JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 9 Number 2 | June 2024: 159-168
DOI: 10.15575/join.v9i1.1305

Strengthening the Authentication Mechanism of Blockchain-Based E-Voting System Using Post-Quantum Cryptography

Sonitema Laia[1], Ari Moesriami Barmawi[2]
[1]Graduate School of Informatics, School of Computing, Telkom University, Indonesia
[2] Graduate School of Forensic Science and Cyber Security, School of Computing, Telkom University, Indonesia

Article Info

Article history:

Received February 19, 2024
Revised April 17, 2024
Accepted April 22, 2024
Published August 24, 2024

Keywords:

Anonymous ID
Goldreich-Goldwasser-Halevi
Impersonation Attack
Man-in-the-middle Attack
Signature Scheme
Voter Anonymity

ABSTRACT

Election systems often face severe challenges regarding security and trust. Threats such as vote falsification and lack of transparency in vote counting have shaken the integrity of elections in various countries. The use of blockchain technology in e-voting has been proposed as an attractive solution to overcome this problem. Several studies use blockchain for the security of electronic voting systems. The existing methods are not resistant against impersonation attacks and man-in-the-middle attacks. This research proposes a new scheme to strengthen a blockchain-based e-voting system. The blockchain used in the proposed method is Ethereum. The proposed scheme uses the modified framework and The Goldreich-Goldwasser-Halevi (GGH) signature scheme. Digital signatures generated using Goldreich-Goldwasser-Halevi (GGH) can strengthen the identity of the message sender so that enemies cannot imitate someone. In this research, the Voter's public key and anonymous ID are used by the Voter to maintain the Voter's anonymity. Based on the experimental results, it can be concluded that the proposed scheme is stronger than the previous scheme because the probability of success in impersonating the sender with the proposed scheme using an impersonation attack and man-in-the-middle attack is small.

Corresponding Author:

Ari Moesriami Barmawi
Graduate School of Forensic Science and Cyber Security, School of Computing, Telkom University
Jalan Telekomunikasi No.1, Bandung 40257, Indonesia
Email: mbarmawi@melsa.net.id

1. INTRODUCTION

The development of blockchain technology has opened opportunities for strengthening the security and transparency of digital systems. One area with great potential for applying this technology is electronic voting systems (e-voting). E-voting has become more popular recently[1][2]. In addition to being essential for democratic nations, electoral integrity plays a significant role in boosting public trust and accountability [3][4][5]. E-voting system security concerns have been one of the subjects that has been thoroughly researched in the literature [6]. According to studies, electronic voting may generate security concerns [7][8]. Blockchain technology in e-voting has been proposed as an attractive solution to overcome this problem[9][10][11]. Blockchain is already well used in the electronic voting process

[12]. Several studies use blockchain increased [13][14] for the security of electronic voting systems, such as research by Wu & Yang [15] entitled "A blockchain-based network security mechanism for voting systems." In this research, there is a weakness in the sender authentication. This weakness makes possible impersonation attacks and man-in-the-middle attacks against the sender possible. In this attack, the attacker can impersonate a sender, threatening the integrity of the election.

In Wu & yang, there was a weakness, namely that in the Voter authentication session, the Certifying Authority sent a message to the Voter. In this session, the message sent by the Certifying Authority contains the message hash and the Voter's public key-based encrypted message. The Certifying Authority does not send its signature to the Voter, so the Voter cannot authenticate the validity of the Certifying Authority. As a result of this weakness, attackers can impersonate a Certifying Authority and send false data to Voters. The inability of Voters to verify the Certifying Authority can undermine public confidence in the integrity of electronic voting systems.

This research contributes to modifying the framework by adding a certified public key and implementing GGH cryptography and signature in all communication. It is proven that the proposed method is resistant to impersonation attacks and man-in-the-middle attacks. This condition is occurred because the probability of successful impersonation attacks and man-in-the-middle attacks for the proposed method scheme is $\frac{1}{d^{n\times n}}$. In this system also achieve the anonymity, transparency, and immutability of data with the blockchain.

2. METHOD

In this research, electronic voting begins when the Voter registers with the electronic voting system, and the Certifying Authority will register the Voter into the electronic voting system. Once registered, the Voter gives his vote to the Node, where his vote will be stored. The Government will then count all votes stored by the Node to obtain the results of this electronic voting, and the Government will provide or announce the results of the electronic vote count. The voting process is divided into three stages, namely, registration, voting, and vote reporting. Details of the electronic voting process are shown in Figure 1.

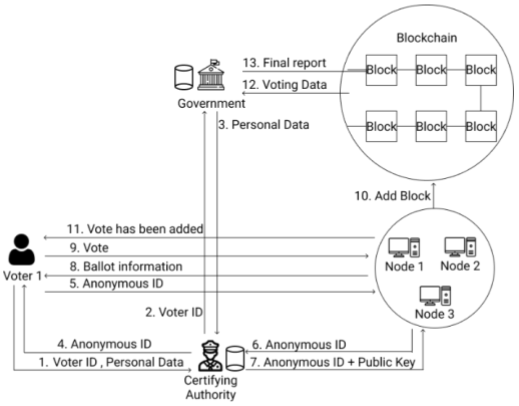


Figure 1. The E-Voting Scenario

In this research, a blockchain-based voting system is designed using a security scheme that uses GGH cryptography for message encryption and digital signature generation. Encryption protects the messages sent, while signatures verify the message's sender. GGH cryptography is used in security systems because it is a cryptographic method with a high level of security and is resistant to quantum computer attacks. The terms in Table 1 will be used in this research.

Table 1. Notation

Notation	Description	Notation	Description
V	Voter	SKCA	Private Key Certifying Authority
CA	Certifying Authority	PKCA	Public Key Certifying Authority
G	Government	SKG	Private Key Government
N	Node	PKG	Public Key Government
PK	Public Key	SKN	Private Key Node
SK	Private Key	PKN	Public Key Node
SKV	Private Key Voter	M	Message
PKV	Public Key Voter	$\{\{m\}_{SK}\}_{PK}$	Message m signed by the private key (SK) and encrypted by the public key (PK)

2.1 The GGH cryptosystem

In 1997, Goldreich, Goldwasser, and Halevi proposed an efficient way to build a cryptosystem that uses network theory (lattice theory) known as the GGH cryptosystem [16]. Lattice L is the infinite set of points in n-dimensional Euclidean space with a periodic structure [17]. In cryptography, lattices are used as a mathematical basis for developing complex security algorithms to solve certain mathematical problems. The lattice L generated by the vector $v_1, v_2, \dots, v_n$ is the set of all linear combinations of these vectors, where the coefficients are integers. In other words, this lattice is the set of all vectors that can be expressed as shown in Equation 1.

$$L = \{a_1 v_1 + a_2 v_2 + \cdots + a_n v_n;\ a_1, a_2, \dots, a_n \in \mathbb{Z} \tag{1}$$

A basis for L is any set of independent vectors that can generate L. two such basis sets have the same number of elements. The dimension of L is the number of vectors in one basis for L. Let $v_1, v_2, \dots, v_n$ be the basis for a lattice L, and $w_1, w_2, \dots, w_n \in L$ be a collection of other vectors that are also in L. As in vector spaces, each w_j is a linear combination of vectors the basis vector, as shown in Equation 2.

$$\begin{gathered} w_1 = a_{11}v_1 + a_{12}v_2 + \cdots + a_{1n}v_n, \\ w_2 = a_{21}v_1 + a_{22}v_2 + \cdots + a_{2n}v_n, \\ \vdots \\ w_n = a_{n1}v_1 + a_{n2}v_2 + \cdots + a_{nn}v_n \end{gathered} \tag{2}$$

When expressing v_i in terms of w_j, this involves matrix inversion as shown in Equation 3.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \tag{3}$$

As a one-way function for building a public key cipher that depends on lattice reduction difficulty, GGH provides a trapdoor [18]. The message must first be encoded as a lattice vector using the public basis to implement the encryption procedure. Next, a little error vector must be inserted. When implementing the decryption process, the closest lattice vector must be efficiently computed using the private basis [19]. GGH has several processes: first key creation, second encryption, and third decryption. Table 2 shows the GGH cryptosystem.

Table 2. The GGH Cryptosystem

Alice	Bob
Key Creation	
Choose a good basis $v_1, \dots, v_n$. Choose an integer matrix U, satisfying $\det(U) = \pm 1$ Compute a bad basis $w_1, \dots, w_n$ as the rows of $W = UV$ Publish the public key $w_1, \dots, w_n$	
Encryption	
	Choose small plaintext vector m Choose random small vector r Use Alice's public key to compute $e = x_1 v_1 + \cdots + x_n v_n + r$ Send the ciphertext e to Alice
Decryption	
Use babai's algorithm to compute the vector $v \in L$ closest e Compute vW^{-1} to recover m	

2.2 The GGH signature

GGH Signature is a digital signature algorithm based on the closest vector problem in a lattice. To approximate CVP, GGH applies the first Babai's approach [20]. In GGH Signature, the signing and verification process uses calculations in a lattice network. The security of the GGH Signature is based on the difficulty of solving the nearest vector problem in a lattice, which makes it resistant and difficult to attack [21]. In GGH Signature, users generate a digital signature using their private key, and the recipient of the signature can verify the signature using the corresponding public key. This algorithm has applications in various fields of digital security and cryptography. The digital signature scheme at GGH consists of 3 processes: first, the key creation process; second, the signature process; and third, the verification process. The GGH digital signature scheme is briefly explained in Table 3.

Table 3. The GGH Signature

Alice	Bob
Key Creation	
Choose a good basis $v_1, \dots, v_n$. Choose an integer matrix U, satisfying $\det(U) = \pm 1$ Compute a bad basis $w_1, \dots, w_n$ as the rows of $W = UV$ Publish the public key $w_1, \dots, w_n$	
Signing	
Choose document $d \in \mathbb{Z}^n$ to sign Use Babai's algorithm to compute a vector $s \in L$ that is close to d Write $s = a_1 w_1 + \cdots + a_n w_n$ Publish the signature $(a_1, \dots, a_n)$	
Verification	
	Compute $s = a_1 w_1 + \cdots + a_n w_n$ Verify that s is close to d

2.3 The Registration process

This session aims to register Voters who will take part in electronic voting. At this stage, Voter data will be checked for validity by validating the data sent by the Voter and authenticating the digital signature sent by the Voter. The Voter sends his encrypted signed message containing his ID and personal data to the Certifying Authority to register. After the Certifying Authority receives the message, The Certifying Authority validates the data from the Voter. The Certifying Authority sends his encrypted signed message containing the Voter ID to the Government to validate it. After getting the message from the Certifying Authority, the Government sends his encrypted signed message containing the requested data based on the ID. After receiving data from the Government, the Certifying Authority validates data from the Voter and the Government, if the data from the Voter and data from the Government are matched, the Certifying Authority creates anonymous ID for the Voter and sends his encrypted signed message containing the anonymous ID to the Voter. The registration process can be seen in Figure 2.

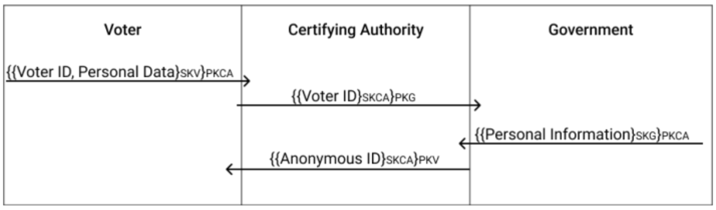

Voter
Certifying Authority
Government
$\{\{\text{Voter ID, Personal Data}\}_{SKV}\}_{PKCA}$
$\{\{\text{Voter ID}\}_{SKCA}\}_{PKG}$
$\{\{\text{Personal Information}\}_{SKG}\}_{PKCA}$
$\{\{\text{Anonymous ID}\}_{SKCA}\}_{PKV}$

Figure 2. The Registration Process

2.4 The Voting Process

This session aims to ensure that Voters who will carry out e-voting are valid, and their votes are sent to the blockchain. The Voter sends his encrypted signed message containing an anonymous ID to the Node. After getting the anonymous ID from the Voter, the Node authenticates the Voter. The Node sends his encrypted signed message containing the anonymous ID to the Certifying Authority. After getting the message from the Node, the Certifying Authority sends his encrypted signed message containing the public key based on the requested data from the anonymous ID. After getting the public key from the Certifying Authority, the Node then authenticates the signature from the Voter using the public key from the Certifying Authority. If the signature is valid, the Node sends his encrypted signed message containing ballot information to the Voter. After the Voter gets ballot information, the vote sends his encrypted signed message containing his vote to the Node. After getting the vote from the Voter, the Node adds the vote to the blockchain, and after the vote is added to the blockchain Node sends his encrypted signed message containing the notification message to the Voter. The voting process can be seen in Figure 3.

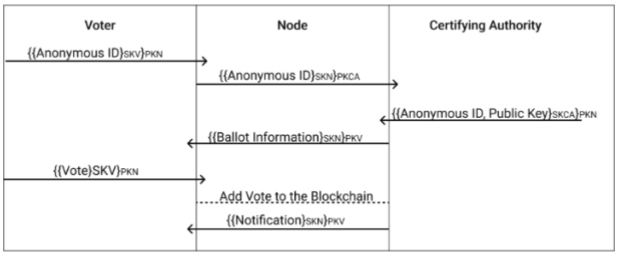

Voter
Node
Certifying Authority
{{Anonymous ID}$_{SKV}$}$_{PKN}$
{{Anonymous ID}$_{SKN}$}$_{PKCA}$
{{Anonymous ID, Public Key}$_{SKCA}$}$_{PKN}$
{{Ballot Information}$_{SKN}$}$_{PKV}$
{{Vote}SKV}$_{PKN}$
Add Vote to the Blockchain
{{Notification}$_{SKN}$}$_{PKV}$

Figure 3. The Voting Process

2.5 *The final report process*

This session aims to officially calculate and record the election results and announce the results. The Government retrieves the vote data that has been stored on the blockchain. Then, the Government calculates the results of e-voting. The Government sends the results of e-voting to the blockchain.

3. RESULT AND DISCUSSION

This research uses an experiment to test the security of communication used in the electronic voting system, especially against impersonation attacks and man-in-the-middle attacks.

3.1 *The implementation of GGH*

This section consists of implementing GGH key creation, encryption, decryption signing, and verification in the registration process and in the blockchain-based voting process.

3.1.1 The implementation of GGH key creation, encryption, decryption, signing, and verification in the registration process

The Voter chooses a nearly orthogonal basis V_v as Private Key, and W_v as public key.

$$V_v = \begin{pmatrix} 33 & -100 & 35 \\ -55 & -5 & -64 \\ -10 & -65 & -72 \end{pmatrix}, W_v = \begin{pmatrix} -102 & 595 & 76 \\ 23 & -165 & -37 \\ -83 & 995 & 286 \end{pmatrix}$$

Step 1. The Voter signs the message, encrypts it, and sends it to the Certifying Authority
- The Voter sends a message containing his ID and personal data to the recipient (which contains 12345,Soni). Then the sender creates a matrix m to represent the message
- The Voter computes the vector a close to m using his private key and Babai's algorithm
- The Voter computes his digital signature {Voter ID, Personal Data}$_{\text{SKS}}$, using Equation 4.

$$s = aW_v \tag{4}$$

$$m = \begin{pmatrix} 49 & 50 & 51 \\ 52 & 53 & 44 \\ 32 & 83 & 111 \\ 110 & 105 & 36 \end{pmatrix}, a = \begin{pmatrix} 22 & 105 & 29 \\ 22 & 105 & 29 \\ 10 & 65 & 72 \\ 79 & 85 & -22 \end{pmatrix}, s = \begin{pmatrix} 1 & 9 & 1 \\ 1 & 9 & 1 \\ -1 & -4 & 0 \\ 4 & 32 & 3 \end{pmatrix}$$

- Elements of s including symbols supporting s, are converted into ASCII code. The Voter concatenates m and s as M, chooses a random small vector r, and uses W_{ca} (as Certifying Authority's public key) to encrypts the message. {{Voter ID, Personal Data}$_{\text{SKS}}$}$_{\text{PKR}}$
- The Voter encrypts the message, {{Voter ID, Personal Data}$_{\text{SKS}}$}$_{\text{PKR}}$ using Equation 5.

$$e = mW_{ca} + r \tag{5}$$

$$W_{ca} = \begin{pmatrix} 227 & -72 & 255 \\ 503 & 178 & 145 \\ 138 & -113 & 261 \end{pmatrix}, M = \begin{pmatrix} 49 & 50 & 51 \\ 52 & 53 & 44 \\ 32 & 83 & 111 \\ 110 & 105 & 43 \\ 44 & 57 & 44 \\ 91 & 91 & 49 \\ 44 & 57 & 44 \\ 49 & 93 & 44 \\ 57 & 44 & 49 \\ 93 & 44 & 91 \\ 45 & 49 & 44 \\ 45 & 52 & 44 \\ 48 & 93 & 44 \\ 91 & 52 & 44 \\ 51 & 50 & 44 \\ 51 & 93 & 93 \end{pmatrix}, r = \begin{pmatrix} -1 & -2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & -3 & -1 \\ 2 & -3 & 0 \\ 2 & -3 & -1 \\ 0 & -3 & -3 \\ 2 & -1 & 2 \\ 0 & -2 & -3 \\ -1 & 0 & -1 \\ 1 & 1 & 0 \\ 3 & -3 & 2 \\ -3 & 0 & 1 \\ 3 & 2 & 3 \\ -1 & 0 & 3 \\ 2 & -1 & -3 \end{pmatrix}, e = \begin{pmatrix} 43310 & -393 & 33056 \\ 44537 & 720 & 32429 \\ 64331 & -73 & 49167 \\ 83719 & 5908 & 54499 \\ 73190 & 4106 & 49189 \\ 44733 & 2003 & 30968 \\ 63974 & 8051 & 37461 \\ 51378 & -2803 & 41796 \\ 41833 & -1811 & 33701 \\ 55800 & -9147 & 53845 \\ 40935 & 511 & 30064 \\ 42446 & 1041 & 30501 \\ 63744 & 8126 & 37210 \\ 52888 & -2266 & 42232 \\ 42798 & 256 & 31742 \\ 71192 & 2372 & 50760 \end{pmatrix}$$

Step 2. The Certifying Authority decrypts the messages the Voter sends using his public key and verifies the signature using the Voter's public key.
- After getting the encrypted message, the Certifying Authority decrypts the encrypted message.
- The Certifying Authority uses babai's method to compute vector v that is close to e
- The Certifying Authority then decrypts the encrypted message using Equation 6.

$$M = vW_{ca}{}^{-1} \tag{6}$$

$$v = \begin{pmatrix} 43311 & -391 & 33056 \\ 44535 & 718 & 32429 \\ 64331 & -73 & 49166 \\ 83719 & 5911 & 54498 \\ 73192 & 4109 & 49189 \\ 44731 & 2006 & 30969 \\ 63974 & 8054 & 37464 \\ 51376 & -2802 & 41794 \\ 41833 & -1809 & 33704 \\ 55801 & -9147 & 53846 \\ 40934 & 510 & 30064 \\ 42443 & 1044 & 30499 \\ 63747 & 8126 & 37209 \\ 52885 & -2268 & 42229 \\ 42799 & 256 & 31739 \\ 71190 & 2373 & 50763 \end{pmatrix}, \quad M = \begin{pmatrix} 49 & 50 & 51 \\ 52 & 53 & 44 \\ 32 & 83 & 111 \\ 110 & 105 & 43 \\ 91 & 91 & 49 \\ 44 & 57 & 44 \\ 49 & 93 & 44 \\ 91 & 49 & 44 \\ 57 & 44 & 49 \\ 93 & 44 & 91 \\ 45 & 49 & 44 \\ 45 & 52 & 44 \\ 48 & 93 & 44 \\ 91 & 52 & 44 \\ 51 & 50 & 44 \\ 51 & 93 & 93 \end{pmatrix}$$

- The Certifying Authority then verifies the digital signature {Voter ID, Personal Data}$_{\mathrm{SKV.}}$ It is represented as s , W_v as public key Voter. Finally, the Certifying Authority Computes a using Equation 7.

$$a = sW_v \tag{7}$$

$$s = \begin{pmatrix} 1 & 9 & 1 \\ 1 & 9 & 1 \\ -1 & -4 & 0 \\ 4 & 32 & 3 \end{pmatrix}, W = \begin{pmatrix} -102 & 595 & 76 \\ 23 & -165 & -37 \\ -83 & 995 & 286 \end{pmatrix}, m = \begin{pmatrix} 49 & 50 & 51 \\ 52 & 53 & 44 \\ 32 & 83 & 111 \\ 110 & 105 & 36 \end{pmatrix}, a = \begin{pmatrix} 22 & 105 & 29 \\ 22 & 105 & 29 \\ 10 & 65 & 72 \\ 79 & 85 & -22 \end{pmatrix}$$

- Verify that a is close to m by calculating v using Equation 8.

$$v = \|a - m\| \tag{8}$$

$$v = \begin{pmatrix} 65 & 61 & 48 & 69 \end{pmatrix}$$

- Since the value of v is less than 100, so the signature is valid.

Step 3. The Certifying Authority signs the Voter's identity number (the message is 12345) using his private key and encrypts it using the Government's public key. The method of the Certifying Authority signing and encrypting the message is the same as in Step 1. Furthermore, the Certifying Authority sends the encrypted and signed message to the Government.

Step 4. The Government decrypts the messages sent by the Certifying Authority using his public key and verifies the signature using the Certifying Authority's public key. The method by which the Government decrypts the message and verifies the signature is the same as in Step 2.

Step 5. The Government signs the messages (which is 12345, Soni) using his private key and encrypts them using the Certifying Authority's public key. The method by which the Government signs and encrypts the message is the same as in Step 1. Furthermore, the message is sent to the Certifying Authority.

Step 6. The Certifying Authority decrypts the messages sent by the Government using its public key and verifies the signature using the Government's public key. The method for decrypting the message and verifying the signature by the Certifying Authority is the same as Step 2.

Step 7. The Certifying Authority signs the anonymous ID of the Voter (the message is 39082) using his private key and encrypts it using the Voter's public key. The method for signing and encrypting the message by Certifying Authority is the same as Step 1. Furthermore, the Certifying Authority sends the message to the Voter.

Step 8. The Voter decrypts the messages sent by the Certifying Authority using his public key and verifies the signature using the Certifying Authority's public key. The method for decrypting the message and verifying the signature is the same as Step 2.

3.1.2 The Implementation of GGH key creation, encryption, decryption, signing, and verification in the blockchain-based voting process

Step 1. The Voter signs his anonymous ID (the message is 39082) using his private key and encrypts it using the Node’s public key. The method by which the Voter signs and encrypts the message is the same as Step 1 in the registration process. Furthermore, the Voter sends the anonymous ID to the Node.

Step 2. The Node decrypts the messages sent by the Voter using his public key and verifies the signature using the Voter’s public key. The method for decrypting the message and verifying the signature by the Node is the same as Step 2 in the registration process.

Step 3. The Node signs the anonymous ID (The message is 39082) using his private key and encrypts the message using the Certifying Authority’s public key. The method for signing and encrypting the message by the Node is the same as Step 1 in the registration process. The Node sends the encrypted message to the Certifying Authority.

Step 4. The Certifying Authority decrypts the messages sent by the Node using its public key and verifies the signature using the Node’s public key. The method for decrypting the message and verifying the signature by the Certifying Authority is the same as Step 2 in the registration process.

Step 5. The Certifying Authority signs the messages (which is 39082, the Voter’s Public Key) using his private key and encrypts the message using the Node’s public key. The method for signing and encrypting the message by Certifying Authority is the same as Step 1 in the registration process.

Step 6. The Node decrypts the messages sent by the Certifying Authority using its public key and verifies the signature using the Certifying Authority’s public key. The method for decrypting the message and verifying the signature by the Node is the same as Step 2 in the registration process.

Step 7. The Node signs the Ballot (the message is Ballot) using his private key and encrypts the message using the Voter’s public key. The method for signing and encrypting the message by the Node is the same as Step 1 in the registration process. The Node sends the Ballot to the Voter.

Step 8. The Voter decrypts the messages sent by the Node using his public key and verifies the signature using the Node’s public key. The method for decrypting the message and verifying the signature by the Voter is the same as Step 2 in the registration process.

Step 9. The Voter signs his vote (The message is a vote) using his private key and encrypts the message using the Node’s public key. The method for signing and encrypting the message by the Voter is the same as Step 1 in the registration process. Furthermore, the message is sent to the Node.

Step 10. The Node decrypts the messages sent by the Voter using his public key and verifies the signature using the Voter's public key. The method for decrypting the message and verifying the signature by the Node is the same as Step 2 in the registration process. The vote, the anonymous ID, and the Signature of the Vote are stored in the blockchain.

Step 11. The Node signs the notification message (The message is a notification) using his private key and encrypts the message using the Voter's public key. The method for signing and encrypting the message by the Node is the same as Step 1 in the registration process. The notification message is sent to the Voter.

Step 12. The Voter decrypts the messages sent by the Node using his public key and verifies the signature using the Node's public key. The method for decrypting the message and verifying the signature by the Voter is the same as Step 2 in the registration process.

3.2. *Impersonation of the Certifying Authority in the registration process*

This attack aims to impersonate a Certifying Authority and give false data to Voters, such that the Voter cannot vote because he gets the wrong data. In this case, the attacker impersonates a Certifying Authority. If the attacker wants to impersonate the Certifying Authority, he needs to generate the correct Certifying Authority's private key to create the Certifying Authority's signature. So, if the recipient wants to verify the signature, he will get a valid signature. To generate the Certifying Authority's private key, an attacker needs to know the dimension of the matrix and length d, where d is the range of the private key value. So, the probability that the attacker can generate the correct Certifying Authority's private key is $\frac{1}{d^{n\times n}}$. In this case, n is the number of dimensions. Assuming the attacker knows the private key length is three dimensions, the range of d or private key values is from -100 to 100. The probability of success guessing the private key by the attacker is $\frac{1}{201^9}$. This attack can be seen in Figure 4.

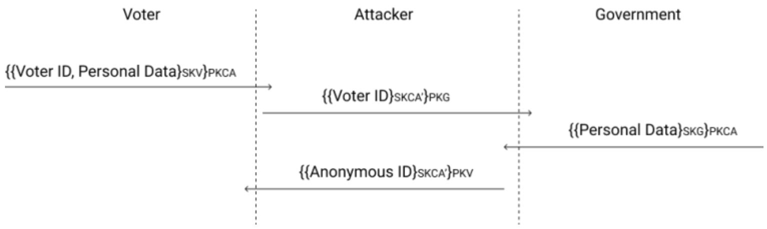

Voter
Attacker
Government
$\{\{\text{Voter ID, Personal Data}\}_{SKV}\}_{PKCA}$
$\{\{\text{Voter ID}\}_{SKCA'}\}_{PKG}$
$\{\{\text{Personal Data}\}_{SKG}\}_{PKCA}$
$\{\{\text{Anonymous ID}\}_{SKCA'}\}_{PKV}$

Figure 4. Impersonation of the Certifying Authority

In Figure 5, the attacker is trying to impersonate the Certifying Authority. The attacker sends a message to the Voter, but after getting the message, the Voter verifies the signature.

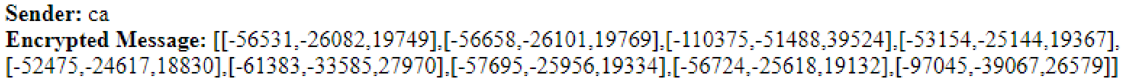
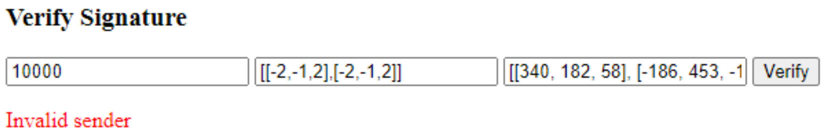

Sender: ca
Encrypted Message: [[-56531,-26082,19749],[-56658,-26101,19769],[-110375,-51488,39524],[-53154,-25144,19367],[-52475,-24617,18830],[-61383,-33585,27970],[-57695,-25956,19334],[-56724,-25618,19132],[-97045,-39067,26579]]

Figure 5. The attacker sent messages to the Certifying Authority

In Figure 6, the result is invalid because the attacker generates the wrong signature of the Certifying Authority.


Verify Signature

10000 | [[-2,-1,2],[-2,-1,2]] | [[340, 182, 58], [-186, 453, -1

Invalid sender

Figure 6. The Voter Verifies the signature.

Based on discussion and evaluation, the proposed scheme is stronger than Wu and Yang scheme [15] against impersonation attacks because the probability of successful impersonation attacks for the proposed scheme is $\frac{1}{d^{n\times n}}$, and for Wu & Yang is 1.

3.3. *MITM between the Voter and the Certifying Authority in the registration process*

This attack aims to change the data sent by the Certifying Authority using incorrect data. The attacker then intercepts communications between the Voter and the Certifying Authority. Suppose the attacker wants to impersonate the Certifying Authority and send a new message to the Voter. In that case, the attacker needs to create a new signature using the Certifying Authority's private key so that the Voter still recognizes that the message's sender is the Certifying Authority. To generate the Certifying Authority's private key, an attacker needs to know the dimension of the matrix and length d. So, the probability that the attacker can generate the correct Certifying Authority's private key is $\frac{1}{d^{n\times n}}$. Assuming the attacker knows the private key length is three dimensions, the range of d or private key values is from -100 to 100. The probability of the attacker generating the correct private key is $\frac{1}{201^9}$. This attack can be seen in Figure 7.

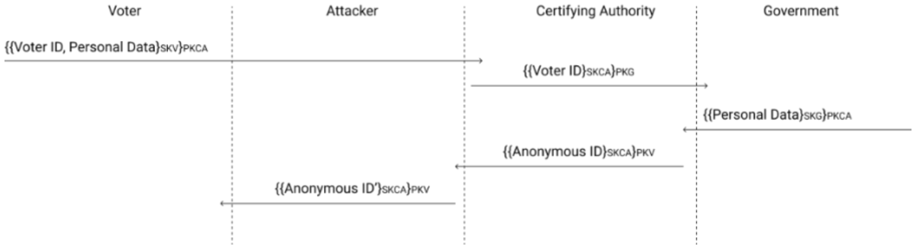
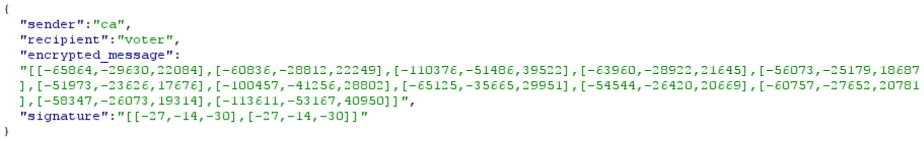


Figure 7. MITM attack between the Voter and the Certifying Authority

The result of this attack can be seen in Figure 10. Figure 8 shows how the attacker intercepts a message from the Certifying Authority.



```
{
  "sender":"ca",
  "recipient":"voter",
  "encrypted_message":
  "[[-65864,-29630,22084],[-60836,-28812,22249],[-110376,-51486,39522],[-63960,-28922,21645],[-56073,-25179,18687
  ],[-51973,-23626,17676],[-100457,-41256,28802],[-65125,-35665,29951],[-54544,-26420,20669],[-60757,-27652,20781
  ],[-58347,-26073,19314],[-113611,-53167,40950]]",
  "signature":"[[-27,-14,-30],[-27,-14,-30]]"
}
```

Figure 8. Attacker Intercepts the communication

In this simulation, the attacker creates a new message and sends it to the Voter. The Voter receives an encrypted message in the second row in Figure 9.

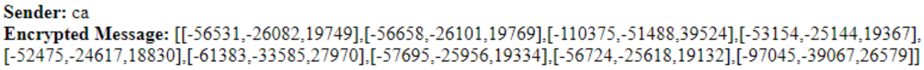
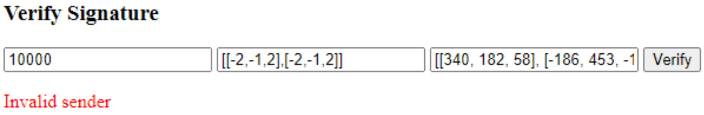

Sender: ca
Encrypted Message: [[-56531,-26082,19749],[-56658,-26101,19769],[-110375,-51488,39524],[-53154,-25144,19367],
[-52475,-24617,18830],[-61383,-33585,27970],[-57695,-25956,19334],[-56724,-25618,19132],[-97045,-39067,26579]]

Figure 9. The atacker sends new message to the Voter

After getting the message, the Voter decrypts the message using his private key. After decrypting the message, the Voter verifies the signature of the sender. The result is invalid because the attacker generates the wrong signature of the Certifying Authority.


Verify Signature

10000 | [[-2,-1,2],[-2,-1,2]] | [[340, 182, 58], [-186, 453, -1

Invalid sender

Figure 10. The voter verifies the signature

Based on discussion and evaluation, the proposed scheme is stronger than Wu and Yang scheme [15] against man-in-the-middle attacks because the probability of successful man-in-the-middle attacks for the proposed scheme is $\frac{1}{d^{n\times n}}$, and for Wu & Yang is 1.

4. CONCLUSION

The proposed scheme using GGH digital signatures can overcome the problem of existing method, which are impersonation attacks and man-in-the-middle attacks. Based on the results of experiments and analysis, it has been proven that the proposed method is secure against man-in-the-middle attacks and impersonation attacks because the probability of succeeding man-in-the-middle attack and impersonation attack is $\frac{1}{d^{n\times n}}$. This research also evaluates the time complexity of the proposed scheme. The encryption time complexity used in the proposed scheme is $O(mnp)$. In this system also achieve the anonymity, transparency, and immutability of data with the blockchain but, in this proposed scheme the Voter cannot directly input the vote to the blockchain. To input the vote to the Blockchain the Voter needs the Node. For future work, it is necessary to evaluate the security of the blockchain system.

REFERENCES

[1] A. Benabdallah, A. Audras, L. Coudert, N. El Madhoun, and M. Badra, "Analysis of Blockchain Solutions for E-Voting: A Systematic Literature Review," *IEEE Access*, vol. 10, pp. 70746–70759, 2022, doi: 10.1109/ACCESS.2022.3187688.

[2] M. Sallal, R. de Fréin, and A. Malik, "PVPBC: Privacy and Verifiability Preserving E-Voting Based on Permissioned Blockchain," Future Internet, vol. 15, no. 4, p. 121, Mar. 2023, doi: 10.3390/fi15040121.

[3] R. Taş and Ö. Ö. Tanrıöver, "A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting," Symmetry (Basel), vol. 12, no. 8, p. 1328, Aug. 2020, doi: 10.3390/sym12081328.

[4] R. AlAbri, A. K. Shaikh, S. Ali, and A. H. Al-Badi, "Designing an E-Voting Framework Using Blockchain Technology," International Journal of Electronic Government Research, vol. 18, no. 2, pp. 1–29, Mar. 2022, doi: 10.4018/IJEGR.298203.

[5] M. J. Beck and D. A. Hensher, "Insights into the impact of COVID-19 on household travel and activities in Australia – The early days under restrictions," Transp Policy (Oxf), vol. 96, pp. 76–93, Sep. 2020, doi: 10.1016/j.tranpol.2020.07.001.

[6] P. Y. A. Ryan, S. Schneider, and V. Teague, "End-to-End Verifiability in Voting Systems, from Theory to Practice," IEEE Secur Priv, vol. 13, no. 3, pp. 59–62, May 2015, doi: 10.1109/MSP.2015.54.

[7] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," Telematics and Informatics, vol. 36, pp. 55–81, Mar. 2019, doi: 10.1016/j.tele.2018.11.006.

[8] Prof. Amree Khan, Jayesh Bhaisare, Kajal Chandekar, and Aditi Lichade, "Online Voting and Information Management," International Journal of Advanced Research in Science, Communication and Technology, pp. 514–517, Dec. 2022, doi: 10.48175/IJARSCT-7717.

[9] S. Park, M. Specter, N. Narula, and R. L. Rivest, "Going from bad to worse: from Internet voting to blockchain voting," J Cybersecur, vol. 7, no. 1, Feb. 2021, doi: 10.1093/cybsec/tyaa025.

[10] R. Krimmer, D. Duenas-Cid, and I. Krivonosova, "New methodology for calculating cost-efficiency of different ways of voting: is internet voting cheaper?," Public Money & Management, vol. 41, no. 1, pp. 17–26, Jan. 2021, doi: 10.1080/09540962.2020.1732027.

[11] D. Duenas-Cid, I. Krivonosova, R. Serrano, M. Freire, and R. Krimmer, "Tripped at the Finishing Line: The Åland Islands Internet Voting Project," 2020, pp. 36–49. doi: 10.1007/978-3-030-60347-2_3.

[12] C. H. Roh and I. Y. Lee, "A study on electronic voting system using private blockchain," Journal of Information Processing Systems, vol. 16, no. 2, pp. 421–434, Apr. 2020, doi: 10.3745/JIPS.03.0135.

[13] P. Baudier, G. Kondrateva, C. Ammi, and E. Seulliet, "Peace engineering: The contribution of blockchain systems to the e-voting process," Technol Forecast Soc Change, vol. 162, p. 120397, Jan. 2021, doi: 10.1016/j.techfore.2020.120397.

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

[15] H.-T. Wu and C.-Y. Yang, "A Blockchain-Based Network Security Mechanism for Voting Systems," in 2018 1st International Cognitive Cities Conference (IC3), IEEE, Aug. 2018, pp. 227–230. doi: 10.1109/IC3.2018.00-15.

[16] Q. K. Kadhim, B. M. Al-Nedawe, and E. M. Hameed, "Encryption and Decryption of Images using GGH Algorithm: Proposed," IOP Conf Ser Mater Sci Eng, vol. 1090, no. 1, p. 012063, Mar. 2021, doi: 10.1088/1757-899X/1090/1/012063.

[17] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, New York, NY, USA: ACM, May 2005, pp. 84–93. doi: 10.1145/1060590.1060603.

[18] A. Sipasseuth, T. Plantard, and W. Susilo, "Enhancing Goldreich, Goldwasser and Halevi's scheme with intersecting lattices," Journal of Mathematical Cryptology, vol. 13, no. 3–4, pp. 169–196, Sep. 2019, doi: 10.1515/jmc-2016-0066.

[19] M. Belkasmi, F. El Bouanani, Institute of Electrical and Electronics Engineers. Morocco Section., and Institute of Electrical and Electronics Engineers, 2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS'16) : proceedings : October 17-19, 2016, Marrakesh, Morocco.

[20] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," Combinatorica, vol. 6, no. 1, pp. 1–13, Mar. 1986, doi: 10.1007/BF02579403.

[21] T. Plantard, W. Susilo, and K. T. Win, "A Digital Signature Scheme Based on CVP ∞," in Public Key Cryptography – PKC 2008, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 288–307. doi: 10.1007/978-3-540-78440-1_17.