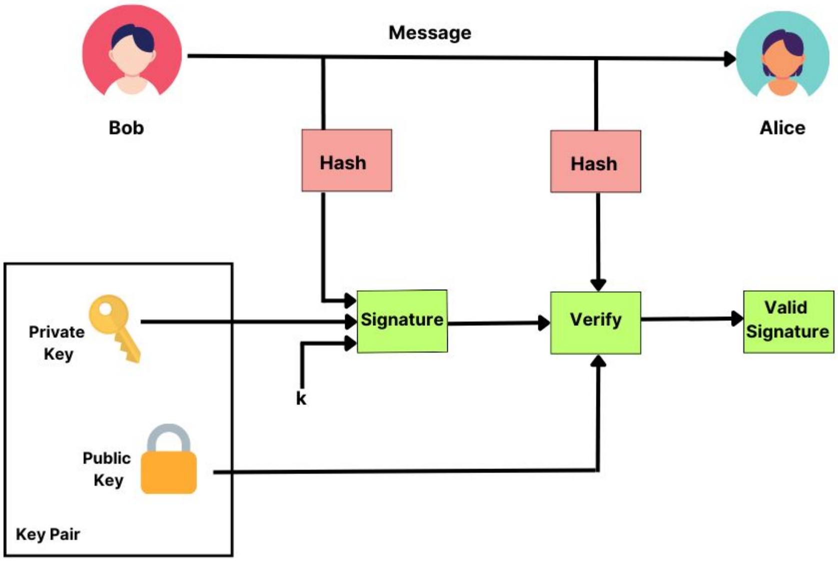


Figure 2.4: DSA Digital Signature

- **Samir's Secret Sharing Scheme**

  In this scheme n number of people share a secret and only k or more (k¡s) number of shares is required to get back the secret. Each participant holds a fragment of the main message, but the total message can be found out without the collaboration of all the shareholders [34].

- **Distributed Key Generation**

  Here the signers are given a public and private key pair after generating them. The private key will never be known to other signers. The key is generated only in collaboration of all the signers, but the whole of the private key is not known to none [9].

Applications of Threshold Signature

After Threshold Signature was popularized, it is now used for many applications.

- **Securing Bitcoin Wallets:** Bitcoin wallets have been prone to vulnerabilities for long. Kaspersky labs mentions that there are over millions of attacks every month [14] This number is very alarming especially considering the importance these wallets hold and the immense price that is kept on these wallets. Threshold signature gives multilevel protection to these wallets [11].

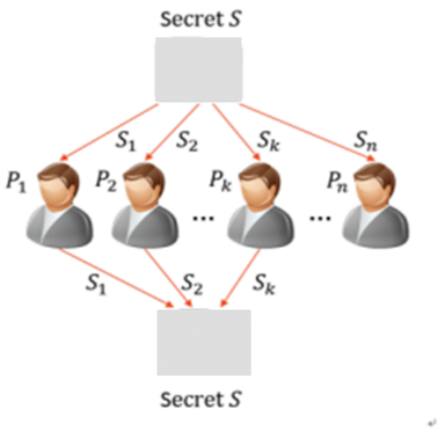


Figure 2.5: Samir's Secret Sharing Scheme

- **Securing message sharing among peers:** Through threshold signatures the group conversation can now be protected or kept secret from the rest of the world with the signature created in collaboration with the members of the group. This form of communication is already into existence [7].

- **Cloud protection:** Another technology that has got a boom in the last few years is Cloud Computing. With more businesses using cloud computing, the integrity of the information needs to be maintained more than ever. These sensitive data can now be protected through the use of threshold signature [18].

- **Internet Of things:** Many times components that fall under the internet of things do have sophisticated mechanisms to protect their information from vulnerabilities. Most of the time the builder does not heed any importance to the safety of the information transferred by the device over the internet. Any information surfacing on the internet is very vulnerable. Threshold signature has emerged as a solution here [42].

To conclude we come to a common spot on the importance of threshold signature, that this finding has given blockchain a new turn in terms of security of the transactions. It allows the transaction to come over from the previous digital signature mechanisms with a single point of vulnerability. There is no single point to attack now. With this advantage in mind many other applications of the threshold signature have emerged and will continue to emerge in the near future.

Chapter 3

Literature Review

In this article, Thuy et al [33] thoroughly examine the prerequisites before presenting Votereum, an E-voting system that leverages blockchain technology. The suggested system is fortified by the Ethereum platform, which comprises a singular server that oversees the entire system, and another that handles all blockchain-related requests. Additionally, the implementation has been deployed to the Rinkeby testing network utilizing Angular UI, NodeJS servers with RabbitMQ to evaluate the feasibility, along with a discussion on some security concerns. The primary functions of the Votereum system encompass generating fresh ballots, registering voters, casting votes, and retrieving the current vote count. Overall, the system allows users to access the voting system through the user interface, interacts with the Ethereum network through the Online Ballot Regulator (OBR), and ensures data credibility and security via blockchain technology.

In this study, Ahn et al [46] introduce a system for electronic voting on the Ethereum platform. This system effectively addresses the issue of fraudulent voting through the enhancement of safety and reliability in the electronic voting process. The system's implementation is based on Ethereum blockchain technology, employing Express.js for webpage development and Web3.js for front-end integration. The system secures voter accounts through AES encryption and stores them in a database. Voting is conducted through smart contracts and includes functions for casting, confirming, and counting votes. Nevertheless, challenges persist in ensuring direct participation and preventing proxy voting in remote scenarios.

In this paper, Zhang et al [48] propose a protocol for electronic voting (e-voting) on the Ethereum platform called Ques-Chain. The protocol ensures authentication without compromising confidentiality and anonymity without exposure to scams. The essay highlights three essential contributions. Firstly, it proposes a message authentication and transmission mechanism that enables permission checking while preserving anonymity in various scenarios. Secondly, it decouples the blind signing and checking process into three steps, which maintain anonymity and message confidentiality while not sacrificing authentication. Thirdly the protocol has been developed utilizing dependable computing technology on the Ethereum platform, thereby ensuring the credibility of all parties involved.

In this article[30], the author highlights Ethereum as a highly suitable platform for implementing e-voting applications. The study specifically focuses on implementing and testing a sample e-voting application as a smart contract on the Ethereum network using Ethereum wallets and the Solidity language. To address the issue of a transparent and reliable environment, the author suggests utilizing peer-to-peer technology. The results of the contracts can be calculated by all peers since Ethereum does not require a centralized organization to deliver proof-of-work (PoW).

In this scholarly article, Khoury et al [27] present a novel method to address trust issues in voting systems using Blockchain technology. The system ensures data integrity, transparency, and privacy while enforcing a singular vote per mobile phone number. The authors verify their proposed voting scheme, which utilizes Solidity, NodeJS, Web3js, and an HTML5 web app compiled with Apache Cordova. The Ropsten Testnet simulates the Blockchain network, and Twilio is used as the SMS gateway API. The system architecture facilitates the creation and management of voting events, authentication of voters, and real-time visualization of voting results. User registration requires a unique PIN, and voting confirms registration and adjusts the vote count accordingly.

In this article[41], the authors explain their objective to design a digital voting architecture, including a smart contract, to provide authentication, transparency, anonymity, and accuracy. The architecture works by creating a chain of hashes, with each hash function created from each voter's information. Any changes in the hash would be detectable. The miner is selected by the smart contract, considering various factors including energy usage and data transmission. In order to manage enormous data created during voting, the system records the data systematically using three different storage systems: one to store voters' information, one for blockchain storage, and a separate copy of the Election Commission's database. Finally, their system's counting mechanism reduces time consumption, as the process counts votes in each block. After the voting process terminates, the last block reveals the total votes.

In this study [26], the author proposes a blockchain-based e-voting system utilizing Crypto-voting. The system is designed as a multichannel hybrid system, and the author aims to employ a permissioned blockchain to ensure access control and preserve anonymity. The model consists of two interconnected chains: the first side chain governs all voting operations, while the second side chain manages the final result. To facilitate the voting process and outcome, a smart contract is employed. The author asserts that this model would particularly benefit remote users, such as voters living abroad, as they would be able to cast their votes using mobile devices or personal computers.

In this paper, Yi et al [35] emphasize the several methods that can be utilized to exploit blockchain technology in P2P networks to strengthen the security of e-voting. Firstly, a synchronized voting record model is recommended, which is founded on distributed ledger technology (DLT) and aims to diminish the likelihood of vote falsification. Secondly, for authentication and non-repudiation of credentials, a cryptographic algorithm has been developed by utilizing Elliptic Curve Cryptography (ECC). Additionally, a withdrawal model is introduced, which empowers voters to modify their vote until a predetermined deadline. For significant protocols of e-voting systems, they proposed a blockchain-based e-voting scheme in P2P network by deploying the above designs. To verify and substantiate the proposed plan, a blockchain-based electronic voting system, designed for multiple candidates, is implemented on Linux platforms within P2P networks.

In this scholarly article, Khan et al [38] introduce a thorough investigation into the limitations of functionality and scalability of an electronic voting platform. The study involves rigorous experimentation with permissioned and permissionless blockchain settings in various scenarios, considering factors such as voting population, block size, block generation rate, and transaction speed. Initially, voters and candidates participate in designated locations using voting machines, and votes are recorded as unconfirmed until miners confirm and update the blockchain ledger. The system ensures voting secrecy, verifies entitlement, maintains confidentiality, and represents votes as blockchain assets. The registered voters are grouped into polling stations, and only those registered voters at a station can use voting machines. The votes are cast by transferring tokens from voter to candidate addresses through blockchain transactions, confirmed by miners, and tallied for results. The experiments have emphasized interesting insights into the impact of these parameters on overall efficiency and scalability.

In this study [39], the author proposes a combined model. This model combines two consensus protocols, Proof of Stake(PoS) and Proof of credibility(PoC). In this model, two blockchains can work in parallel. Thus a private chain functions as a side chain and stores hashes of the public chain inside of its own block. Sharding is introduced for effective data management during the process. Here nodes are assigned at random using Verifiable Random Functions (VRF) and Verifiable Delay Functions (VDF). To ensure a reliable public bulletin board and a secure computing environment, the model intends to use a smart contract. Finally, a dedicated server enables node authentication and includes user credentials to publish ballots to public blockchain.

In this research [19], the author proposes a model by pointing out that the existing voting schemes based on homomorphic encryption need an overseer which makes it vulnerable to single point of failure. The proposed model of this research thus intends to use a multiplicative homomorphic approach. This approach would provide special privileges to voters and these privileges would help to protect voters' confidentiality, anonymity, and reliability on the whole system. The suggested method relies on both public and private clouds. It utilizes the advantages of public cloud to authenticate, compute, and multiply encrypted messages before showing the results.Finally, the private cloud would be used by election administration to store candidate and voter information and to calculate the final results after decrypting the data.

In this study [20], the author proposes a model based on the Paillier homomorphic encryption scheme. This scheme has an additive homomorphic property and is a probabilistic asymmetric encryption method. The suggested model uses public and private keys as its main components. A unique integer produced from two prime numbers is contained in the public key. A special formula is used to transform it into a hard-to-read form to encrypt the vote. To decrypt it, only the private key would be needed. The private key also provides functionalities like performing calculations on encrypted messages. To sum up, the Paillier cryptosystem enables performing calculations on encrypted messages in e-voting systems without revealing the actual content of the messages until the final decryption step. The proposed system includes a graphical user interface (GUI) and will be implemented using the Java programming language, with MySQL utilized as the database.

Authors	Platform	Applied Technique
Thuy et al	*Ethereum*	*Deployed Rinkeby testing network utilizing Angular UI, NodeJS servers with RabbitMQ to evaluate the feasibility* *Generate fresh ballots, registering voters, casting votes, and retrieving the current vote count through OBR*
Ahn et al	*Ethereum*	*Employed Express.js for webpage development and Web3.js for front-end integration* *Secured voter accounts through AES encryption and stores them in a database* *Voting is conducted through smart contracts and includes functions for casting, confirming, and counting votes*
Zhang et al	*Ques-Chain* *(Ethereum based)*	*Proposed a message authentication and transmission mechanism that enables* *permission checking while preserving anonymity* *Decoupled the blind signing and checking process into three steps,* *which maintain anonymity and message confidentiality while not sacrificing authentication*
Yi et al	*Linux* *(within P2P Network)*	*A synchronized voting record model based on DLT* *For authentication used a cryptographic algorithm called ECC* *Introduced a withdrawal model to modify vote before predetermined deadline*

Table 3.1: Comparative Analysis

Yavuj et al	*Ethereum*	*Used Proof of Work algorithm* *Peer to Peer Technology*
Khoury et al	*Ethereum*	*Enforced a singular vote per mobile phone number technique* *Verified their proposed voting scheme, which utilizes Solidity, NodeJS, Web3js, and an HTML5 web app compiled* *with Apache Cordova* *Ropsten Testnet simulates the Blockchain network, and Twilio is used as the SMS gateway API* *User registration requires a unique PIN, and voting confirms registration and adjusts the vote count accordingly*
Fusco et al	*Permissioned* *Blockchain*	*System is designed as a multichannel hybrid system* *The model consists of two interconnected chains: the first side chain governs all voting operations,* *while the second side chain manages the final result*
Azougaghe et al	*Not mentioned*	*A system based on a multiplicative homomorphic approach that provides special privileges to voters* *to ensure privacy, anonymity, and reliability* *A public cloud, which authenticates, computes, and performs the multiplication of encrypted messages,* *displaying the final results* *Store candidate and voter information and to calculate the final results after decrypting the data*

Table 3.2: Comparative Analysis

Anggriane et al	***Not mentioned***	***The system has two main components: the private key and the public key*** ***The public key contains a special number derived from two prime numbers. To encrypt a vote, a formula*** ***is used to transform it into a encrypted form*** ***The private key is needed to decrypt the message and convert it back to its original form*** ***The proposed system includes a graphical user interface (GUI) and will be implemented using the*** ***Java programming language, with MySQL utilized as the database***

Table 3.3: Comparative Analysis

Chapter 4

Methodology

Our proposed system focuses on employing Ethereum smart contracts to achieve transparency and homomorphic encryption along with threshold signatures and trustee shares in order to encrypt data, preserve the anonymity of the voters, elevate security, and decrypt the results depending on the trustee shares. In order to carry out the entire process it has adhered to the following steps-

1. Voter Registration
2. Voting
3. Party Registration
4. Contract Deploy
5. Vote Aggregation Homomorphically
6. Threshold Signature and Homomorphic Decryption to Get Result

4.1 Voter Registration

To get voting rights a voter has to be of age and has to abide by some credentials. For example, for being eligible to vote a voter has to have citizenship, be 18 years and above, have national identification, etc. The voters have to be validated with their respective information. The validators will match the given credentials with the recorded information in the nation database. If all information matches the voter will get the voting right.

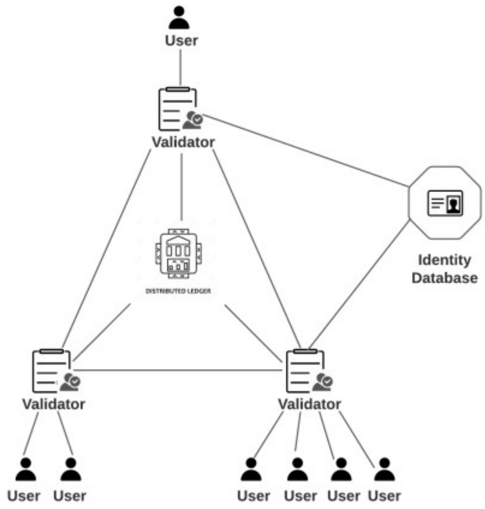


Figure 4.1: Voter Registration

4.2 Voting

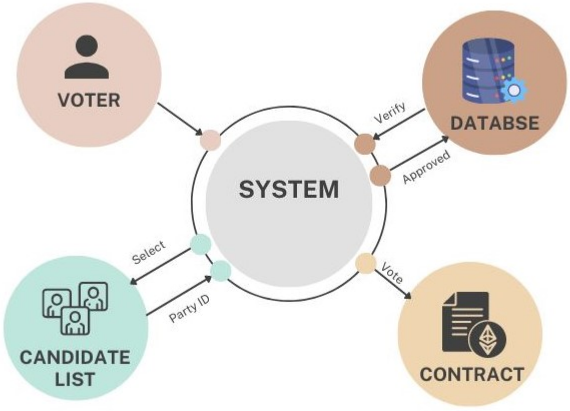


Figure 4.2: Voting Mechanism

On the voting day, the user will be able to vote online and login in with required information. The login information will be verified from the already updated Election Commission database, and the voters now can see all the candidates to vote from. The voters will choose from the candidates list and a pop up will show if the voter is sure to vote for the specified candidate. If yes, then the vote will directly be taken to the smart contract with the party ID of the selected party.

4.3 Party Registration

The Party name and the party ID of the nominated candidate is taken in by the system and checked with the already updated national database and match the information. If a match is found the address of the candidate will be kept in an array using the pardyId as index. A separate array will keep the other information related to the candidate. All this will be done by the election commissioner after he calls the method responsible for registering the parties.

4.4 Contract Deploy

In figure 4.3 we can see that the contract will be deployed by the Election Commissioner with the number of trustees, number of minimum trustees needed to get the result (threshold) and the public IDs of the trustees. The contract will first check if the given threshold is lower than the number of trustees, The public keys of individual trustees will be mapped to an integer as a public key. Also the public keys will be stored in an array.

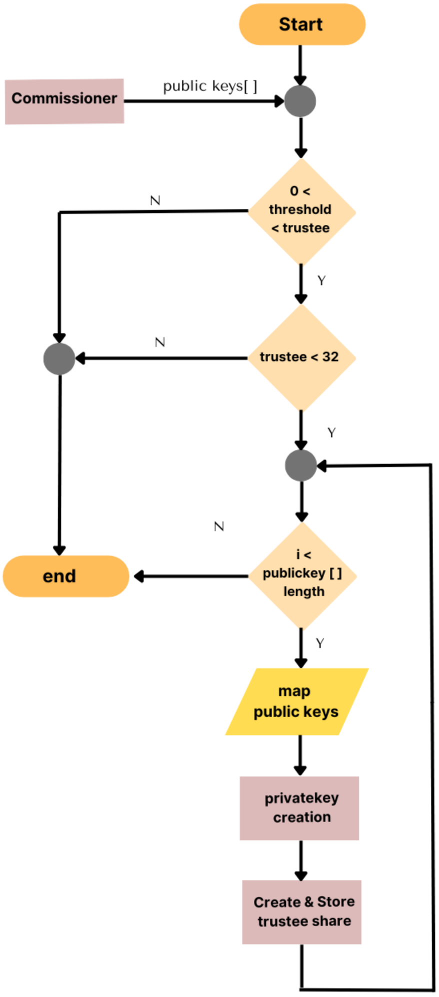


Figure 4.3: Contract Deploy

Now we take an array of length equal to the number of trustees. For each trustee corresponding to their public key we produce private keys, such that private equals to an integer. This number will now be pushed into the array for private keys for the trustees. Now with the given public key and the newly computed private key, a share will be produced for each of the trustees. For simplicity of calculation we produced a share which is equal to the product of 5 and the private key added to the public key. This share calculation can be done in a more complex manner. The trustee share now will be kept in the list of trustee shares according to a specific index reserved for specific trustees. While we calculate each share, we also aggregate them for future reference.

4.5 Vote Agregation Homomorphically

First we have to check if the party ID of the voted candidate is a valid one. Then for counting the vote we have to add 1 to the already counted vote for the specific party. But as we dont want to reveal the aggregate vote, we have to aggregate the vote homomorphically. For this we have to first homomorphically encrypt the value 1.

For encryption we have to maintain the following calculation: We take two co-prime numbers p and q, We set;
$p = 42$;
$q = 43$; (p and q are two co-prime numbers so that gcd(pq, (p-1)(q-1))=1)
$n = p * q$;
$g = n + 1$; (g can be randomized; but simpler to take n+1 if p and q are of equal length)
$\lambda = lcm(p - 1, q - 1)$;
$\mu = (L(g^{\lambda} \mod n^2))^{-1} \mod n$; (here, L(x) = (x-1)/n)
Public key, pk=(n,g)
Private key, sk=(λ, μ) [22]

4.5.1 Encryption

Now for encryption, if m is the message
Then cypher text, $c = g^m * r^n \mod n^2$; where r is a random number (0<r<n)

4.5.2 Cyphered Vote Aggregation

For summation, we get
$Sum = (CypherText1 * CypherText2) \mod n^2$ [22]

4.6 Threshold Signature and Homomorphic Decryption to Get Result

As in figure 4.4 we see that, to get back the result we have to have a required number of shares from the trustees to decrypt the result and also the deadline for the vote needs to be passed. Again in the function to get results, the Election Commissioner has to send the partyId of the party whose vote count the he/she would like to see. Along with partyID the shares of the trustees who will participate in the vote decipher process also needs to be passed as parameters. Here the number of the participating trustees in getting the result needs to be equal or more than the threshold. Then if the party id is valid the shares of the trustees are aggregated. The sum is matched with the aggregated share from the previously counted share list. If they match then the vote count of the particular party is pulled out of the array.

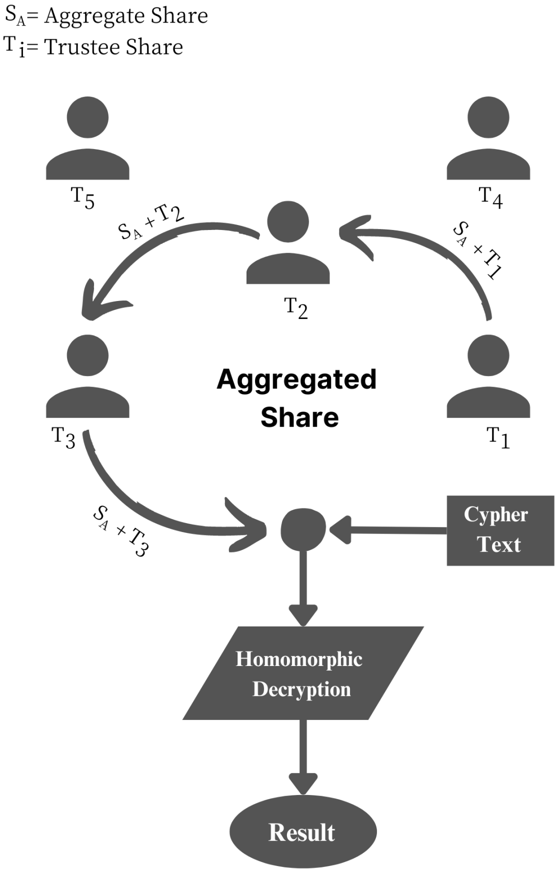


Figure 4.4: Vote Decryption

Lastly, the encrypted result is to be deciphered. Due to having computational complexity the decryption is done outside the ethereum contract. The description will be done in the following way:

4.6.1 Decryption

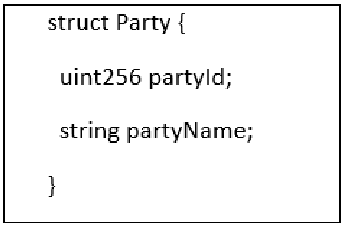
$m = L(c^{\lambda} \mod n^2)\mu \mod n$ [22]

Chapter 5

Result and Evaluation

5.1 Party registration

In our system we are going to have a party and voter registration process, where the voter registration is done off contract and the party registration will be on contract. Firstly, when we give a party ID and a party name then the party registration will be done. This can be confirmed through the outcome of the struct which will be updated with the new party instance of the registered party.



```
struct Party {

  uint256 partyId;

  string partyName;

}
```

Figure 5.1: Party Registration

5.2 Vote

Here to vote the vote function is to be called by the voter. After all the conditions are met the voter can vote for their desired party. Here when the voter selects the candidate, in this case for a party with partyId 1, the vote will be homomorphically encrypted, and will be aggregated with the previous vote count. As the aggregation is done on encrypted data, the sum is also a cipher text, which holds the actual result. If we see the party information for partyId 1, then we will see, in encrypted vote count the value becomes 911858, which is the cipher text for vote count 1, as after the first vote the vote count became 1.

5.3 Share Calculation and Aggregated threshold Signature

When the Election Commissioner deploys the contract; the contract takes in the public keys of the trustees and calculates the individual share of the trustees. When we want to see the trustee share of the first trustee, we can just call the first instance from the trustee share list. Like this all the shares from each trustee can be retrieved from the list.

For the aggregated share, we only take the share of the trustee members who will take part to create the threshold signature to retrieve the result. The function takes a list as input, containing the shares of the participating trustees (who are taking part in building the threshold signature). The aggregation of the shares gives a uint256 value. If it matches the threshold signature previously counted then the result is retrieved and deciphered.

5.4 Case Scenario

Here we take a case scenario, of four Parties party A,B,C and D. Their partyIds are respectively 1,2,3 and 4. Now if we cast 4 votes on A, 6 votes on B, 6 votes on C, and 3 on party D, then let us see what result we get,

Party ID	Party Name	Votes	Encrypted Votes
1	A	4	1644320
2	B	6	2239612
3	C	6	2239612
4	D	3	291970

Table 5.1: Vote Count

5.5 Cost Analysis

5.5.1 Comparison between trustee number and cost

Here we are putting a number of trustee vs cost calculated when the contract is developed.
Here in the x-axis, there is a number of trustees increasing from left to right; and in y axis the cost is increasing from bottom to top in the gas unit.
From the graph in figure 5.2 we can see that as the number of trustees increases the price for deploying the contract increases. So the more the number of trustees, the more the cost. Similarly the number of trustees participating in threshold signature the more the cost. In the above graph we see with 1 increase in the number of trustees there is a little growth in cost. With more increase in trustees, the increase in cost is at the same rate. So the growth is linear. So there is a proportional relation between number trustees and the gas cost.

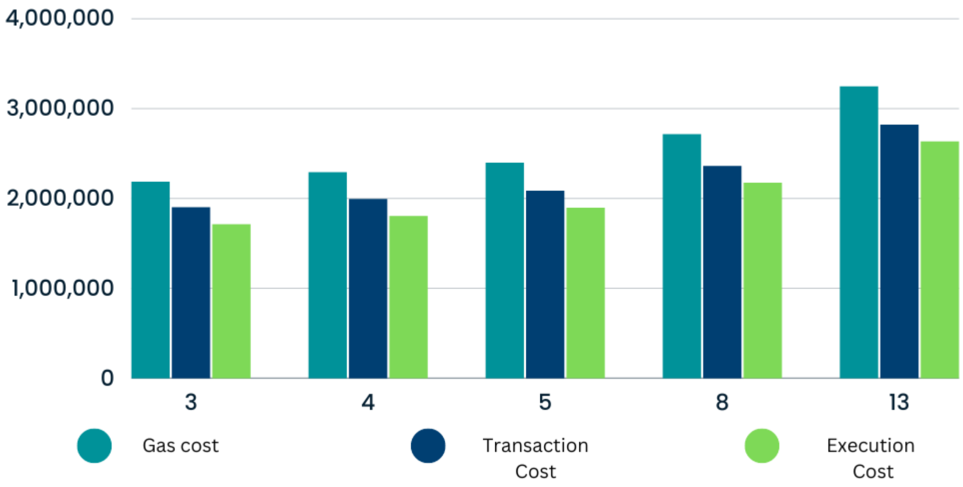


Figure 5.2: Cost To Deploy Contract

5.5.2 Function Call Cost

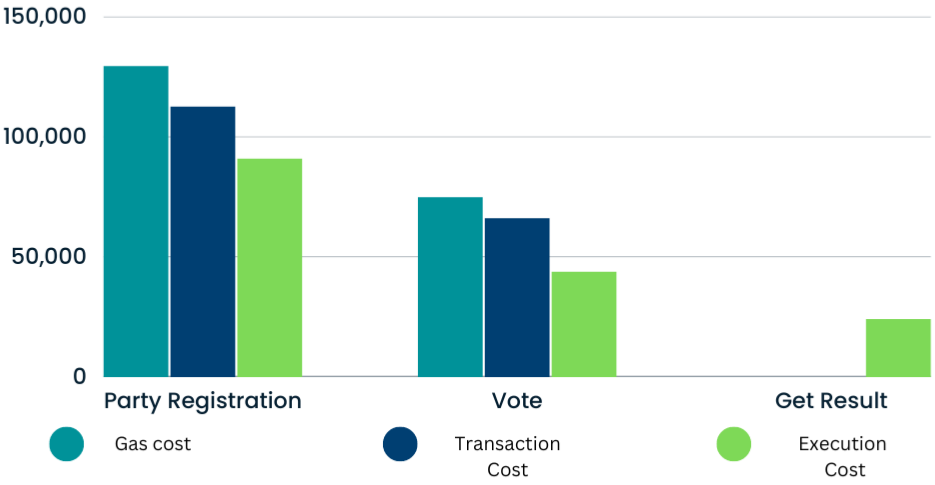
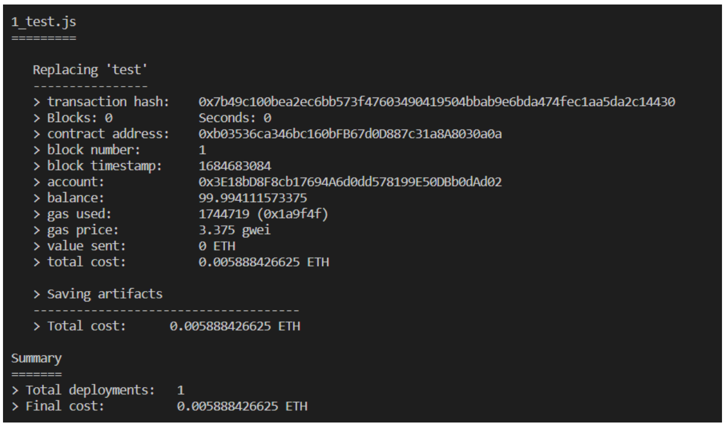


Figure 5.3: Cost For Calling Different Function

Here we can see the cost related to different functions. There are other functions but the core functions and cost related to them are described in the graph. The most is spent in the Party registration method. Least gas cost is for the Get Result method. It is also noticeable that the Get Result function do not have any transaction cost or gas cost. This is because there is no transaction happening in the function.

5.6 Genesis Block Creation

This is the information of the first block created on deployment of the smart contract



```
1_test.js
=========

   Replacing 'test'
   ----------------
   > transaction hash:    0x7b49c100bea2ec6bb573f47603490419504bbab9e6bda474fec1aa5da2c14430
   > Blocks: 0            Seconds: 0
   > contract address:    0xb03536ca346bc160bFB67d0D887c31a8A8030a0a
   > block number:        1
   > block timestamp:     1684683084
   > account:             0x3E18bD8F8cb17694A6d0dd578199E50DBb0dAd02
   > balance:             99.994111573375
   > gas used:            1744719 (0x1a9f4f)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.005888426625 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:      0.005888426625 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.005888426625 ETH
```

Figure 5.4: Genesis Block

Chapter 6

Conclusion and Future Work

In this paper, we have attempted to replace the conventional centralized voting system by introducing a blockchain based e-voting system where, by incorporating homomorphic encryption with threshold signature and trustee share a fair election can be processed. Because of its decentralization concept, it has achieved much recognition by the developed countries. Many developing nations like Bangladesh are also adopting this into their security systems in order to maintain high confidentiality, pliability and authenticity. The proposed system offers a complete modification of the existing faulty voting system by demonstrating an upgraded blockchain based electronic voting system which if implemented in the correct fashion would bring about outstanding outcomes in the voting system as it would enhance the most important features such as- security, confidentiality, transparency, viability, verifiability, and cost of a voting system with the exceptional attributes of homomorphic encryption integrated with threshold signature and trustee share. Although there are some constraints such as restricted accessibility, this system has tried to engage the utmost use of these methods to bring out the maximum results in order to guarantee a fair election. In the future, by modifying this system using more advanced algorithms that are yet to be discovered, it can also be used at international levels. Our future work involves optimizing algorithms, improving energy efficiency, scalability, and addressing anomalies in blockchain systems. We aim to make the system more accessible and secure by focusing on user-friendly interfaces, strict security measures, and continuous advancements. Besides, we will introduce N to N Verify Ability. This is the ways for voters to know their vote was counted in right way. Also, we are planning to remove the receiptfreeness of this system. If we remove this, every voter will get a receipt after their voting process. To conclude, gaining the desired level of security and fairness in the electoral process is not a far-fetched idea anymore as the use of decentralized blockchain is increasing profoundly leading to a safer, more transparent world.

While the proposed study aims to revolutionize the voting system, it's important to acknowledge and address the limitations that can affect its effectiveness. There are few limitations that our model is unable to control-

- Malicious use by individuals: The proposed system may not achieve its goal of transparency and fairness if individuals with malicious intent manipulate the voting process. For example, teams participating in the voting system could potentially visit voters' houses and forcefully cast votes in favor of their team.

- Coerciveness: The proposed system may not achieve its goal of transparency and fairness if individuals with malicious intent manipulate the voting process. For example, teams participating in the voting system could potentially visit voters' houses and forcefully cast votes in favor of their team.
- Limited accessibility of blockchain: While blockchain technology holds promise for enhancing transparency and security in voting systems, its widespread adoption and accessibility may still be limited, particularly in developing countries. Factors like limited internet connectivity, a lack of infrastructure, and a high illiteracy rate can pose challenges to the implementation of blockchain-based voting systems.
- Hindrance of Universal Verifiability: This systems are not allowed to give access to anyone the right for verifying the result after the voting. Anyone can not have the clarity about the result.
- Inability to replace votes: The proposed system's limitation of not allowing vote replacement after casting can be problematic if voters using devices accidentally make incorrect selections.

Since blockchain technology is still on the rise, there might be some anomalies in the systems implemented using it. Because these systems require a lot of different complicated algorithms, optimizing their time and space complexity requires a lot of work, which is inefficient. Different methodologies of blockchain such as the Ethereum smart contract, Threshold signature, Ring signature etc., demand high energy input thus resulting in scalability issues and sometimes generating unyielding data and reducing performance. Such issues with the existing algorithms are making the familiarization of these systems much more difficult as blockchain is a whole new concept. Our proposed system attempts to make the voting process a lot easier, secure and viable by using the concept of homomorphic encryption along with threshold signature and trustee share. However, due to the lack of advanced algorithms and more feasible methodologies, some limitations might still persist for instance, limited accessibility because of poor internet connectivity and inability to replace votes as this system does not allow voters to vote multiple times. As the world has yet to discover more about this technology, these limitations can be solved in future with much more advanced algorithms and by improving the existing algorithms thus establishing a completely faultless and safe online voting system in future.

Bibliography

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978. [Online]. Available: https://doi.org/10.1145/359340.359342.

[2] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[3] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985. [Online]. Available: https://doi.org/10.1109/tit.1985.1057074.

[4] NIST, "The digital signature standard," *Communications of the ACM*, vol. 35, no. 7, pp. 36–40, 1992. DOI: 10.1145/129902.129904. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/129902.129904.

[5] Y. Desmedt, "Threshold cryptosystems," in *Lecture Notes in Computer Science*, 1993, pp. 1–14. [Online]. Available: https://www.iacr.org/cryptodb/data/paper.php?pubkey=167.

[6] D. S. Malik, J. N. Mordeson, and M. K. Sen, *Fundamentals of abstract algebra.* 1997. [Online]. Available: http://ci.nii.ac.jp/ncid/BA30881770.

[7] C. Wang, *Generalization of threshold signature and authenticated encryption for group communications*, 2000. [Online]. Available: https://search.ieice.org/bin/summary.php?id=e83-a_6_1228.

[8] Y. Cao and C. Fu, "An efficient implementation of rsa digital signature algorithm," in *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2008. DOI: 10.1109/icicta.2008.398. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4659731.

[9] *Distributed key generation for the internet*, 2009. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5158416.

[10] C. Gentry, *A fully homomorphic encryption scheme*, 2009. [Online]. Available: https://evervault.com/papers/gentry.pdf.

[11] A. Kate and I. Goldberg, *Distributed key generation for the internet*, 2009. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5158416.

[12] R. Rothblum, "Homomorphic encryption: From private-key to public-key," pp. 219–234, 2011. [Online]. Available: https://doi.org/10.1007/978-3-642-19571-6_14.

[13] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the 44th Symposium on Theory of Computing - STOC '12*, 2012. [Online]. Available: https://doi.org/10.1145/2213977.2214086.

[14] K. Labs, *Financial cyber threats in 2013. part 2: Malware*, 2013. [Online]. Available: http://securelist.com/analysis/kaspersky-security-bulletin/59414/financial-cyber-threats-in-2013-part-2-malware/.

[15] R. McMillan, *Apple finally reveals how long siri keeps your data*, Apr. 2013. [Online]. Available: https://www.wired.com/2013/04/siri-two-years/.

[16] L. Morris, *Analysis of partially and fully homomorphic encryption*, 2013. [Online]. Available: http://gauss.ececs.uc.edu/Courses/c5156/pdf/homo-outline.pdf.

[17] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014. [Online]. Available: https://ethereum.org/ethereum.pdf.

[18] *Confidentiality in the cloud*, 2015. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7031826.

[19] A. Azougaghe, M. Hedabou, and M. Belkasmi, *An electronic voting system based on homomorphic encryption and prime numbers*, 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7492759/.

[20] S. M. Anggriane, S. M. Nasution, and F. Azmi, *Advanced e-voting system using paillier homomorphic encryption*, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7905741/.

[21] A. B. H. Ayed, "A conceptual secure blockchain based electronic voting system," *International Journal of Network Security and Applications*, vol. 9, no. 3, pp. 01–09, 2017. DOI: 10.5121/ijnsa.2017.9301.

[22] R. Harerimana, S. Tan, and W. Yau, *A java implementation of paillier homomorphic encryption scheme*, 2017. [Online]. Available: https://doi.org/10.1109/icoict.2017.8074646.

[23] J. Hsiao, R. Tso, C. Chen, and M. Wu, "Decentralized e-voting systems based on the blockchain technology," 2017. [Online]. Available: https://www.researchgate.net/publication/321947971_Decentralized_E-Voting_Systems_Based_on_the_Blockchain_Technology.

[24] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018. [Online]. Available: https://doi.org/10.1145/3214303.

[25] C. Adiputra, R. Hjort, and H. Sato, "A proposal of blockchain-based electronic voting system," in *2018 International Conference on Information Networking (ICOIN)*, IEEE, 2018, pp. 614–619. [Online]. Available: https://ieeexplore.ieee.org/document/8611593.

[26] F. Fusco and et al., *Crypto-voting, a blockchain based e-voting system*, 2018. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220%2F0006962102230227.

[27] D. Khoury, E. F. Kfoury, A. Kassem, and H. Harb, "Decentralized voting platform based on ethereum blockchain," in *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, 2018. DOI: 10.1109/imcet.2018.8603050.

[28] P. G. Naidu and P. R. Mishra, "Blockchain technology architecture and key characteristics," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 3, pp. 188–193, 2018. [Online]. Available: https://ijariie.com/AdminUploadPdf/BLOCKCHAIN_TECHNOLOGY_ARCHITECTURE_AND_KEY_CHARACTERISTICS_ijariie9037.pdf.

[29] M. Yadav and Pooja, "Digital signature," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2018. [Online]. Available: https://tinyurl.com/Digital-Signature-Yadav.

[30] E. Yavuz, A. K. Koc, U. C. Cabuk, and G. Dalkilic, "Towards secure e-voting using ethereum blockchain," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, 2018. DOI: 10.1109/isdfs.2018.8355340.

[31] H. G. Liddell and R. A. Scott, *An Intermediate Greek-English Lexicon: Founded Upon the Seventh Edition of Liddell and Scott's Greek-English Lexicon.* 2019. [Online]. Available: http://ci.nii.ac.jp/ncid/BA00282642.

[32] M. S. U. Miah, M. Rahman, M. S. Hossain, and A. A. Rupai, "Introduction to blockchain," 2019. [Online]. Available: https://www.researchgate.net/publication/343601688_Introduction_to_Blockchain.

[33] L. V.-C. .-. Thuy, K. Cao-Minh, C. Dang-Le-Bao, and T. A. Nguyen, "Votereum: An ethereum-based e-voting system," in *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2019. DOI: 10.1109/rivf.2019.8713661.

[34] R. Tso, Z.-Y. Liu, and J.-H. Hsiao, "Distributed e-voting and e-bidding systems based on smart contract," *Electronics*, vol. 8, no. 4, p. 422, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/4/422.

[35] H. Yi, "Securing e-voting based on blockchain in p2p network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019. DOI: 10.1186/s13638-019-1473-6.

[36] S. Ergenzer, H. Kinklin, and F. Rezabek, *A survey on threshold signature schemes*, 2020. [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2020-11-1/NET-2020-11-1_10.pdf.

[37] V. Holotescu and D. Andone, "Challenges and emerging solutions for public blockchains," *Broad Research in Artificial Intelligence and Neuroscience*, vol. 11, no. 1, pp. 58–83, 2020. [Online]. Available: https://www.researchgate.net/publication/339985291_Challenges_and_Emerging_Solutions_for_Public_Blockchains.

[38] K. M. Khan, J. Arshad, and M. M. Khan, "Investigating performance constraints for blockchain-based secure e-voting system," *Future Generation Computer Systems*, vol. 105, pp. 13–26, 2020. DOI: 10.1016/j.future.2019.11.005.

[39] R. Kumar, T. Yang, and J. Onginjo, "Secure large-scale e-voting system based on ...," *Wiley Online Library*, 2020. DOI: 10.4218/etrij.2019-0362.

[40] D. Pawar *et al.*, “Implementation of secure voting system using blockchain,” *International Journal of Engineering Research & Technology*, vol. 9, no. 06, pp. 141–146, 2020. [Online]. Available: https://www.ijert.org/research/implementation-of-secure-voting-system-using-blockchain-IJERTV9IS060974.pdf.

[41] S. Tasmia Alvi, M. N. Uddin, and L. Islam, *Digital voting: A blockchain-based e-voting system using biohash and smart contract*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9214250.

[42] *A review of distributed access control for blockchain systems towards securing the internet of things*, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9310182.

[43] P. K. Aithal, P. S. Saavedra, S. Aithal, and S. Ghoash, “Blockchain technology and its types-a short review,” 2021. [Online]. Available: https://www.researchgate.net/publication/359051731_Blockchain_Technology_and_its_Types-A_Short_Review.

[44] B. Lashkari and P. Musilek, “A comprehensive review of blockchain consensus mechanisms,” *IEEE Access*, vol. 9, pp. 43 620–43 652, 2021. [Online]. Available: https://www.researchgate.net/publication/350031088_A_Comprehensive_Review_of_Blockchain_Consensus_Mechanisms.

[45] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *Advances in Cryptology — EUROCRYPT ’99*, 2021, pp. 223–238. [Online]. Available: https://doi.org/10.1007/3-540-48910-x_16.

[46] B. Ahn, “Implementation and early adoption of an ethereum-based electronic voting system for the prevention of fraudulent voting,” *Sustainability*, vol. 14, no. 5, p. 2917, 2022. DOI: 10.3390/su14052917.

[47] S. Nakamoto, “A peer-to-peer electronic cash system, bitcoin,” [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[48] Q. Zhang, B. Xu, H. Jing, and Z. Zheng, *Ques-chain: An ethereum based e-voting system*, n.d. [Online]. Available: https://arxiv.org/pdf/1905.05041.pdf.