Secure Blockchain-Based Electronic Voting Mechanism

Pin-Chang Su
Department of Information Management, National
Defense University, Taiwan
spc.cg@msa.hinet.net

Tai-Chang Su
Department of Information Management, National Defense
University, Taiwan
believe50405@gmail.com

Abstract: *Many countries have strived to popularise electronic voting (e-voting), but owing to various security concerns, large-scale elections are still invariably held using paper ballots. Electronic voting systems must find solutions to various issues with authentication, data privacy and integrity, transparency, and verifiability. On the other hand, Blockchain technology offers an innovative solution to many of these problems. In this study, we constructed a private blockchain network with a large number of nodes, which is only accessible to the relevant voters. Because of its decentralised design, the system is robust against attacks by malicious actors. The security of the system was enhanced using an elliptic curve discrete logarithm problem-based blind multi-document signcryption mechanism. As this mechanism can be used to blindly sign and encrypt multiple voting documents in a single pass, it will minimise redundant signing processes and thus improve efficiency. Furthermore, a self-certification mechanism was used in lieu of centralised certificate servers, so that the voters can participate in the computation of public and private keys. In summary, we designed an electronic voting mechanism that is sufficiently secure for practical purposes, which will improve trust in e-voting, and reduce the costs associated with vote checking.*

Keywords: *Blockchain, e-voting, blind multi-document signcryption, self-certification.*

Received July 21, 2021; accepted November 7, 2022

https://doi.org/10.34028/iajit/20/2/12

1. Introduction

Owing to advances in digital and Internet technologies, it is now possible to conduct conversations, discussions, and even voting through the Internet to make group decisions. However, most large-scale elections are still conducted using paper ballots. In addition to being labour-intensive, paper voting makes it difficult for voters who are working or studying overseas to participate in voting processes. These problems can be solved using electronic voting (e-voting). For instance, the 2020 U.S. presidential election was conducted using paper ballots and e-voting machines. Although these machines are more cost- and labour-efficient than paper ballots, voters were required to submit their votes in person. [17] Furthermore, e-voting machines are costly to purchase and maintain. Therefore, an Internet-based e-voting mechanism would be far more convenient for voters.

Any e-voting system that is being used to replace paper ballots will inevitably attract concerns regarding security. To address these, blockchain technology, which is the basis of the Bitcoin cryptocurrency [16], can be employed. In essence, blockchains are decentralised databases that use a consensus mechanism to record a ledger of the data they store. Because a blockchain is collectively stored in multiple nodes, any attempt to manipulate its data will be immediately detected. The decentralised, immutable, and anonymous nature of blockchains can be used to ensure transparency in voting processes. A blockchain can be categorised as a public, private, or consortium chain, depending on its level of centralisation. Because the proposed e-voting mechanism is only meant for voters and the objective of this study was to create a relatively cost-efficient, safe, and fast blockchain mechanism, private blockchains were deemed to be optimal for our purposes.

Blind Signatures was first proposed by Chaum [4], who presented a blind Rivest Shamir Adleman (RSA) signing scheme that allows a signer to sign a message from a message author without revealing its contents, ensuring privacy. However, this scheme has security problems pertaining to message integrity, untraceability, and non-repudiation. Camenisch *et al*. [3] proposed a blind signing framework based on the Discrete Logarithm Problem (DLP), whose security lies in the difficulty of solving the DLP. However, Harn [10] showed that the framework of Camenisch *et al*. cannot be used to achieve untraceability. Horster *et al*. [11] then argued that the proof of Harn is incorrect. Lee *et al*. [14] showed that the comments of Horster *et al*. [11] were improper; they also designed another DLP-based blind signature scheme that ensures untraceability. Jeng *et al*. [12] proposed a blind signature scheme based on Elliptic-Curve Cryptography (ECC), which is capable of achieving untraceability while being computationally efficient. In a departure from conventional e-signature algorithms, Zheng [25] proposed a cryptographic technique called signcryption, which combines the functions of digital signatures and symmetric encryption. This approach provides the security functions of signatures and encryption while being significantly more efficient than DLP-based signature-then-encryption approaches. Yu [24] proposed a blind signcryption scheme combining encryption with blind signatures, which allows documents to be transferred in a secure and untraceable manner. However, this scheme does not exhibit the avalanche effect, which limits its security and efficiency. To address this weakness, Su et al. created an ECC-based blind multi-document signcryption mechanism [20] that scrambles all documents into ciphertext using a knapsack problem-based signcryption algorithm, giving rise to the avalanche effect without increasing the computational complexity. This scheme also significantly reduces the number of signing instances, thus improving the efficiency while ensuring security.

Self-certified public-key cryptography was introduced by Girault [9], with the aim of allowing users to participate in the computation of public and private keys during the authorisation stage. Thus, the users can independently certify their identities without identity verification by a certificate authority, which addresses the weaknesses of other identity-verification methods. This provides a high level of security, reduces the key-management burden, and allows for fast identity verification. In the present work, an ECC-based self-certification mechanism (instead of the RSA-based scheme of Su *et al*. [21]) was used to design an identity-verification scheme for all the roles in our e-voting system. A formal logic analysis was then performed using the Burrows–Abadi–Needham logic (BAN logic) method [1] to validate the completeness of the proposed mechanism.

Electronic voting is becoming a more essential and widespread issue in the context of blockchain and information systems. The unique characteristics of this technology, such as decentralization and immutability, were crucial in ensuring that the voting system followed the same norms as more conventional elections and voting fields. Democracy is founded on voting and will not work well if people do not trust the voting system [15]. This analysis gives our rise to an alternative kind of cryptosystem. In this work, our aim is to highlight the security concerns of the existing blockchain electronic voting protocol. Relying on its transparency, decentralization, verifiability and other characteristics, the trusted third parties become replaceable, and the voters' level of trust in the mechanism can be enhanced. In addition, the blind signature mechanism with the complexity as the elliptic curve discrete logarithmic problem is used to strengthen the security features related to electronic voting. Last but not least, the self-certification mechanism is introduced to replace the centralized certificate authority, so that the voters can calculate the public and private keys by themselves to alleviate the concerns of impersonation by the certificate authority as a trusted third party. For verifying such a design, the BAN-Logic and several security features are used to prove that it serves as a mechanism with sufficient security, and that it enhances the voters' level of trust in electronic voting. The extra costs for re-verifying the voting results due to voters' distrust are therefore reduced. The paper is organized as follows. In the next section, we briefly introduce blockchain technology, electronic voting mechanism and electronic signatures. In section 3, we propose an original essay to construct an electronic voting scheme for blind digital signatures. In section 4, we evaluate the performance of the proposed solution and prove its security features. Finally, section 5 describes concluding the paper.

2. Literature Review

In this section, we present an extensive review of the research literature on blockchain, e-voting, and e-signature technologies, to establish a foundation for this study.

2.1. Blockchain Technology

In recent years, blockchain technology has become very popular and been used in different domains, such as healthcare, IoT, supply chain, etc., [5, 8, 18]. The concept of a smart contract was proposed by prolific cross-disciplinary legal scholar Nick Szabo, who defined it as follows: 'A smart contract is a set of promises defined in digital form, including agreements on which contract participants can execute these commitments'. Smart contracts are specified in algorithmic form and executed using computers; thus, they can be safer than conventional contracts. Therefore, smart contracts reduce contract-related transaction costs [22]. In 2013 Buterin [2] the inventor of Ethereum-published a white paper about Ethereum called 'A Next-Generation Smart Contract and Decentralized Application Platform'. This paper states that Ethereum allows users to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. Wood [23] further noted that Ethereum is a decentralised blockchain-based cryptocurrency that can be used to execute any code within a smart contract. In the Ethereum system, the Ethereum Virtual Machine (EVM) is a closed environment where all contract code is executed in isolation from the outside world. In this environment, it is possible to restrict access rights between different smart contracts. In Ethereum blockchains, smart contracts are written using the Solidity language and then compiled into code that can be executed by the EVM. In summary, Ethereum is a full-featured blockchain application platform that can be fully isolated from the outside world, which has well-rounded development tools. For these reasons, we chose to design our secure e-voting mechanism using Ethereum private blockchains and smart contracts.

2.2. E-Voting

As its name suggests, e-voting is defined as the use of electronic devices (instead of paper ballots) for voting. According to Khan *et al*. [13], e-voting refers to the use of e-voting machines for voting. As this approach uses expensive voting machines and requires the voter to cast their votes in person, it is not relevant to the present study. In contrast, Internet voting (I-voting) refers to the use of the Internet to facilitate voting processes. I-voting does not require voters to be present at polling stations. Instead, the voter uses a suitable electronic device (a personal computer, tablet, or smartphone) to connect to the voting system, verify his/her identity by inputting their ID and password, and then submit his/her vote once the system has confirmed eligibility (e.g., the voter is not attempting to vote twice). The vote is then transmitted in encrypted form to a vote-tallying centre, where a computer system is used to tally all the votes at the end of the voting period. Here, we describe the electronic voting systems proposed by Song and Cui [19] and Zhou and Yan [26].

2.2.1. E-Voting Mechanism of Song and Cui

Song and Cui [19] proposed an e-voting mechanism that uses a blind signature scheme based on RSA and ElGamal encryption. The ElGamal encryption system was proposed by ElGamal [7], and the security of this scheme is based on the difficulty of the DLP. The e-voting scheme of Song and Cui consists of five stages: key generation, identity verification, ballot blinding, voting, and vote tallying. Key generation is performed using the RSA scheme, followed by ballot blinding using the ElGamal blind signature scheme. Finally, voting and vote tallying are performed via the transmission of XML files.

2.2.2. Blockchain-Based Anonymous E-Voting Protocol

Zhou and Yan [26] proposed an e-voting protocol based on blind signatures and timed-release encryption that operates using Ethereum smart contracts. In this system, blind signatures are used to authenticate voter identities, protect their privacy, and prevent losses of anonymity due to external attacks. The timed-release encryption algorithm is used for simultaneous vote tallying at the end of the election, which ensures uniqueness and fairness. Smart contracts are used to replace trusted third parties; in effect, this creates a trust-free system that guarantees the integrity and security of the voting process.

2.3. Electronic Signatures

Su *et al*. [20] proposed an ECC-based blind multi-document signcryption mechanism that is computationally efficient owing to the short length of the ECC keys. In the blinding phase, the data are divided into multiple blocks, and each document is divided in half before being hashed. The plaintext is then converted into points via plaintext-to-point mapping. Finally, a blinding factor is used to blind the messages, followed by the signing, unblinding, and verification stages. This approach significantly reduces the computational costs and outperforms encryption algorithms such as RSA and ElGamal with regard to the execution efficiency.

3. System Design

The proposed system is based on blind signcryption and blockchain technology. First, an Ethereum private chain is used to create a secure electronic voting mechanism. On this private chain, the smart contracts for the voting and vote-tallying processes are published to ensure transparency in the voting process. An Elliptic Curve Discrete Logarithm Problem (ECDLP)-based blind multi-document signcryption scheme is then used to enhance the security of the system. This also improves the efficiency by eliminating redundant signing processes. A self-certification mechanism with Girault's Level-3 security is used instead of a centralised trust server, so that the voters can participate in the computation of public and private keys. This mechanism prevents voter impersonation by untrustworthy certificate authorities, reduces the computational and storage burdens of the certificate server, and improves the execution efficiency. The architecture and processes of the proposed e-voting system are described below.

3.1. System Architecture

The architecture of the proposed system is shown in Figure 1. The voters and smart contracts are registered by the certificate authority through the blockchain network, and they participate in the computation of public/private keys and signatures. The voter and voting smart contract then mutually authenticate each other, while the ballots are encrypted by a one-time encryption scheme. When the ballot ciphertext is passed to the vote-tallying smart contract, a ciphertext digest is generated, which is blinded and then sent to the voting smart contract. Next, the voting smart contract passes the digest (signature) of the blinded ciphertext to the vote-tallying smart contract. Finally, the vote-tallying smart contract unblinds, verifies, and decrypts the votes.

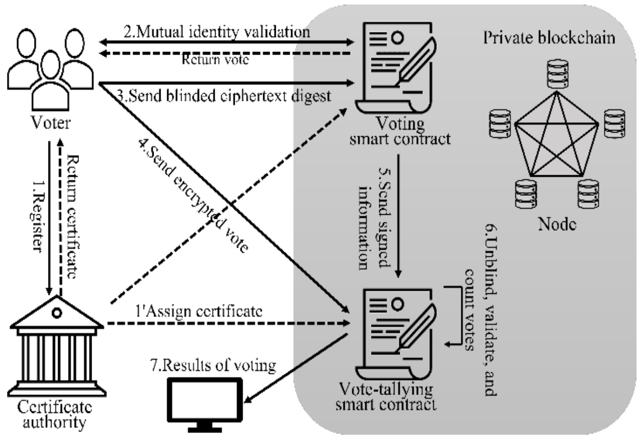

Figure 1. Operational architecture of the system.

3.2. Procedures and Algorithms of System

The procedures of the system can be divided into eight stages: initialisation, identity verification, encryption, blinding, signing, unblinding, signature verification, and decryption.

3.2.1. Initialisation

During initialisation, the election centre configures the parameters of the encryption system and the smart contracts on the private blockchain (Table 1).

Table 1. System parameters.

Item	Symbol	Description
1	A, B, C, D	Participants of the system, i.e., the voter, voting smart contract, vote-tallying smart contract, and certificate authority, respectively
2	id_z	ID information of z, with z being the identity of the participant
3	S_z, n_z	Public and private keys of z
4	r_z	Value randomly selected when calculating the correlation value of z
5	V_z	Signature for the registration application of z
6	PK_z	Verification key obtained by z after registration with the certificate authority
7	W_z	Signature calculated by the certificate authority and z
8	$h_1(), h_2()$	Hash function (value to value), hash function (point series to value)
9	$f_{m2p}()$	Function for converting the message into elliptic-curve points
10	$f_{p2m}()$	Function for converting elliptic-curve points into a message
11	w	0–1 knapsack value of the vote information
12	b	Blinding factor
13	m	Vote information
14	M	Hash of the vote information
15		Set the start- and end-time nodes of the voting and vote-tallying smart contracts

1. Key generation by certificate authority

The certificate selects a secure elliptic curve $E(F_q)$ in the finite domain F_q, where q represents a prime larger than 160 bits, and then selects a base point G of order d on $E(F_q)$ such that

$$d \cdot G = O \tag{1}$$

Where O represents the infinity point of the elliptic curve. Two collision-free one-way hash functions, i.e., $h_1()$ and $h_2()$, are then chosen to compute the key and message.

$$S_D = n_D \cdot G \tag{2}$$

E, G, q, S_D, $h_1()$, and $h_2()$ are then publicised.

2. Registration of voters and smart contracts (using voter A as exemplar)

The voter A selects a random encryption parameter $r_A \in [2, n-2]$ to generate the signature file V_A and then passes id_A (the ID of the voter) and V_A to the certificate authority D. V_A is calculated as Equation (3):

$$V_A = h_1(r_A \,||\, id_A)G \tag{3}$$

The certificate authority D then selects a secret parameter $r_D \in [2, n-2]$ to calculate the verification key and signature of voter A, i.e., PK_A and W_A, respectively, and passes them back to the voter A. PK_A and W_A are given as Equations (4) and (5):

$$PK_A = V_A + (r_D - h_1(id_A))G = (q_{Ax}, q_{Ay}) \tag{4}$$

$$W_A = r_D + n_D(q_{Ax} + h_1(id_A)) \tag{5}$$

The voter A uses the parameters returned by the certificate authority (verification key PK_A and signature W_A) to generate his/her own private key n_A and uses W_A to validate PK_A. n_A is given as Equation (6)

$$n_A = [\,W_A + h_1(r_A \,||\, id_A)\,] \tag{6}$$

The voter A then calculates his/her public key S_A as Equation (7):

$$S_A = n_A\, G \tag{7}$$

The same procedures are used to register the voting and vote-counting smart contracts with the certificate authority D. Once all the participants have registered with the certificate authority and thus obtained their verification keys and signatures (PK_z and W_z), they can independently calculate their public and private keys and verify the correctness. Additionally, they can authenticate their identities with parties that require identity validation using (id_z, PK_z, S_z) without relying on the certificate authority D.

3.2.2. Identity Verification

At this point, the voter A and voting smart contract B have obtained valid identity certificates from the certificate authority D. If the voting smart contract B receives (id_A, PK_A, S_A) from the voter A, it authenticates the voter A via the Equations (8 and 9):

$$S_A{}' = PK_A + h_1(id_A)\, G + [(q_{Ax} + h_1(id_A))] \cdot S_D \tag{8}$$

$$S_A{}' \overset{?}{=} S_A \tag{9}$$

Similarly, the voter *A* authenticates the voting smart contract *B* using Equation (10):

$$S_B' \overset{?}{=} S_B \tag{10}$$

Once the identities of both parties are successfully authenticated, the voting smart contract *B* sends *n* ballots to the voter *A*.

3.2.3. Encryption

After the voter *A* has received the ballots and submitted a vote, the cleartext contents of the votes are divided into “n” blocks, as Equation (11):

$$m_{ij} = m_{11}, m_{12}, \ldots, m_{n1}, m_{n2} \text{ ‚ } 1 \le i \le n \tag{11}$$

Each document is divided into two blocks. The m_{ij} cleartext is then hashed by converting the cleartext into points via cleartext-to-point mapping, as Equations (12-14):

$$\overline{m_{ij}} = \{m_{11}, m_{12}, \ldots, m_{n1}, m_{n2}\} \tag{12}$$

$$h_2(\overline{m_{ij}}) = m) \tag{13}$$

$$f_{m2p}(m) = P_1, P_2, \ldots, P_n \tag{14}$$

The following Equation (15) is assumed:

$$\bar{x} = \{x_1, x_2, \ldots, x_i\} \in (0,1) \tag{15}$$

If x_i is 1 while x_{i+1} is 0, the point “≫ 1” is right-shifted by one block. If x_i is 0 and x_{i+1} is 1, the point “≪ 1” is left-shifted by one block. If x_i is 1 and x_{i+1} is 1, the point “≫ 3” is right-shifted by three blocks. If x_i is 0 and x_{i+1} is 0, the point “≪ 3” is left-shifted by three blocks.

$$\text{if } x_i = 1 \text{ ; } x_{i+1} = 0 \gg 1 \tag{16}$$

$$x_i = 0 \text{ ; } x_{i+1} = 1 \ll 1 \tag{17}$$

$$\text{if } x_i = 1 \text{ ; } x_{i+1} = 1 \gg 3 \tag{18}$$

$$x_i = 0 \text{ ; } x_{i+1} = 0 \ll 3 \tag{19}$$

The binary number *w* is then calculated:

$$w = \left\{x_1 \cdot 2^{1-1}, x_2 \cdot 2^{2-2}, \ldots, x_n \cdot 2^{n-n}\right\} \tag{20}$$

A random value r_A, where $r_A \in Z_n^*$ and $r_A \in [2, n-2]$, is used to compute

$$R_A = r_A \cdot G \tag{21}$$

The public key of the vote-tallying smart contract *C* and r_A are then used to encrypt the ciphertext:

$$C_0 = \left[f_{m2p}(w, m) + r_A \cdot S_C\right] \tag{22}$$

$$C_1 = \left[P_1 + x_1 \cdot C_0 + r_A \cdot S_C\right] \tag{23}$$

$$C_2 = \left[P_2 + x_2 \cdot C_1 + r_A \cdot S_C\right] \tag{24}$$

$$C_n = \left[P_n + x_n \cdot C_{n-1} + r_A \cdot S_C\right] \tag{25}$$

$$\bar{C} = \{C_0, C_1, C_2, \ldots, C_n\} \tag{26}$$

$h_2()$ is used to process the ciphertext $\bar{C}$ to generate the ciphertext digest *M*:

$$h_2(\bar{C}) = M \tag{27}$$

3.2.4. Blinding

The public key of the vote-tallying smart contract *C*, i.e., S_C, and the random value r_A are used by the voter *A* to blind the ciphertext digest *M*, via the following Equations (28 and 29):

$$X = \left[f_{m2p}(r_A) + n_A \cdot S_C\right] \tag{28}$$

$$Y = r_A \cdot M \cdot S_C \tag{29}$$

X, the ciphertext $\bar{C}$, and the random values r_A and R_A are then passed to the vote-tallying smart contract *C*, while the blinded ciphertext digest *Y* is sent to the voting smart contract *B* for signing.

3.2.5. Signing

After the voting smart contract *B* receives *Y* from the voter *A*, it uses its private key n_B on *Y* to generate the signed document *MS*:

$$MS = n_B \cdot Y \tag{30}$$

The signed document *MS* is then passed to the vote-tallying smart contract *C*.

3.2.6. Unblinding

After the vote-tallying smart contract *C* receives the ciphertext $\bar{C}$, r_A, R_A, and *X* from the voter A and the signed document *MS* from the voting smart contract *B*, it applies $h_2()$ to the ciphertext $\bar{C}$ from the voter *A* to generate a second ciphertext digest *M*’. It then uses its private key n_C and the public key S_A of the voter *A* to unblind the signed document *MS*. This process is described by the following Equations (31 and 32):

$$h_2(\bar{C}) = M' \tag{31}$$

$$f_{m2p}(r_A) = X - n_C \cdot S_A \tag{32}$$

3.2.7. Signature Verification

The vote-tallying smart contract *C* then computes *MS*’ and thus verifies *M*’ using the public key of the voting smart contract *B*, i.e., S_B, as Equations (33 and 34):

$$r_A = f_{p2m}\left[f_{m2p}(r_A)\right] \tag{33}$$

$$MS' = r_A \cdot M' \cdot n_C \cdot S_B \tag{34}$$

The vote-tallying smart contract *C* then compares *MS*' with *MS*:

$$MS' \overset{?}{=} MS \tag{35}$$

If *MS*' and *MS* are equal, *MS*' is unaltered, and the signature of the vote-tallying smart contract *C* is valid.

3.2.8. Decryption

The vote-tallying smart contract *C* uses its private key n_C and R_A to decrypt the ciphertext $\bar{C}$:

$$f_{m2p}(w,m) = C_0 - n_C{\cdot}R_A \tag{36}$$

$$(w,m) = f_{p2m}\left[f_{m2p}(w,m)\right] \tag{37}$$

w is converted back into a series of numbers, $\bar{x}$. In this binary series, if x_i is 1 and x_{i+1} is 0, the point is left-shifted by one block. If x_i is 0 and x_{i+1} is 1, the point is right-shifted by one-block. If x_i is 1 and x_{i+1} is 1, the point is left-shifted by three blocks. If x_i is 0 and x_{i+1} is 0, the point is right-shifted by three blocks.

$$w = \left\{x_1{\cdot}2^{1-1}, x_2{\cdot}2^{2-2}, \dots, x_n{\cdot}2^{n-n}\right\} \tag{38}$$

$$\text{if} x_i = 1 \ ; \ x_{i+1} = 0 \ \ll 1 \tag{39}$$

$$x_i = 0 \ ; \ x_{i+1} = 1 \ \gg 1 \tag{40}$$

$$\text{if } \ x_i = 1 \ ; \ x_{i+1} = 1 \ll \ 3 \tag{41}$$

$$x_i = 0 \ ; \ x_{i+1} = 0 \ \gg 3 \tag{42}$$

$$\bar{x} = \{x_1, x_2, \dots, x_n\} \tag{43}$$

The ciphertext $\bar{C}$ is decrypted as Equation (44-48):

$$P_1' = \left[C_1 - x_1{\cdot}C_{1-1} - n_C{\cdot}R_A\right] \tag{44}$$

$$P_2' = \left[C_2 - x_2{\cdot}C_{2-1} - n_C{\cdot}R_A\right] \tag{45}$$

$$P_i' = \left[C_i - x_i{\cdot}C_{i-1} - n_C{\cdot}R_A\right] \tag{46}$$

$$\bar{P}' = \{P_1', P_2', P_3', \dots, P_n'\} \tag{47}$$

$$f_{p2m}(\bar{P}') = \overline{m_{\iota J}}' \tag{48}$$

$\overline{m_{\iota J}}'$ is a set consisting of multiple votes. Another one-time hash is performed on $\overline{m_{\iota J}}'$ using $h_2()$ to obtain *m'*:

$$h_2\left(\overline{m_{\iota J}}'\right) = m'$$

The contents of the vote are validated by comparing *m* with *m'*. If the vote is valid, it is included in the voting results.

4. Security and Benefits Analysis

We presented a blockchain-based e-voting system that uses smart contracts, ECC-based blind multi-document signcryption, and self-certified public-key cryptography. The system is capable of secure key distribution, ensuring voter privacy, and securely transferring the contents of each vote. In this section, a BAN logic analysis is performed to verify the security of the self-certified framework. Additionally, our system is analysed with regard to security metrics, for predicting the security level of the system and comparing its benefits with those of other e-voting systems.

4.1. BAN Logic

BAN-Logic is a logical concept used to analyse information exchange protocols. It can help each participant to trust the exchanged messages through necessary assumptions and it is a widely employed method for analysing authentication protocol [1].

Prior to the first transaction, the participants of our system mutually authenticate each other to ascertain whether they are authorised users. Therefore, BAN-logic analysis is performed to determine whether the voter *A* and smart contract *SC*—the two parties involved in the self-certification mechanism—can trust the public key *S* that they send to each other. If so, this proves the correctness and security of the proposed e-voting mechanism. The BAN-logic analysis is used to prove that the goals of the proposed mechanism can be achieved. The goals of the mechanism are formalised as follows:

Goal 1: $SC| \equiv S_A$
Goal 2: $\text{A}| \equiv S_{SC}$.

First, the protocol messages of the proposed mechanism must be expressed using BAN-logic syntax to ensure notational consistency in the subsequent derivation. The amended expressions of the messages are shown below:

Message 1: $A \rightarrow SC \ \colon (PK_A, S_A, ID_A)$

Message 2: $SC \rightarrow A \ \colon (PK_{SC}, S_{SC}, ID_{SC})$.

The following assumptions are proposed for our system, to facilitate further analysis.

Assumption 1: $A| \Rightarrow r_A$
Assumption 2: $SC| \equiv A| \sim (ID_A \, , d_A)$
Assumption 3: $SC| \Rightarrow r_{SC}$
Assumption 4: $SC| \equiv CA| \sim W_A$
Assumption 5: $A| \equiv CA| \sim W_{SC}$
Assumption 6: $A| \equiv SC| \sim (ID_{SC} \, , r_{SC})$
Assumption 7: $A| \equiv SC| \equiv (n_{SC}, CA| \sim PK_{SC})$
Assumption 8: $SC| \equiv A| \equiv (n_A, CA| \sim PK_A)$
Assumption 9: $A| \equiv ID_{SC}$
Assumption 10: $SC| \equiv ID_A$

According to these assumptions and the rules of BAN logic, we will prove that the voter A and smart contract SC can trust the authentication messages that they send to each other through the self-certification mechanism.

When *SC* receives Message 1, it can be proved that *SC* can see the message sent by *A*:

SC $\lhd$ $(PK_{\mathrm{A}}, S_{\mathrm{A}}, ID_{\mathrm{A}})$.
By jurisdiction,
SC $\lhd$ (S_{A}).

According to Equations (6) and (7) and Assumptions 1, 2, and 4, the following can be inferred:

$SC| \equiv \mathrm{A}| \Rightarrow S_A$ and $SC| \equiv \mathrm{A}| \equiv S_A$.

Therefore, according to the jurisdiction rules, it can be proved that

$\mathrm{SC}| \equiv S_A$(Goal 1).

Because the registration processes of n_{SC} and S_{SC} for *SC* are identical to those for the voter *A*, after *A* receives Message 2, the following can be proved using Assumptions 3, 5, and 6:

$\mathrm{A}| \equiv \mathrm{SC}| \Rightarrow S_{SC}$ and $\mathrm{A}| \equiv \mathrm{SC}| \equiv S_{SC}$.
It is then proved by jurisdiction that
$\mathrm{A}| \equiv S_{SC}$(Goal 2).

According to the aforementioned assumptions and goals, it can be concluded that the participants and smart contracts can both trust the *PK*, *S*, and *ID* that they send to each other. Therefore, a certificate authority is not necessary for identity verification, and the proposed system satisfies the requirements for self-certification. Furthermore, the participants have jurisdiction over their random values (*r*), which prevents impersonation by a third party. The security of the proposed self-certification mechanism is thus proven. Hence, the participants and smart contracts in the system can trust each other.

4.2. Security Analysis

In this section, we summarize the security items regulated by VVSG 2.0 [6] and analyse the security of the proposed ECDLP-based e-voting mechanism with regard to confidentiality, integrity, authenticity, anonymity, non-repudiation, and untraceability.

4.2.1. Confidentiality

Confidentiality is the characteristic that prevents data (i.e., documents and their contents) from being accessed by or revealed to unauthorised individuals, entities, or programs during their transmission. Therefore, if a system ensures confidentiality, no parties other than the sender and receiver may access the contents of the transmitted data. In our system, the voter *A* uses his/her private key n_A and the public key of the vote-tallying smart contract *C*, i.e., PK_C, to encrypt his/her vote message, in accordance with Equation (25). If an external party intercepts this encrypted message, it cannot decrypt the ciphertext without brute-forcing a solution for the ECDLP, because it does not possess the aforementioned public or private keys. Therefore, the proposed mechanism provides vote confidentiality.

4.2.2. Integrity

Integrity is the characteristic that prevents the alteration of data during their transmission and ensures the accuracy and completeness of all the data in the system. In the proposed system, the ciphertext digest signed by the voting smart contract *B* is hashed by the voter *A* using the one-way hash function $h_2()$, in accordance with Equation (13). If a third party intercepts the ciphertext sent by the voter *A* and sends a falsified or altered ciphertext to the vote-tallying smart contract *C*, the irreversibility of the one-way hash ensures that the ciphertext digest produced during signature verification will not match the correct signature. Hence, if a vote is verified by the vote-tallying smart contract *C*, the contents of the vote must be correct and complete, because the hash function produces the same ciphertext digest. It is thus proven that the proposed system can ensure vote integrity.

4.2.3. Authenticity

Authenticity pertains to the ability of the receiver to authenticate the message and ensure that the message came from the announced sender. In the proposed mechanism, the sender is the voter *A*, and the receiver is the voting smart contract *B*. During identity verification, the voting smart contract *B* may use Equations (18) and (19) to verify the identity of the voter *A*. If a third party wishes to impersonate the voter *A*, it must brute-force a solution for the ECDLP. Hence, the proposed mechanism ensures identity authenticity.

4.2.4. Anonymity

In the context of this study, anonymity is the characteristic that prevents the signatory from knowing the contents of the documents that they sign. Because the voting smart contract *B* only signs ciphertext digests of the votes, it cannot infer the contents of the votes. Furthermore, the ciphertext digest from the voter *A* has been blinded via the blind-signature technique, in accordance with Equations (28) and (29). The random number *k* is used to ensure that the ciphertext digest is always changing, which prevents the voting smart contract *B* from correlating ciphertext digests with vote data. This ensures the anonymity of the voter *A* in the signing process.

4.2.5. Non-Repudiation

Non-repudiation refers to the ability to prove that an action or event occurred in the past, so that the parties involved in the action or event cannot deny its occurrence. Because the certificates of the voter *A* are solely possessed by the voter *A*, once the voting smart contract *B* has verified the voter *A* by receiving his/her certificate and thus accepted his/her vote, the voter *A* cannot repudiate his/her vote to submit a second vote. Although the voter *A* can decide whether they wish to submit a vote after receiving their ballots, they are only allowed to vote once. Furthermore, because n_B belongs to the voting smart contract *B* alone, its signature (Equation (30)) can be verified by the vote-tallying smart contract *C* using Equation (34). This prevents the voting smart contract *B* from repudiating its signature.

4.2.6. Untraceability

In this context, untraceability refers to the impossibility of tracing the contents of a vote to a voter. Because voting smart contract *B* only signs the ciphertext digest *Y* (Equation (29)), it cannot infer the contents of the vote from the voter *A*. Furthermore, because the vote-tallying smart contract *C* only uses its own private key n_C to decrypt the votes (Equation (46)), it cannot identify the voter *A* from the contents of the vote. This ensures the untraceability of the voter *A*.

4.3. Comparison of Benefits

Table 2 compares the benefits of our e-voting mechanism with those of methods proposed by other researchers. The e-voting mechanism of Song and Cui does not provide anonymity, as it allows votes to be traced to their voters. In contrast to the e-voting mechanism of Zhou and Yan [26], our system uses a self-certification mechanism instead of a centralised certificate authority. This renders the participation of a trusted third party unnecessary, provides a highly decentralised structure, and reduces the key-management burden.

Table 2. Comparison of benefits between the proposed mechanism and other e-voting mechanisms.

Benefit	Song and Cui [19]	Zhou and Yan [26]	Proposed mechanism
Blockchain architecture	X	O	O
Minimal third-party participation	X	X	O
Decentralised	X	△	O
Consensus	X	O	O
Confidentiality	O	O	O
Integrity	O	O	O
Authenticity	X	O	O
Anonymity	O	O	O
Non-repudiation	O	O	O
Untraceability	△	O	O
Note: O, compliant; △, partially compliant; X, noncompliant			

5. Conclusions

This work presents an electronic voting mechanism that is sufficiently secure for practical use. We used ECC cryptography for our system because it has the same level of security as RSA and ElGamal encryption while having shorter key lengths, which allows the system to perform encryption and decryption operations more efficiently. We also used a blind multi-document signcryption mechanism that can be employed for simultaneous voting on multiple issues, which reduces the number of signing instances (particularly during multi-voting) and thus reduces the computational loads of the voting system. This private blockchain-based electronic voting mechanism is convenient and practical and fulfils all security requirements for E-voting, including confidentiality, integrity, authenticity, anonymity, non-repudiation, and untraceability. It will allow election organisers to quickly analyse voting results and obtain useful and objective data, facilitating the management of the multilevel elections. Furthermore, the self-certification mechanism helps to prevent identity forgery during certificate issuance and reduces the costs associated with the storage and management of public keys. In the future, we will investigate the possibility of including weights in the ballots and voting population, to give more weight to expert opinions and thus improve the diversity and reliability of the voting process with regard to decision making.

References

[1] Burrows M., Abadi M., and Needham R., “A Logic of Authentication,” *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18-36, 1990.

[2] Buterin V., “A Next-Generation Smart Contract and Decentralised Application Platform,” 2015, https://ethereum.org/zh/whitepaper/, Last Visited, 2021.

[3] Camenisch J., Piveteau J., and Stadler M., “Blind Signatures Based on the Discrete Logarithm Problem,” *in Proceedings of the Advances in Cryptology- EUROCRYPT '94*, Perugia, pp. 428-432, 1994.

[4] Chaum D., “Blind Signatures for Untraceable Payments,” *in Proceedings of the Advances in Cryptology*, Boston, pp. 199-203, 1983.

[5] Dutta P., Choi T., Somani S., and Butala R., “Blockchain Technology in Supply Chain Operations: Applications, Challenges and Research Opportunities,” *Transportation Research Part e: Logistics and Transportation Review*, vol. 142, 2020.

[6] Election Assistance Commission, “Voluntary Voting System Guidelines,” https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines, Last Visited, 2021.

[7] ElGamal T., “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.

[8] Fernández-Caramés T. and Fraga-Lamas P., “A Review on the Use of Blockchain for the Internet of Things,” *IEEE Access*, vol. 6, pp. 32979-33001, 2018.

[9] Girault M., “Self-Certified Public Keys,” *in Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*,

Brighton, pp. 490-497, 1991.

[10] Harn L., “Cryptanalysis of the Blind Signatures Based on the Discrete Logarithm Problem,” *IEEE Electronic Letters*, vol. 31, no. 14, pp. 1136-1137, 1995.

[11] Horster P., Michels M., and Petersen H., “Comment: Clyptanalysis of the Blind Signatures Based on the Discrete Logarithm Problem,” *IEEE Electronic Letters*, vol. 31, no. 21, pp. 1S27, 1995.

[12] Jeng F., Chen T., and Chen T., “All ECC-Based Blind Signature Scheme,” *Journal of Networks*, vol. 5, no. 8, pp. 921-928, 2010.

[13] Khan K., Arshad J., and Khan M., “Investigating Performance Constraints for Blockchain Based Secure E-voting System,” *Future Generation Computer Systems*, vol. 105, pp. 13-26, 2020.

[14] Lee C., Hwang M., and Yang W., “A New Blind Signature Based on the Discrete Logarithm Problem for Untraceability,” *Applied Mathematics and Computation*, vol. 164, no. 3, pp. 837-841, 2005.

[15] Lin S., Zhang L., Li J., Ji L., and Sun Y., “A Survey of Application Research Based on Blockchain Smart Contract,” *Wireless Networks*, vol. 28, pp. 635-690, 2022.

[16] Nakamoto S., “Bitcoin: A Peer-to-Peer Electronic Cash System,” pp. 1-9, 2008, http://bitcoin.org/bitcoin.pdf, Last Visited, 2021.

[17] National Conference of State Legislatures, “Online Voter Registration,” https://www.ncsl.org/research/elections-and-campaigns/electronic-or-online-voter-registration.aspx, Last Visited, 2021.

[18] Sabah N., Sagheer A., and Dawood O., “Blockchain-Based Solution for COVID-19 and Smart Contract Healthcare Certification,” *Iraqi Journal for Computer Science and Mathematics*, vol. 2, no. 1, pp. 1-8, 2021.

[19] Song F. and Cui Z., “Electronic Voting Scheme about ElGamal Blind-Signatures Based on XML,” *Procedia Engineering*, vol. 29, pp. 2721-2725, 2012.

[20] Su P., Yang L., and Wang P., “Multiple Blind Signcryption Scheme Based on ECC Technology -Design of the E-voting at one Time for Multiple Polls,” *Journal of Information Management*, vol. 14, pp. 73-94, 2014.

[21] Su P., Chang C., and Huang T., “Design and Implement of Self-Certified PKI Mechanism for E-commerce,” *Electronic Commerce Studies*, vol. 12, no. 1, pp. 73-92, 2014.

[22] Szabo N., “Formalizing and Securing Relationships on Public Networks,” vol. 2, no. 9, 1997. https://firstmonday.org/ojs/index.php/fm/article/download/548/469, Last Visited, 2021.

[23] Wood G., “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” 2014, https://gavwood.com/paper.pdf, Last Visited, 2021.

[24] Yu X. and He D., “A New Efficient Blind Signcryption,” *Wuhan University Journal of Natural Sciences*, vol. 13, no. 6, pp. 662-664, 2008.

[25] Zheng Y., “Digital Signclyption or How to Achieve Cost (Signature and Encryption) << Cost (Signature)+Cost (Encryption),” *in Proceedings of the Advances in Cryptology-Crypto '97*, California, pp. 165-179, 1997.

[26] Zhou Z. and Yan G., “Design of Block Chain-based Anonymous E-Voting Scheme,” *Software Guide*, vol. 19, no. 1, pp. 229-233, 2020.

Pin-Chang Su is presently working as a Professor in the Department of Information Management at National Defense University, Taiwan. He received his Ph.D. degree in Electrical Engineering from Chang Gung University, Taiwan in 2007. His research mainly focuses on Algorithms Design in Error-Control Coding, Information Security, Cryptographic Systems and E-Commerce Technologies. His published articles can be found in most academic journals like KSII Transactions on Internet and Information Systems, Computers and Electrical Engineering, Security and Communication Networks, Applied Mathematics and Computation, Journal of e-Business and so forth.

Tai-Chang Su is a young researcher, presently working as a MIS Manager in the National Defense University, Taiwan. He received the degree of Master of Management Science (2022) awarded by Department of Information Management, National Defense University, Taiwan. His research interests include Blockchain, Algorithms Design, and Cryptographic Systems.