Journal of Theoretical and Applied Information Technology
15th April 2025. Vol.103. No.7
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN IOT BASED EFFICIENT E-VOTING SYSTEM USING QR CODE AND EFFT-SWIFFT WITH BLOCKCHAIN FOR ENHANCED SECURITY AND TRANSPARENCY

T. PRABAKAR[1], S. KANCHANA[2]

[1]Department of Computer Science, Faculty of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu 603203, India
[2]Department of Computer Science, Faculty of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur 603203, Tamil Nadu, India
Email : [1]patprabakar@gmail.com, [2]kanchans@srmist.edu.in

ABSTRACT

To modernize the democratic voting process, electronic voting systems have appeared as a promising solution. Nevertheless, concerns about the disclosure of sensitive information and the potential compromise of voter privacy might be raised by the transparency of Blockchain (BC)-centric systems. For overcoming these challenges, an Efficient E-Voting System with a QR Code using Blockchain (EVS-QCB) is proposed in this research. To strike a balance between transparency and privacy, the proposed approach incorporates an authority management mechanism. The proposed E-Voting system includes a QR code-based multi-factor authentication mechanism, unlike conventional single-factor authentication system, reducing the risk of credential theft. The system begins with the Voter Registration Phase, where the voters register to the E-Voting system. For selecting relevant attributes from voter details, the system uses a Linear Scaling-based Fox Hunting Optimization Algorithm (LS-FHOA). Next, QR codes and smart contracts are generated and stored in a secure database. Voters authenticate through usernames, passwords, and QR codes during the voting phase, and the system ensures eligibility and prevents duplicate voting utilizing Smart Contracts. Using the Public key raised to the Power of Private key-Elliptic Curve Cryptography (P3-ECC) algorithm, the voting details are encrypted. Utilizing the Exponential Fast Fourier Transformation- Single-Window Iterative Fast Fourier Transform (EFFT-SWIFFT) hashing technique, the encrypted data is transformed into a unique hash code. The generated hash codes are securely stored in BC technology. During the vote counting time, the voting details are fetched from the BC and processed for determining the final results. Performance evaluations demonstrate superior execution time, security, and memory usage compared to baseline models. This work contributes to advancing blockchain-centric e-voting security while addressing authentication and privacy challenges.

Keywords: *Linear Scaling based Fox Hunting Optimization Algorithm (LS-FHOA); Public key raised to the Power of Private Key-Elliptic Curve Cryptography (P3-ECC); Exponential Fast Fourier Transformation- Single-Window Iterative Fast Fourier Transform (EFFT-SWIFFT); Internet of Things (IoT); Electronic- Voting (E-Voting).*

1. INTRODUCTION

Electronic voting (e-voting) has emerged as a transformative innovation in modern elections, leveraging digital technologies to address inefficiencies in traditional voting systems. However, existing e-voting frameworks continue to face critical challenges such as voter authentication vulnerabilities, privacy concerns, and susceptibility to cyber threats. Recent studies have explored blockchain as a promising solution to these issues, offering transparency and decentralization. However, fully transparent blockchain systems may expose sensitive voting data, compromising voter anonymity.

This study aims to address these concerns by introducing an enhanced e-voting model that integrates QR-based multi-factor authentication with blockchain technology.

The proposed EVS-QCB leverages a novel cryptographic framework, ensuring secure storage, authentication, and vote verification while minimizing computational overhead.

Compared to existing works, this approach offers a more efficient and secure solution to address e-voting limitations. The advancement of technology has significantly transformed various sectors, including the electoral process, ushering in a new era of digital innovations [3]. One of these innovations is Electronic Voting (e-voting), which emerged as a viable alternative to traditional paper-centric voting systems [18]. While conventional voting methods have long served as the standard, they present challenges such as lengthy counting times, errors in vote tallying, voter fraud, and logistical difficulties, especially for geographically dispersed electorates [6]. E-voting systems aim to address these issues by enabling voters to cast ballots electronically, promising potential advantages such as increased efficiency, accessibility, and transparency in the electoral process [24]. However, the adoption of e-voting also brings about critical concerns related to ensuring the security, integrity, and trustworthiness of the voting process. Safeguarding against unauthorized access, vote manipulation, and tampering is essential to maintaining the credibility of the electoral system [15].

A fundamental requirement for a secure e-voting system is the preservation of voter privacy. Voters need assurance that their ballots remain confidential throughout the process. To achieve this, cryptographic techniques such as encryption and blind signatures are commonly used to protect individual vote privacy while allowing for verification during the tallying phase [13]. Another critical aspect of e-voting security is voter authentication, which ensures that only eligible voters participate in the election. Robust identification mechanisms, including digital signatures, biometric authentication, and smart contracts, help verify voter eligibility and authenticity, thereby reducing the risk of fraudulent or duplicate votes [22]. Moreover, verifiability is a key feature that allows various stakeholders—such as voters, election officials, and independent auditors—to confirm the accuracy of election results, thereby enhancing transparency, accountability, and trust in the electoral process [23].

To meet these objectives, secure e-voting systems increasingly leverage emerging technologies like Blockchain (BC), which offers a decentralized and tamper-resistant platform for recording and validating votes [10],17]. Blockchain's distributed ledger technology ensures transparency, immutability, and consensus among participants, thus minimizing the risk of data manipulation or unauthorized alterations [20]. Despite their potential, e-voting systems are still susceptible to security threats, such as cyberattacks, hacking attempts, and insider threats, which can compromise the integrity of the electoral process [7], [19]. To address these challenges, robust security protocols, regular audits, and continuous monitoring are crucial to detecting and mitigating vulnerabilities [22].

This study proposes an innovative e-voting system that integrates QR code authentication with Blockchain and IoT technologies to enhance security and resilience. The primary objectives of the proposed system are to mitigate risks associated with traditional e-voting methods and address key security concerns. Specifically, the system aims to combat shoulder surfing and credential theft by implementing QR code-based multi-factor authentication. Additionally, it employs advanced cryptographic techniques like P3-ECC (Elliptic Curve Cryptography) and EFFT-SWIFFT to protect voting data from unauthorized access or tampering, thereby ensuring data integrity. Furthermore, the use of smart contracts enables secure voter registration and eligibility verification, preventing fraudulent participation. The integration of Blockchain technology enhances the transparency and reliability of vote storage and management, while IoT devices facilitate real-time monitoring for improved threat detection. Through this multi-layered approach, the study seeks to set a new benchmark for secure, efficient, and transparent e-voting systems, ensuring a more robust electoral process.

1.1 Problem statement

Existing techniques have a few common drawbacks, which are listed below:

- In some existing methods, single-factor authentication is used without additional security layers, which might lead to shoulder surfing and credential theft.
- The prevailing e-voting systems are not immune to security risks, namely Cyber-attacks, hacking attempts, or insider threats, which pose significant risks to the confidentiality of the voting system.
- Some baseline works had ineffective voter registration databases, which can cause eligible voters to be excluded or ineligible voters to be included.
- Privacy concerns associated with the collection, storage, and potential misuse of voter information are faced by existing works. A challenge that needs to be addressed is striking the right balance between maintaining privacy and ensuring verifiability and transparency.

The research methodology's major objectives are explained below:

- The proposed work addresses the issue of shoulder surfing, where unauthorized individuals can observe a user's authentication credentials, by introducing the QR Code approach for voter authentication.
- The security of voting data is enhanced by the use of the P3-ECC and EFFT-SWIFFT, which protects it from unauthorized access or tampering. These techniques ensure the integrity of the voting data, which protects it from unauthorized access or tampering.
- In the proposed work, a Smart Contract validation technique is used; this allows only eligible voters to participate in the voting process.
- In the proposed e-voting system, the use of BC technology enhances the data storage process. The proposed work improves the security and reliability of the data storage process by leveraging BC.

The work is structured as: the associated study regarding the proposed work is analyzed in section 2, the proposed approach is shown in section 3, the performance of the proposed system is examined in section 4, and lastly, the paper is concluded with the future work in section 5.

2. RELATED LITERATURE SURVEY

Rathee et al.[16], established a secure and transparent E-voting mechanism via IoT devices based on BC technology. Different threats caused by an intruder at various levels were effectively detected and resolved. The propounded method was investigated against different security parameters and considerably surpassed the prevailing approach. But, the scalability problems were not adequately addressed.

Panja & Roy [14] explored a cryptographic approach for an end-to-end verifiable, authenticated, as well as secret ballot election. The ballot-stuffing attack was prevented. Furthermore, secure multi-party computation and Non-Interactive Zero-Knowledge (NIZK) proof were utilized; this ensured the final vote outcomes. The protocol's potential for real-world deployment was exhibited by the experimental data. However, increased communication complexity was enclosed in the approach.

Kumar et al.[11], presented an End-To-End (E2E) verifiable internet voting system, which provided mobility to a voter and allowed the voting process secretly on public computers with the early voting benefit. An identity-centric blind signature scheme, which ensured the voter's anonymity, was utilized. Under the well-known elliptic curve discrete logarithm and gap Diffie-Hellman assumptions, the privacy of the system was attained. Nevertheless, this scheme was more vulnerable to side-channel attacks.

Abuidris et al.[1], deployed a hybrid consensus system composed of Proof of Credibility along with Proof of Stake. It utilized smart contracts, which provided a trustworthy public bulletin board. Also, the ballot outcome's accuracy was ensured by a secure computing environment. A sharding mechanism that improved the BC-centric e-voting system's scalability and performance was used in this approach. The BC-centric e-voting systems' scalability was exposed by the experiments. However, for processing transactions, the scheme needed high computational power; this resulted in higher costs.

Farooq et al.[9], presented an approach, which maintains the voting process's security via BC. The voting transaction was made more secure by the Chain Security Algorithm deployed in the voting system. A secure connection between the user and the network was provided by smart contracts. As per the performance evaluation, the system was effective even in large-scale populations. But, the approach was ineffective in attaining verifiability and privacy.

Ajish & AnilKumar [2], implemented a secure mobile internet voting system, where a biometric technique authenticates the voter. A wavelet-centric Advanced Encryption Standard (AES) algorithm was utilized, which speeded up the encryption process and minimized the mobile device's CPU utilization. According to the experimental analysis, almost all the mobile-centric threats were defeated by the propounded model. However, the security of wavelet-centric AES highly depends on proper key management. The system's overall security could be weakened by inadequate key management.

Li et al.[12], established a variant of anonymous authentication named Event-Oriented Linkable and Traceable Anonymous Authentication (EOLTAA). The decentralized, anonymous, linkable, and publicly traceable voting scheme centered on BC was used, and the double voting was addressed devoid of any aid from other parties. The system's effectiveness regarding time was stated by the outcomes. However, as the number of participants and transactions increased, the scheme's processing speed and capacity were affected.

Yang et al.[25], implemented a voting protocol centered on BC technology. The Homomorphic property of the encryption that guaranteed the voting was utilized. The election's outcomes were made publicly verifiable. The propounded protocol's feasibility for deployment in real elections was demonstrated by the security and performance investigations. However, increased network bandwidth requirements and communication latency were needed.

Alvi et al.[5], presented a BC approach that ensured the security, privacy, and integrity of the voting. Here, voter anonymity was provided centered on the hash function in the BC. Until the election ended, the encrypted casted vote was safeguarded by this approach. Lastly, the casted votes' verifiability was enabled. As per the outcomes, the approach enhanced regarding security characteristics and the associated cost of infrastructure. However, the approach was highly vulnerable to Sybil attacks.

Fan et al.[8], propounded an electronic voting scheme centered on Homomorphic Signcryption (HS). The HS not only enabled the voter to complete the ballot's signature and encryption in one step but also minimized voter signature verification's computational cost by the Authentication Center (AC). Lastly, proof of the voters' privacy and the security and verifiability of voting were provided by this approach. But, increased processing time was caused by the additional computations that were needed for homomorphic operations.

3. PROPOSED E-VOTING SYSTEM

By incorporating an authority management mechanism, the proposed E-Voting method aims in addressing the challenges of transparency, privacy, as well as data tampering in blockchain-based voting systems. Figure 1 displays the proposed framework's architecture.

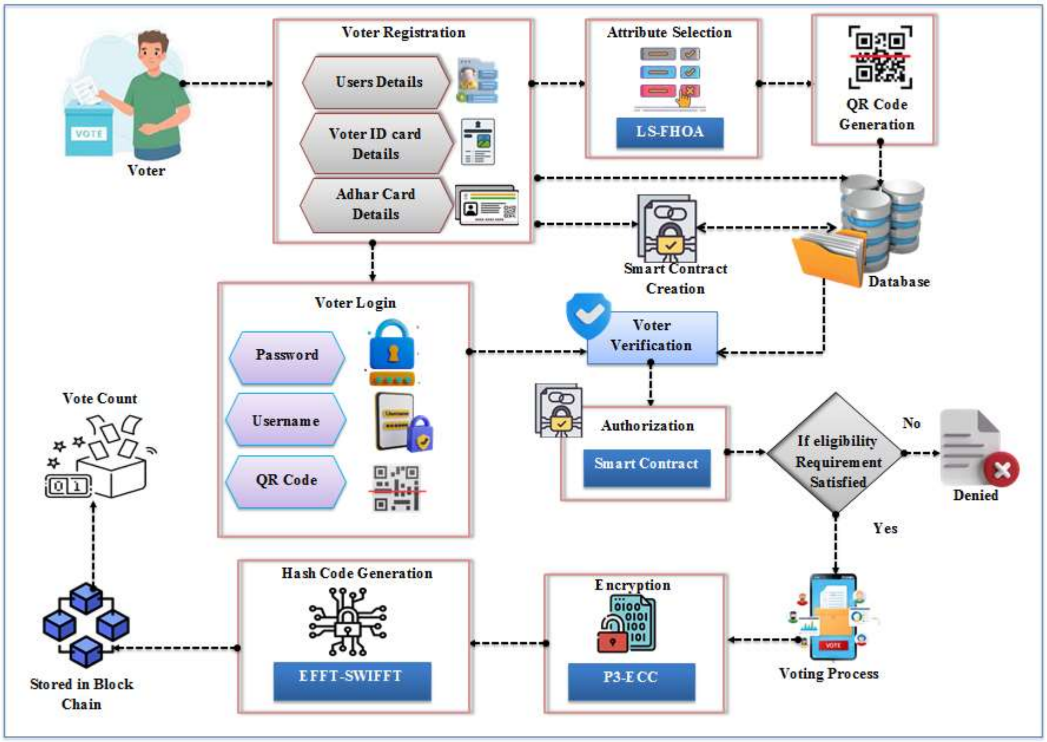

Figure 1: Proposed Architecture

3.1. Voter registration phase

Primarily, for performing secure voting and efficient user authentication through E-Voting, the voter details are collected from the voters. The voters enter their Aadhaar card details $\left(A_{id}\right)$, user details $\left(U\right)$, and voter ID details $\left(V_{id}\right)$. It is described as,

$$\delta = \left\{V_{id}, A_{id}, U\right\} \tag{1}$$

Here, the data entered by the voter is specified as δ .

3.1.1. Attribute Selection

The important attributes (α) related to the voter details are selected from the registered details δ in this phase. In an E-Voting system, the attributes associated with the voter details refer to specific information, namely age, citizenship, address, voting history, et cetera. of an individual voter. Essential details about the voter are provided by these attributes. Attribute selection ensures that for analysis, only the most informative and significant attributes are considered. The system can improve accuracy with minimized computational requirements by eliminating irrelevant or redundant attributes. The proposed method uses LS-FHOA for attribute selection. Owing to the conventional FHOA's efficient exploration and exploitation mechanisms, it exhibits fast convergence rates. However, the optimization process might be hindered by the random process in the FHOA. The algorithm reduces the influence of random fluctuations by replacing the random process with linear scaling in LS-FHOA. This helps to make it more deterministic, providing consistent and reliable outcomes and stabilizing the search process. The steps involved in the procedure are as follows,

Step 1: Initialization

Initialize the fox population. Here, the attributes related to the voter details are represented by the number of foxes, which are mentioned as (α). The fox population is initialized as,

$$\alpha_i = \{\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{i_{Max}}\} \tag{2}$$

Where, the number of foxes (α) in the population is defined as N.

Step 2: Fitness computation

The fitness (Ft) of each fox is calculated; here, the maximum complexity of the QR code $Max(\varsigma(Q_R))$ is termed as the fitness value. This is explained by,

$$Ft = Max(\varsigma(Q_R)) \tag{3}$$

Step 3: Position updation

For each individual fox (α) in the population, the position is updated grounded on its present position, step size, and an LS strategy. The position updation process is determined as,

$$P_{New} = P_{Current} + SS * \chi_s \tag{4}$$

Where, the new position of the fox is defined as P_{New}, the current position of the fox is represented as $P_{Current}$, SS shows step size that determines how far an individual moves from its existing position during the position update process, and a Linear Scaling (LS) vector is specified as χ_s, which helps the procedure to converge faster towards optimal solutions and maintains the relative distances betwixt positions. The χ_s is calculated by,

$$\chi_s = \frac{(R - R_{Min})}{(R_{Max} - R_{Min})} \tag{5}$$

Where, the original value that needs to be scaled is specified as R, the minimum value in the original range is specified as R_{Min}, and the maximum value in the original range is specified as R_{Max}.

Step 4: Weight calculation

After that, based on the fitness values (Ft), the promising weights (pw) are calculated for the updated positions. The formula for calculating pw is,

$$pw = rn\left(\frac{Ft}{q} + \frac{Ft}{10q}\left(rn\left(\frac{q-2i}{2}\right)\right)\right) \quad i = 1,2,3,......, q \tag{6}$$

Where, the number of promising solutions from the group of solutions i is signified as q, and the rounded fitness values are represented as rn.

Step 5: Fitness Update

Recalculate the fitness value of each individual centered on their new positions. Also, update the fitness value of each individual as per the fitness function, the updated fitness is calculated as,

$$Ft = F(P_{New}) \tag{7}$$

Where, the fitness function, which quantifies the quality or performance of an individual's solution in its new position P_{New}, is referred to as F.

Step 6: Selection

Lastly, the individual with the best fitness value is chosen as the present best solution. Therefore, the selected attributes (SA_I) are shown as follows,

$$SA_I = \{SA_1, SA_2, SA_3, \ldots\ldots, SA_{IMax}\} \tag{8}$$

The pseudo-code for the LS-FHOA is given as,

```
Input: Attributes related to voters details $(\alpha)$
Output: Selection of best attributes $(SA)$
Begin
    Initialize the voter's details attributes $(\alpha)$
    Compute the fitness value $(Ft)$
    While $(i = 1 to i_{Max})$ do
        Update the fox position using,
            $P_{New} = P_{Current} + SS * \chi_s$
        Compute the weights for the corresponding position using,
            $pw = rn\left(\frac{AF}{q} + \frac{AF}{10q}\left(rn\left(\frac{q-2k}{2}\right)\right)\right) \quad k = 1,2,3,\ldots\ldots, q$
        Re-compute the fitness value $(Ft)$
        If $(Ft == Satified)$
            End the iteration
        Else
            $i = i + 1$
        End if
            $P_{New} == Ft$
    End while
    Return $(SA)$
End
```

3.1.2. QR Code Generation

The QR code (Q_R) is generated in the e-voting system using the selected attributes (SA). The QR code serves as the selected attribute's compact representation, and during the voter authentication and login process, it is scanned by the system. Several advantages, such as efficient data storage, easy scanning, security, and tamper resistance are offered by the generation of QR codes in the e-voting system, which ensures reliability in transmitting voter attributes in the e-voting system. The Q_R generation process is defined as,

$$Q_R = \Im_{fn}(SA) \tag{9}$$

Here, the function used to encode the selected attributes (SA) into a QR code is represented by $\Im_{fn}$. The generated QR code (Q_R), which serves as a compact representation SA, is the output of this function.

3.1.3. Smart contract creation

The smart contract is generated using the voter's details (δ) in this phase. Establishing a secure and transparent mechanism for governing the voting process is the purpose of generating a smart contract in an E-Voting system. The system can determine if the voter is eligible to participate in the e-voting process by comparing the voter information with the predefined criteria stored in the smart contract. Furthermore, for handling the vote-casting process, the smart contract includes functions and rules. It validates the voter's authenticity and ensures that each voter can cast only one vote. The smart contract maintains the anonymity of the voter and securely records the vote, therefore enabling tamper-proof and a transparent voting process. The details in a smart contract (C_S) are explained by,

$$\mathrm{S}\, C_S \leftarrow \lambda_R(\delta) \tag{10}$$

Where, registered voter details are defined as $\lambda_R(\delta)$.

The predefined criteria (Pc) stored in the (C_S) is,

$$Pc = \left((Age \geq Min_{Age}) \,\&\,\&\, (Citizenship == "Valid") \,\&\,\&\, (RS == "\mathrm{Re}\,gistered")\right) \tag{11}$$

Here, the registration status is represented as RS
.

3.1.4. Database storage process

The smart contract (C_S), QR code (Q_R), and voter registration details (δ) are stored in the database (DB) once the smart contract and the QR code are successfully generated. Storing this information in a database allows for easy access as well as retrieval when required. The data associated with the E-Voting system is securely stored and organized by the database. Mathematically, this process is defined by,

$$C_S + Q_R + \delta \underset{Stored}{\rightarrow} DB \tag{12}$$

3.2. Voter login Phase

The voters can log in to the E-Voting system by providing their username, password, and QR code after completing the voter registration phase. The login process is described as follows,

Username: Voters enter their username into the login page. The username (Un), which serves as their identification within the system, is typically unique to each voter.

Password: A secret code known only to the voter and used for authentication purposes during the login process is termed the password $(Pswd)$.

QR Code Verification: The voter also provides their QR code for verification in addition to the username and password. The QR code comprises encoded information, often comprising the voter's unique identifier.

The voter login details (L_D) are mathematically explained by,

$$L_D = \{Un, Pswd, Q_R\} \tag{13}$$

3.2.1. Voter verification

With the stored user credentials in the database, the E-Voting system verifies the entered username, password, and QR code. The e-voting system allows the voters to access the system if the entered information matches the data stored in the database. The voter verification phase makes sure that only valid as well as authorized voters can gain access. By preventing unauthorized individuals from logging in and participating in the voting process, this verification process improves the integrity as well as security of the voting system. Mathematically, the verification process (Ver) is defined as,

$$Ver = \begin{cases} L_D = \nabla(DB) & Allowed \\ L_D \neq \nabla(DB) & Denied \end{cases} \tag{14}$$

Where, the details stored in the database (DB) are defined as $\nabla(DB)$.

3.2.2. Voter Authorization

Using the smart contract, voter authorization ensures the integrity and validity of the voting process. For evaluating whether a voter is eligible to vote and whether they have already voted, the smart contract applies rules and conditions. This validation process is described as,

Eligibility Criteria (Ec)***:*** The smart contract verifies whether the voter meets the necessary eligibility criteria, which includes checking age, citizenship, registration status, or any other requirements specified by the voting system.

Voting Status Check (Vs) ***:*** The voter's voting status is examined by the smart contract, which verifies whether the voter has already cast their vote in the current election or any other relevant restrictions on multiple voting.

Additional Checks (Ac)***:*** The smart contract incorporates additional checks depending on the specific requirements of the voting system. This can comprise verifying if the voter is within the designated voting period or any other essential verification.

Authorization Decision (Ad) ***:*** Voters are granted authorization (permission) to proceed with voting if they pass all the checks and meet the requirements. The smart contract denies authorization if any of the checks fail. This Ad process is explained as,

$$Ad = \begin{cases} IF\ Ec \oplus Vs \oplus Ac = E & THEN\ Permission\ Granted \\ IF\ Ec \oplus Vs \oplus Ac \neq E & THEN\ Permission\ Denied \end{cases} \tag{15}$$

Where, the eligibility requirements are indicated as E.

3.2.3. Voting process

The actual voting process takes place in the E-Voting system after the voter authorization process.

The voter selects their preferred candidate from the list of candidates in the voting process. After that, via the E-Voting system, the voter submits their vote electronically. The voter's choices and the voter's identification are securely recorded by the system. The voting details are mentioned as (X_{vote}).

3.2.4. Encryption

Then, for ensuring the confidentiality and privacy of the votes, the voting details (X_{vote}) are encrypted. The E-Voting system adds an additional layer of security and confidentiality to the voting process by encrypting the voting details. The integrity of the votes is safeguarded by E-Voting, which helps maintain voter privacy, ensuring that only authorized individuals with the proper decryption keys can access as well as interpret the voting information. The encryption is performed by using the P3-ECC algorithm in this work. ECC needs fewer bits for encryption as well as decryption, leading to lower bandwidth consumption. However, the ECC implementations are vulnerable to various types of attacks. Therefore, an additional secret key is used to enforce strong security. By using the value of the Public key, which is raised to the Power of Private keys (PPP) (P3), the secret key is generated and then added during encryption and subtracted during decryption. Therefore, an extra layer of protection is provided by the inclusion of an additional secret key. The P3-ECC procedure is described as follows,

Firstly, the elliptic curve is explained over a finite field with parameters. The formula for the elliptic curve is,

$$p^2 = q^3 + xp + y \tag{16}$$

Where, prime numbers representing the field size are specified as p and q, and constants defining the elliptic curve are specified as x and y.

Key Generation: The public key (Pb), private key (Pv), and secret key (γ) are generated. With the receiver's public key, the voting details are encrypted, and the receiver decrypts the original data by utilizing its private key. The Pb is generated by,

$$Pb = Pv * \ell \tag{17}$$

$$Pv = Random(\rho) \tag{18}$$

$$\gamma = Pb^{Pv} \tag{19}$$

Where, Pb^{Pv} represents that the Public Key is raised to the Private Key's power, the point on the curve is represented as ℓ, and private key, which is generated randomly from the value of ρ is referred as Pv; $\rho = [1-(n-1)]$, which specifies the range defined by the order of the elliptic curve.

Encryption: The voting details (X_{vote}) have a point $(\hbar)$ on the curve (v). The value u is randomly selected from $[1-(n-1)]$. The two cipher data C_1 as well as C_2 are generated as follows,

$$C_1 = u * \ell \tag{20}$$

$$C_2 = \hbar + u * Pb + \gamma \tag{21}$$

Hence, the encrypted voting data is called as En_{vote}. The pseudo-code for the P3-ECC algorithm is exhibited below:

```
Input: Voting details $(X_{vote})$
Output: Encryption and decryption of $(X_{vote})$
Begin
        Initialize the voting details $(X_{vote})$
        Define the elliptic curve using,
                $p^2 = q^3 + xp + y$
        Generate private keys randomly
        Generate public key using,
                $Pb = Pv * \ell$
        Generate secret key using,
                $\gamma = Pb^{Pv}$
        For each data do
                //Encryption
                        $C_1 = u * \ell$
                        $C_2 = \hbar + u * Pb + \gamma$
                //Decryption
```

```
            $\hbar = C_2 - Pv * C_1 - \gamma$
    End for
End
```

3.2.5. Hash code generation

The En_{vote} is converted into a hash code after the encryption process. The hash code serves as the encrypted data's unique representation. It is utilized for verifying the encrypted data's integrity by analogizing the calculated hash code with the expected hash code. The encrypted data is not tampered with or modified if the hash codes match. Hence, the hash code generation phase assists in verifying the encrypted voting data's integrity without revealing any sensitive information. This research approach used the EFFT-SWIFFT method for producing a unique hash value. Despite the input data size, SWIFFT produces hash codes of fixed sizes. This property is beneficial for applications with limited storage capacities as it allows for efficient storage and retrieval of hash codes. But, during the FFT computation, numerical inaccuracies can occur. Thus, for sorting out this issue, the EFFT algorithm is designed to reduce the errors. EFFT can provide higher accuracy in frequency analysis and transform operations by using exponential functions and more precise calculations. The steps of EFFT-SWIFFT are detailed below:

Step 1: Primarily, the encrypted data (En_{vote}) is converted into a sequence of numerical values $\{y_1, y_2, y_3, \ldots\ldots, y_\eta\}$, which is indicated as,

$$En_{vote} = \{y_1, y_2, y_3, \ldots\ldots, y_\eta\} \tag{22}$$

Step 2: Next, the converted data is partitioned into fixed-size windows or blocks. When analogized to operating on the entire dataset, processing smaller blocks of data is more computationally effective. This enhances the SWIFFT hashing process's overall efficiency and speed. The fixed-size windows are signified as,

$$Wn_1 = \{y_1, y_2, y_3, \ldots\ldots., y_K\} \tag{23}$$

$$Wn_2 = \{y_{K+1}, y_{K+2}, y_{K+3}, \ldots\ldots., y_{2K}\} \tag{24}$$

Step 3: Subsequently, the Window Transformation is implemented. This involves deploying the Fast Fourier Transform (FFT) algorithm to each window of data. The FFT algorithm converts the input data from the time-domain representation to the frequency-domain representation. However, the FFT is more susceptible to numerical inaccuracies. Thus, EFFT is used in this work. The Window Transformation utilizing the EFFT algorithm is given below,

$$Y = EFFT(Z) \tag{25}$$

Here, the input window of data is indicated by Z, and the transformed values after applying the EFFT algorithm are represented by Y. The formula for $EFFT$ is expressed as,

$$Y[J] = \sum Z[K] * Exp\left(\frac{-2\pi imJK}{l}\right) \quad J = 0\, to\, l-1 \tag{26}$$

Here, the transformed values in the frequency domain are signified as $Y[J]$, the input sequence of values in the time domain is indicated as $Z[K]$, frequency index and time index are defined as J and K values, correspondingly, the input sequence's length is given as l, and the imaginary unit is expressed as im.

Step 4: The transformed values (vl) from each window are combined utilizing XOR operations $(\oplus)$ after the window transformation, and it is described as,

$$G_{vl} = vl_1 \oplus vl_2 \oplus vl_3 \oplus \ldots.. \oplus vl_{\max} \tag{27}$$

Here, the combined value is defined as G_{vl}.

Step 5: Lastly, to generate the final hash code, a hash function is deployed to the combined value (G_{vl}); this process is described by,

$$Hc = Hf(G_v) \tag{28}$$

Here, the output of the hash function is depicted as Hc, and Hf refers to the hash function. Hence, Hc_B are mathematically described as,

$$Hc_B = \{Hc_1, Hc_2, Hc_3, ..., Hc_b\} \tag{29}$$

3.2.6. Blockchain

The generated hash codes Hc_B are subsequently stored in the BC, which is a distributed and decentralized network of blocks that collectively maintain the data's integrity and security. The information becomes replicated across multiple blocks in the network by storing hash codes in the BC, which makes it highly resilient against data loss or unauthorized modifications. Figure 2 displays the BC structure.

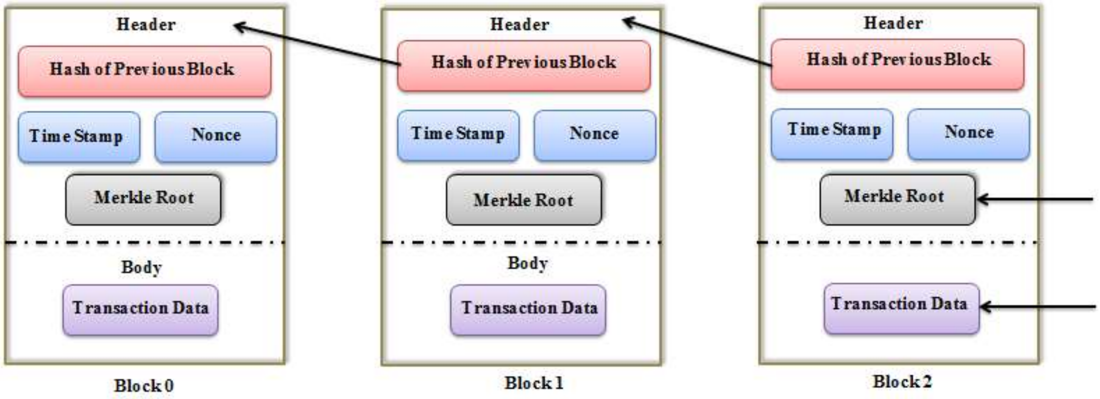

Figure 2: Blockchain Structure

Blockchain's working process is described below:

Block Formation: Blockchain encompasses a chain of blocks. Transactions are grouped together into blocks. Every single block typically possesses a fixed size and can store a certain number of transactions.

Block Header (BH)***:*** Every single block has a header, which comprises significant information, encompassing the hash of the previous block (H_{pr}), a timestamp (T), nonce (Nc), and other relevant data like Merkle root $(\aleph)$. The link is created between blocks by the previous block's hash, forming the chain. This is mathematically defined as,

$$BH \leftarrow H_{pr} + T + Nc + \aleph \tag{30}$$

Hash Function: A hash function (Hf) is deployed to the data in each block, which generates a unique hash value (Hv) that serves as a digital fingerprint of the block's data (B_d). Hv is formulated by,

$$Hv = Hf(B_d) \tag{31}$$

Proof-of-Work (PoW): PoW is a mechanism utilized in BC systems for validating and adding new blocks. The PoW involves finding a nonce value that is merged with the block's data and yields a hash value, which satisfies specific criteria. The formula for PoW is given below:

$$H(B_d + Nc) \le Tg \tag{32}$$

Here, an arbitrary number that adjusts to find a valid hash is signified as Nc, and a value that determines the difficulty of the PoW puzzle is indicated as Tg.

Consensus Mechanism (ζ)***:*** The PoW puzzle broadcasts the new block to the network once it is successfully solved. Other participants in the network validate the block, and if it passes the validation checks (vc), they add it to the BC. Mathematically, this process is defined as,

$$\zeta = \begin{cases} IF\ vc == satisfied & THEN\ Blocks\ added \\ IF vc \ne satisfied & THEN\ Blocks\ ignored \end{cases} \tag{33}$$

Blockchain Validation: The hash of each block and its linkage to the previous block is verified for ensuring the BC's integrity. Furthermore, the PoW solution and the correctness of the transactions enclosed in the block are validated.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2.7. Vote count

Lastly, the encrypted voting details (En_{vote}) are retrieved from the BC. Next, to read the original voting details, the decryption process is done. The formula for the decryption process is expressed as,

$$\hbar = C_2 - Pv * C_1 - \gamma \tag{34}$$

For determining the vote count for each candidate, the decrypted voting details (Dc_{vote}) are utilized. The system performs verification checks for ensuring the vote count's accuracy. The system cross-checks the decrypted voting details against the original encrypted data stored in the BC. The data integrity is verified by analogizing the hash codes generated during the encryption phase with the decrypted data. The hash code verification process is described as,

$$Hc(Dc_{vote}) == Hc(En_{vote}) \tag{35}$$

At last, the final results are calculated by the system. Hence, the proposed model shows the potential to safeguard privacy and mitigate data tampering in electronic voting processes.

4. RESULTS AND DISCUSSION

This section evaluates the proposed approach's performance, and in the working platform of PYTHON, the experiments were done.

4.1. Performance Analysis

The proposed EVS-QCB's performance is validated in this section. For validating the proposed system's efficiency, the performance analysis and the comparative analysis take place here.

Table 1: Evaluation Of The Proposed Method In Terms Of Execution Time

Processes	Execution Time (ms)
Smart Contract Creation Time	747
QR Code Generation Time	1310
Voter Verification Time	400
Smart Contract Authorization time	806

The time taken to perform different distinct processes in the proposed E-Voting system is evaluated in Table 1. The time taken to create a smart contract, generate a QR code, verify a voter's identity, and authorize or validate a smart contract are 747 ms, 1310 ms, 400 ms, and 806 ms, respectively. Reasonable and efficient execution times for its distinct processes are demonstrated by the proposed E-Voting system. Hence, the proposed approach ensures various tasks' timely execution in an E-Voting system.

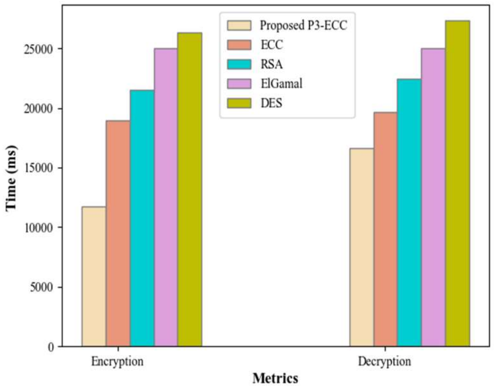

Figure 3: Comparative Analysis Of The Proposed P3-ECC

The performance of the proposed P3-ECC and the prevailing studies, such as ECC, ElGamal, Rivest, Shamir, Adleman (RSA), and Data Encryption Standard (DES) regarding encryption and decryption time are compared in Figure 3. The proposed technique shows faster arithmetic operations. In general, these operations are faster compared to the modular exponentiations needed in prevailing encryption standards. Hence, the proposed technique needs 11771 ms and 16635 ms to encrypt and decrypt the data, correspondingly. This is relatively lower compared to the prevailing approaches.

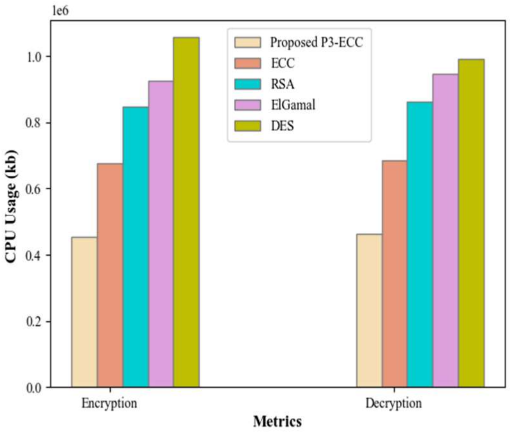

Figure 4: Memory usage comparison

A comparative analysis of the proposed and the existing approaches regarding memory usage is given in Figure 4. For encryption and decryption, the proposed P3-ECC needs 454758 kb and 463358 kb of memory, respectively. However, the existing approach needs an average of 876836 kb for the encryption and 871598 kb for the decryption process. This is due to the P3-ECC generates shorter cipher texts. During encryption and decryption, shorter cipher texts need fewer bits to be stored in memory, which leads to reduced memory needs.

Table 2: Security And Attack Level Analysis

Techniques	Security Level (%)	Attack Level (%)
Proposed P3-ECC	98	2.45
ECC	94	6.14
RSA	91	9.014
ElGamal	90	10.2747
DES	89	11.23541

The security and attack levels of the proposed and the prevailing techniques are evaluated in Table 2. The proposed technique adds an additional layer of security to the algorithm using the secret key. Thus, the proposed technique withstands a high-security level of 98% and a low attack level of 2.45%. However, the existing approaches remain with an average security level of 91% and an attack level of 9.16%. Hence, the proposed P3-ECC is more secure and less susceptible to external attacks.

Table 3: Evaluation Of Proposed EFFT-SWIFFT With Respect To Hash-Code Generation Time

Techniques	Hash-code Generation Time (ms)
Proposed EFFT-SWIFFT	530
SWIFFT	659
MD5	697
SHA512	720
SHA256	820

The Hash-code generation time of the proposed EFFT-SWIFFT and the baseline hashing algorithm, such as SWIFFT, Message Digest-5 (MD5), Secure Hash Algorithm (SHA512), and SHA256 are evaluated in Table 3. The EFFT-SWIFFT generates the hash code at 530 ms. However, to generate the hash code, the baseline approaches need an average of 724 ms. This is because the EFFT-SWIFFT is designed to reduce the computation time overhead. This reduction in computation time contributes to faster hash code generation.

Table 4: Hash collision rate analysis

Techniques	Hash collision rate (%)
Proposed EFFT-SWIFFT	0.09
SWIFFT	0.2
MD5	0.31
SHA512	0.4
SHA256	0.6

The hash collision rate of the proposed EFFT-SWIFFT and the conventional hashing algorithm is depicted in Table 4. The numerical inaccuracies are effectively minimized by the proposed EFFT function. The EFFT algorithm improves the uniqueness of the hash codes it generates by reducing errors in the frequency analysis process. Hence, the proposed EFFT-SWIFFT withstands a minimal collision rate of 0.09%. However, the prevailing algorithm remains with an average hash collision rate of 0.37%. Hence, the probability of collision in the proposed approach is lower than the prevailing works.

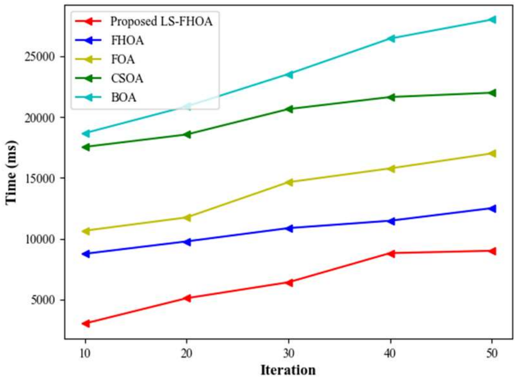

Figure 5: Attribute Selection Time Comparison

Attribute selection time for the proposed LS-FHOA and the prevailing algorithms, such as FHOA, Firefly Optimization Algorithm (FOA), Cuckoo Search Optimization Algorithm (CSOA), and Bayesian Optimization Algorithm (BOA) are presented in Figure 5. Here, the X-axis represents a number of attributes (10, 20, 30, 40, and 50). The values in the Y-axis correspond to the time taken by each algorithm to execute attribute selection for the specified number of attributes. While there are 10 attributes, the attribute selection time for the proposed LS-FHOA is 3021 ms, and the prevailing FHOA, FOA, CSOA, and BOA are 8756 ms, 10656 ms, 17546 ms, and 18675 ms, respectively. The comparisons show that the attributes with the least amount of time are selected by the proposed LS-FHOA. The LS-FHOA algorithm is made more efficient and effective in finding the most appropriate attributes by the elimination of randomness.

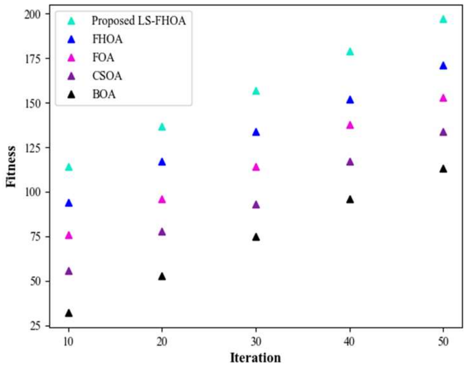

Figure 6: Fitness Vs. Iteration Comparison.

The optimization ability of the proposed LS-FHOA and existing algorithms is compared utilizing fitness vs. iteration in Figure 6. The proposed LS-FHOA's local search strategies and elimination of randomness help it avoid getting stuck in local optima, which ensures that it explores a broader solution space to find more optimal attributes. At 10 iterations, the fitness value for the proposed LS-FHOA is 114, and the existing approaches achieve the fitness of 94 for FHOA, 76 for FOA, 56 for CSOA, and 32 for BOA. Hence, when analogized to prevailing approaches, the proposed LS-FHOA shows faster convergence.

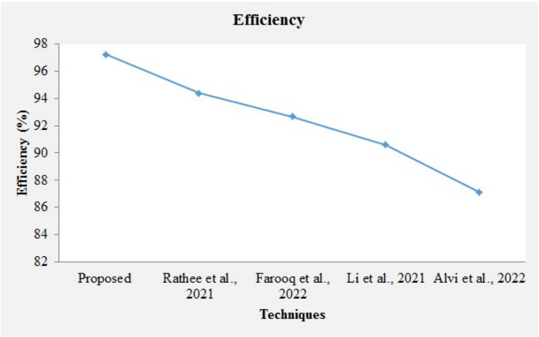

Figure 7: Efficiency Comparison With Related Work

The efficiency of the proposed and the prevailing research works are analogized in Figure 7. In the proposed study, the voting process is streamlined more securely by the use of QR codes and smart contracts. Furthermore, the proposed approach incorporates BC technology. Data immutability and transparency are ensured by storing hash codes in the BC. The proposed system improves the trustworthiness and integrity of the voting data by leveraging BC technology. Hence, the proposed technique remains with an efficiency of 97.24%. Figure 8 concludes that the proposed technique was more effective than the prevailing techniques.

Previous studies have explored blockchain-based e-voting mechanisms, but they lack robust multi-factor authentication and suffer from scalability challenges. The proposed EVS-QCB integrates QR code-based multi-factor authentication with an advanced cryptographic framework comprising P3-ECC and EFFT-SWIFFT techniques. This ensures enhanced security, reduced computational overhead, and improved verifiability, setting it apart from existing methods.

5. CONCLUSION

An efficient and secure solution to modern electronic voting is provided by the proposed E-Voting system. The proposed study establishes a robust and reliable platform for conducting secure elections by addressing the challenges of BC-centric voting systems via authority management, optimized attribute selection, and advanced cryptographic techniques. The proposed study's performance is validated by the experimentation analysis which addresses key limitations of existing models, offering improved security through P3-ECC encryption and EFFT-SWIFFT hashing while ensuring voter authentication integrity.. The proposed technique withstands 98% of the security level and 9.16% of the attack level. Furthermore, the times recorded for smart contract creation, QR code generation, voter verification, and smart contract authorization all fall within acceptable ranges for effective functioning. Overall, the prevailing approaches are outperformed by the proposed technique, which remains to be more reliable and robust. However, network congestion and computational scalability remain challenges requiring further optimization. The parallel processing of transactions in the BC network will be enabled by this work in the future. The system can handle a higher number of transactions during peak voting periods by using multiple processing nodes simultaneously.

REFERENCES

[1] Abuidris, Y., Kumar, R., Yang, T., & Onginjo, J. (2021). Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding. ETRI Journal, 43(2), 357–370. https://doi.org/10.4218/etrij.2019-0362

[2] Ajish, S., & AnilKumar, K. S. (2021). Secure mobile internet voting system using biometric authentication and wavelet based AES. Journal of Information Security and Applications, 61, 1–12. https://doi.org/10.1016/j.jisa.2021.102908

[3] Alotaibi, A., Alhubaidi, L., Alyami, A., Marghalani, L., Alharbi, B., & Nagy, N. (2022). Preventing Phishing Attack on Voting System Using Visual Cryptography. Journal of Computer and Communications, 10(10), 149–161. https://doi.org/10.4236/jcc.2022.1010010

[4] Alvi, S. T., Uddin, M. N., & Islam, L. (2020). Digital voting: A blockchain-based E-voting system using biohash and smart contract. Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, 228–233. https://doi.org/10.1109/ICSSIT48917.2020.9214250

[5] Alvi, S. T., Uddin, M. N., Islam, L., & Ahamed, S. (2022). DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system. Journal of King Saud University - Computer and Information Sciences, 34(9), 6855–6871. https://doi.org/10.1016/j.jksuci.2022.06.014

[6] Baudier, P., Kondrateva, G., Ammi, C., & Seulliet, E. (2021). Peace engineering: The contribution of blockchain systems to the e-voting process. Technological Forecasting and Social Change, 162, 1–11. https://doi.org/10.1016/j.techfore.2020.120397

[7] Çabuk, U. C., Adıgüzel, E., & Karaarslan, E. (2018). A Survey on Feasibility and Suitability of Blockchain Techniques for the E-Voting Systems. International Journal of Advanced Research in Computer and Communication Engineering, 7(3), 124–134. https://doi.org/10.17148/ijarcce.2018.7324

[8] Fan, X., Wu, T., Zheng, Q., Chen, Y., Alam, M., & Xiao, X. (2020). HSE-Voting: A secure high-efficiency electronic voting scheme based on homomorphic signcryption. Future Generation Computer Systems, 111, 754–762. https://doi.org/10.1016/j.future.2019.10.016

[9] Farooq, M. S., Iftikhar, U., & Khelifi, A. (2022). A Framework to Make Voting System Transparent Using Blockchain Technology. IEEE Access, 10, 59959–59969. https://doi.org/10.1109/ACCESS.2022.3180168

[10] Khan, K. M., Arshad, J., & Khan, M. M. (2021). Empirical analysis of transaction malleability within blockchain-based e-Voting. Computers and Security, 100, 1–22. https://doi.org/10.1016/j.cose.2020.102081

[11] Kumar, M., Chand, S., & Katti, C. P. (2020). A Secure End-to-End Verifiable Internet-Voting System Using Identity-Based Blind Signature. IEEE Systems Journal, 14(2), 2032–2041. https://doi.org/10.1109/JSYST.2019.2940474

[12] Li, P., Lai, J., & Wu, Y. (2021). Event-oriented linkable and traceable anonymous authentication and its application to voting. Journal of Information Security and Applications, 60, 1–10. https://doi.org/10.1016/j.jisa.2021.102865

[13] Mukherjee, P. P., Boshra, A. A., Ashraf, M. M., & Biswas, M. (2020). A Hyper-ledger Fabric Framework as a Service for Improved Quality E-voting System. 2020 IEEE Region 10 Symposium, TENSYMP 2020, 394–397. https://doi.org/10.1109/TENSYMP50017.2020.9230820

[14] Panja, S., & Roy, B. (2021). A secure end-to-end verifiable e-voting system using blockchain and cloud server. Journal of Information Security and Applications,

59(April), 1–25. https://doi.org/10.1016/j.jisa.2021.102815

[15] Pawlak, M., & Poniszewska-Marańda, A. (2021). Trends in blockchain-based electronic voting systems. Information Processing and Management, 58(4), 1–23. https://doi.org/10.1016/j.ipm.2021.102595

[16] Rathee, G., Iqbal, R., Waqar, O., & Bashir, A. K. (2021). On the Design and Implementation of a Blockchain Enabled E-Voting Application within IoT-Oriented Smart Cities. IEEE Access, 9, 34165–34176. https://doi.org/10.1109/ACCESS.2021.3061411

[17] Satizábal, C., Páez, R., & Forné, J. (2022). Secure Internet Voting Protocol (SIVP): A secure option for electoral processes. Journal of King Saud University - Computer and Information Sciences, 34(6), 3647–3660. https://doi.org/10.1016/j.jksuci.2020.12.016

[18] Shankar, A., Pandiaraja, P., Sumathi, K., Stephan, T., & Sharma, P. (2021). Privacy preserving E-voting cloud system based on ID based encryption. Peer-to-Peer Networking and Applications, 14(4), 2399–2409. https://doi.org/10.1007/s12083-020-00977-4

[19] Sheela, A. C. S., & Ramya, G. F. (2021). E-voting system using homomorphic encryption technique. Journal of Physics: Conference Series, 1770(1), 1–10. https://doi.org/10.1088/1742-6596/1770/1/012011

[20] Sherine, A., Peter, G., Stonier, A. A., Leh Ping, D. W., Praghash, K., & Ganji, V. (2022). Development of an Efficient and Secured E-Voting Mobile Application Using Android. Mobile Information Systems, 2022, 1–11. https://doi.org/10.1155/2022/8705841

[21] Taş, R., & Tanriöver, Ö. Ö. (2021). A Manipulation Prevention Model for Blockchain-Based E-Voting Systems. Security and Communication Networks, 2021, 1–16. https://doi.org/10.1155/2021/6673691

[22] Taş, R., & Tanrıöver, Ö. Ö. (2020). A systematic review of challenges and opportunities of blockchain for e-voting. Symmetry, 12(8), 1–24. https://doi.org/10.3390/sym12081328

[23] Tejedor-Romero, M., Orden, D., Marsa-Maestre, I., Junquera-Sanchez, J., & Gimenez-Guzman, J. M. (2021). Distributed remote e-voting system based on shamir's secret sharing scheme. Electronics (Switzerland), 10(24), 1–19. https://doi.org/10.3390/electronics10243075

[24] Yang, X., Yi, X., Kelarev, A., Han, F., & Luo, J. (2021). A distributed networked system for secure publicly verifiable self-tallying online voting. Information Sciences, 543, 125–142. https://doi.org/10.1016/j.ins.2020.07.023

[25] Yang, X., Yi, X., Nepal, S., Kelarev, A., & Han, F. (2020). Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities. Future Generation Computer Systems, 112, 859–874. https://doi.org/10.1016/j.future.2020.06.051