PSWS: A Private Support-Weighted Sum Protocol for Blockchain-Based E-Voting Systems

Chenyu Deng(✉)

School of Computer Science, Hubei University of Technology, Wuhan, China
csmwzhang@gmail.com

Abstract. Nowadays, e-voting systems are receiving a lot of attention. Voters can cast their votes for the candidates they support on the e-voting system. However, how to reflect the voters' support level for the candidates in the e-voting system in a fair and privacy-preserving way is still a problem that needs to be solved. This article proposes a private support-weighted sum (PSWS) protocol, which is a fair, and privacy-preserving weighted sum protocol. The PSWS protocol privately calculates the weighted sum of each voter's support degree for a candidate. After the protocol's execution, others on the system can only access the weighted sum, without any additional information. The PSWS protocol has two novel characteristics. Firstly, the voting terminal or polling station provides encryption services for ballots immediately after each ballot is cast. All voting information is expressed in ciphertext throughout the weighting and counting processes, until the final result of the weighted vote is passed to the decryption server in ciphertext. This design avoids the disclosure of voter privacy and ballot information, and the ciphertext format also prevents malicious users from cheating or tampering with voter or ballot information during the counting process. Security analysis is conducted to validate security properties. Secondly, our protocol not only achieves the function of privacy-preserving support-weighted voting but also is relatively lightweight and efficient. Finally, the efficiency analysis results of the experiments in terms of calculation show that the protocol meets the requirements applicable to real-world applications. In summary, PSWS can be harmoniously applied to candidate election in blockchain systems.

Keywords: Privacy protection · Homomorphic encryption · Weighted sum protocol · Blockchain

This work is supported in part by the Major Research Plan of Hubei Provience under Gran No. 2023BAA027, and the Key Research and Development Program of Hubei Province under Grant 2021BEA163.

J. Chen and Z. Xia (Eds.), BlockTEA 2023, LNICST 577, pp. 79–93, 2024.
https://doi.org/10.1007/978-3-031-60037-1_5

1 Introduction

In today's increasingly digital world, the demand for secure and transparent e-voting systems increasingly critical [1]. Electronic voting machine hacking, vote rigging, electoral fraud, and capturing the polling booth are some of the major issues in current election system [15]. Ensuring the integrity of elections is fundamental to upholding the principles of democracy and maintaining public trust in the electoral process [16]. Traditional voting methods have faced numerous challenges, including concerns about voter fraud, ballot tampering, and the logistical complexities of conducting elections. To address these challenges, emerging technologies such as blockchain offer a promising avenue for revolutionizing the way we conduct and secure elections. Blockchain is totally transparent, secure and immutable technique because it uses concepts like encryption, decryption, hash function, consensus and Merkle tree [9].

Let us consider the following application scenario within electronic voting systems. We have a voter who wishes to cast votes for multiple candidates in an election. The voter supports several candidates to varying degrees but is unwilling to disclose his private information. In such a scenario, the problem to address is how to enable voting with varying degrees of support for multiple candidates without revealing individual voters' preferences or compromising privacy. Therefore, this paper proposes a privacy-preserving weighted sum protocol, private support-weighted sum(PSWS). If the critical problem underlying an application scenario can be translated to the privacy-preserving aggregation problem of a series of integer arrays (each array can be viewed as the support degree received by each candidate), then the PSWS protocol applies to this scenario.

1.1 Contributions

The main contributions of our work can be summarized as follows:

- To address the challenge of facilitating a voter's ability to cast votes for multiple candidates within a blockchain-based election voting system, while accurately reflecting the level of support for each candidate, we propose an innovative solution: the PSWS protocol.
- Expanding upon our proposed solution, we introduce a privacy-preserving support-weighted voting algorithm that prioritizes privacy protection while accurately computing the overall support degree for each candidate. This algorithm incorporates advanced techniques, such as homomorphic encryption, to enable the computation of candidate support degree without compromising individual voters' privacy. This approach guarantees the accurate determination of total support degree for each candidate while respecting the confidentiality of voters' choices.
- We undertake an extensive security analysis to validate the effectiveness of the PSWS protocol in preserving both the privacy of user data and computation results. This rigorous analysis serves to establish the algorithm's resilience against potential attacks and its capacity to offer a substantial level of security in safeguarding user data privacy while upholding the confidentiality of computed results.

1.2 Plan of This Paper

The remaining structure of this paper is organized as follows: In Sect. 2, we provide some preliminary insights. In Sect. 3, we present the system model, threat model, security requirements, and design goals. Section 4 showcases the specific details of the system design, followed by security proofs in Sect. 5. Section 6 covers the experimental part, including performance evaluation. In Sect. 7, we delve into related work, and in Sect. 8, we conclude the entire paper while providing prospects for future work.

2 Preliminaries

In this section, we reviewed the relevant knowledge of BGN [4] homomorphic encryption and blockchain that were utilized throughout the entire work. As shown in Table 1, we outline the notations used in this paper.

Table 1. Table captions should be placed above the tables.

Notation	Remarks
τ	Security parameter
e	Bilinear map
Can	Set of candidates
n	Number of voters
u	Number of candidates
$a_{i,j}$	The jth vote for the ith candidate
t_i	The ith candidate supported by the voters
$m_{i,j}$	The support degree paid by the jth voter of the ith candidate
$[\![\frac{1}{n}]\!]$	Ciphertext of $\frac{1}{n}$
C_{t_i}	Ciphertext of t_i
$[\![\frac{1}{n^2}]\!]$	Ciphertext of $\frac{1}{n^2}$
$C_{m_{i,j}}$	Ciphertext of $m_{i,j}$
V_{m_i}	The set of ciphertexts of the support degrees for candidate t_i
T_i	Sum of support degrees for the ith candidate
C_{T_i}	Ciphertext of T_i
Var_i	Variance of support degrees for the ith candidate
C_{Var_i}	Ciphertext of Var_i
$score(t_i)$	The sum of the weighted support degree of the candidate t_i

2.1 BGN Homomorphic Encryption Scheme

The BGN homomorphic cryptosystem is composed of the following:

- **Gen**(τ): Input the security parameter τ and running **Gen**(τ) yields the tuple (q_1, q_2, G, G_1, e), where G and G_1 are groups of order $N = q_1 * q_2$, $e : G * G \rightarrow G_1$is a bilinear map, and two random generators $k, U \leftarrow G$. Setting $h = U^{q_2}$ results in h being a random generator of the q_1-order subgroup of group G.Set the public key $PK = (N, G, G_1, e, h, k)$ and the private key $SK = q_1$.
- **Encryption** (PK, m): Give a plaintext space $\{0, 1, 2, ..., T\}(T < q_2)$, and given a random selection of $r_N \xleftarrow{R} \{0, 1, ..., N-1\}$, along with the input plaintext message m and the public key PK, the output ciphertext $C = k^m h^r \in G$ is generated.
- **Decryption** (SK, C): Input the ciphertext C and the private key SK, you can calculate $C^{q_1} = (k^m h^r)^{q_1} = (k^{q_1})^m$. By utilizing Pollard's lambda algorithm [6] to solve the discrete logarithm modulo k^{q_1}, you can recover the plaintext message m.
  BGN encryption scheme has the following homomorphic property:
  Homomorphic Addition: $C = c_1 c_2 h^r = k^{m_1+m_2} h^{r_1+r_2+r}$.
  Homomorphic Multiplication: $C = e(C_1, C_2) h_1^r = k_1^{m_1 m_2} h_1^r$.

2.2 Blockchain

Blockchain is the underlying technology of a number of digital cryptocurrencies [10]. It is a decentralized, distributed ledger technology that offers transparency, security, and trust in a digital world plagued by vulnerabilities and mistrust [5].

At the heart of blockchain lies decentralization. Unlike traditional centralized systems, where a single authority controls data and transactions, blockchain relies on a network of nodes [2]. This decentralized structure ensures no single entity has unilateral control, making it resistant to censorship and tampering [13].

Every transaction on the blockchain is recorded in a public ledger, accessible to all participants in the network [11]. This transparency fosters trust, as anyone can verify the integrity of transactions independently [12].

In conclusion, blockchain technology represents a groundbreaking innovation with the potential to transform the way data is stored, shared, and transacted in various industries [19]. Its decentralized, transparent, and secure nature has led to a burgeoning body of research and practical implementations, shaping the future of digital transactions and trust mechanisms [18].

2.3 Support-Weighted Voting Algorithm

In order to reflect the level of voter support for a candidate, we introduce a support-weighted voting algorithm. The support-weighted voting algorithm is described in detail below. Suppose that the set of u candidates is denoted as $Can = \{t_1, t_2, ..., t_u\}$. For each candidate $t_i \in Can$, a total of n voters cast

their votes for it, and the corresponding support degree's are denoted as the set $M_{t,i} = \{m_{i,1}, m_{i,2}, ..., m_{i,n}\}$, where $m_{i,j}$ denotes the support degree paid by the jth voter of the ith candidate for that vote. The support score for each candidate was calculated according to the following formula.

$$score(t_i) = \frac{\sum_{j=1}^{n} m_{i,j}}{1 + Var} \tag{1}$$

$$Var_i = \frac{1}{n} \sum_{j=1}^{n} (\widehat{m_i} - m_{i,j})^2 \tag{2}$$

$$\widehat{m_i} = \frac{1}{n} \sum_{j=1}^{n} m_{i,j} \tag{3}$$

Equation 1 calculates the candidate's support score using the variance as a bias term. Equation 2 calculates the variance of support degree. Equation 3 calculates the everage value of support degree. This support voting algorithm with variance as a bias term is used mainly to solve the problem of choosing the best result among multiple candidates who have received votes with the same support degree. In the case of multiple candidates receiving the same total vote score, using variance as a bias term tends to select results with similar support votes from different voters. That is, we believe that the result of a vote in which the voters are more in agreement is more likely to be the accurate result. To facilitate subsequent calculations, we change the variance to the following form:

$$Var_i = \frac{\sum_{j=1}^{n} m_{i,j}^2}{n} - \frac{\left(\sum_{j=1}^{n} m_{i,j}\right)^2}{n^2} \tag{4}$$

3 Models and Design Goals

In this section, We focus on system model, threat model and design goals.

3.1 System Model

There exist five types of entities in the system: Blockchain, Voter, Encryption Server, Decryption Server and Canditate, which is shown in Fig. 1

- **Blockchain**: The blockchain is responsible for storing voting and ballot information, and the content of each ballot, as well as the corresponding support degree for each candidate, is recorded on the blockchain.
- **Canditates**: Candidates create ballots and submitting them to the blockchain in order to allow voters to cast their votes.
- **Voters**: Each voter can vote for any of the candidates on the ballot, and the ballot includes the candidate the voter supports and the support degree for that candidate. Then, the voter needs to encrypt and upload the voting information to the encryption server using the public key distributed by the decryption server.

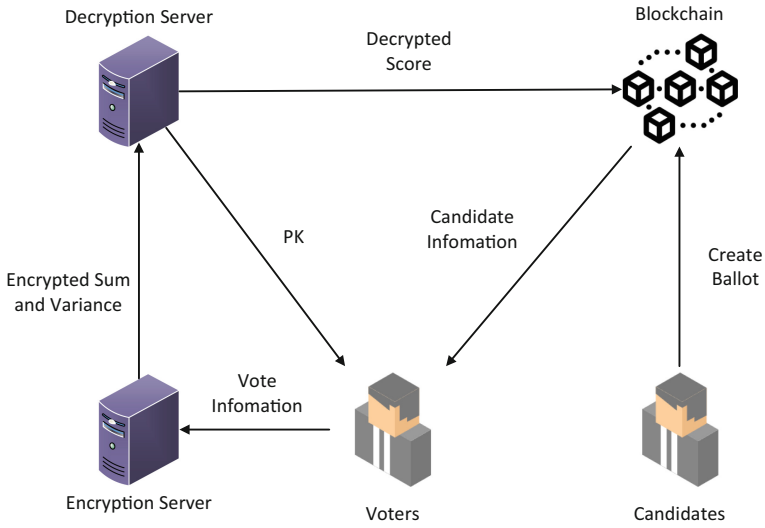

Fig. 1. System model.

- **Encryption Server**: The encryption server is responsible for aggregating the support degree uploaded by the voter over the ciphertext through homomorphic encryption. The sum and variance of the aggregated support degree is passed to the decryption server for further processing.
- **Decryption Server**: The decryption server decrypts the sum and variance of each candidate's support degree and then uses a support-weighted voting algorithm to calculate the total support degree for each candidate. After that, it is uploaded to the blockchain.

3.2 Threat Model and Security Requirements

In our threat model, Voters are assumed to be semi-honest (i.e., they follow the protocol while harboring a curiosity regarding each other's multi-dimensional data). Similarly, the encryption server is also considered semi-honest (i.e., it adheres to the protocol but exhibits curiosity towards the data shared by Voters). Furthermore, external attackers have the ability to eavesdrop on communications in order to obtain transmitted reports. In this context, we require that the decryption server cannot collude with any other entities.

In this paper, we regard the voting data of voters as their private information. We further emphasize the requirement that the aggregated support degree for each candidate, obtained through weighted aggregation, remain private until the voting results are made public. The following security requirements should be met:

- The encryption server and external attackers are unable to obtain the voting data of voters.
- External adversaries are unable to access detailed voting results before they are publicly disclosed.

3.3 Design Goal

Under the aforementioned system model, threat model, and security requirements, our design objective is to develop a privacy-preserving support-weighted voting algorithm. Specifically, we aim to achieve the following two desirable goals.

- The proposed privacy-preserving support-weighted voting algorithm should meet the defined security requirements. Failure to do so may result in potential violations of voters' privacy, leading to a lack of willingness among voters to provide their true support degree data during the voting process.
- Through the privacy-preserving support-weighted voting algorithm, the blockchain is able to obtain the accurate weighted aggregation result of support for each candidate.

4 The Proposed Methodology

To achieve the PSWS protocol, it is necessary to provide an algorithm that enables the computing party to calculate the weighted voting results without decrypting the voting data of the voters.

4.1 A Privacy-Preserving Support-Weighted Voting Algorithm

To fulfill the security requirements mentioned, we incorporated a homomorphic encryption scheme (such as BGN encryption) into the support-weighted voting algorithm, thus forming the proposed privacy-preserving support-weighted voting algorithm. The framework of the privacy-preserving support-weighted voting algorithm is illustrated in Fig. 2. Note that the figure shows the flow of one round of voting in the algorithm, i.e., it shows n voters voting for one candidate t_i(the ith candidate). If there are n voters who want to vote on u candidates, then it is only necessary to carry out the process in the diagram for the u round. This framework consists of five stages. In the first stage, the blockchain sends information about the candidates publicly to all voters. In the second stage, the decryption server sends the public key PK to all voters. In the third stage, the voters encrypt their votes using the public key PK and then package them and send them to the encryption server. In the fourth stage, the encryption server calculates the sum and the variance of the candidate's support degree on the ciphertext and sends them to the decryption server. In the fifth stage, the decryption server decrypts the sum as well as the variance of the candidate's support degree and calculates the candidate's total support *score* using Eq. 1 and sends it to the blockchain for public disclosure.

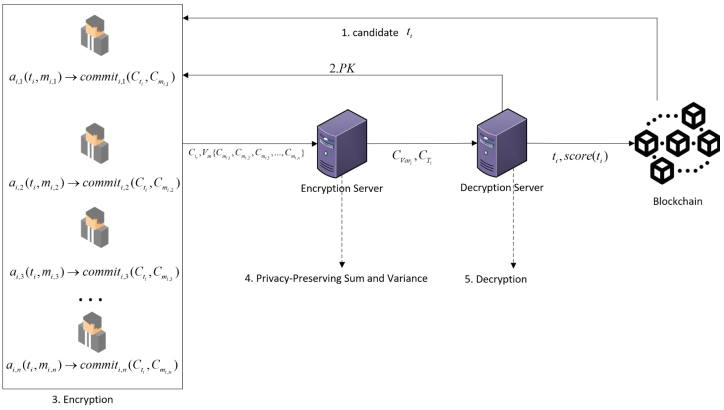

Fig. 2. The framework of the proposed voting algorithm

Algorithm 1. Encryption

Input: $a_{i,j} = (t_i, m_{i,j}), PK = (N, G, G_1, e, h, k)$
Output: $commit_{i,j} = (C_{t_i}, C_{m_{i,j}})$
1: Random $r \leftarrow \{0, 1, ..., N-1\}$
2: Compute $C_{t_i} = k^{Ct_i} h^r, C_{m_{i,j}} = k^{m_{i,j}} h^r$
3: Return $commit_{i,j} = (C_{t_i}, C_{m_{i,j}})$;

In Algorithm 1, $a_{i,j}$ denotes the jth vote for the ith candidate, where t_i denotes the ith candidate supported by the voters, and $m_{i,j}$ denotes the support degree paid by the jth voter of the ith candidate for that vote. The value of $m_{i,j}$ is in the range of [0,10]. 10 indicates full support for the candidate and 0 indicates no support at all for the candidate. If the voter wants to indicate no support at all for the candidate, then the support degree in the vote is entered as 0. PK is the public key of the BGN homomorphic encryption algorithm generated by the decryption server. This algorithm utilizes this public key to encrypt the voter's vote $a_{i,j}$. The algorithm's output $commit_{i,j}$ is an encrypted representation of a vote.

In Algorithm 2, C_{t_i} represents the ciphertext of a candidate obtained after encryption using the BGN algorithm in Algorithm 1. V_{m_i} denotes the collection of ciphertexts representing the support degrees received by this candidate. C_{T_i} represents the sum of all support degrees in V_{m_i} in ciphertext form. It can be further represented using the BGN homomorphic encryption algorithm as follows:

$$C_{T_i} = k^{\sum_{j=1}^{n} m_{i,j}} \times h^{\sum_{j=1}^{n} r_j + r} \tag{5}$$

where k and h are both generators used in the BGN homomorphic encryption algorithm. The $C_{i,j}$ in step 3 of Algorithm 2 represents the square of each support degree in V_{m_i} in ciphertext form. It can be further represented using the BGN

Algorithm 2. Privacy-Preserving Sum and Variance

Input: $C_{t_i}, V_{m_i} = \{C_{m_{i,1}}, C_{m_{i,2}}, C_{m_{i,3}}, ..., C_{m_{i,n}}\}$
Output: $Sum_i = (C_{t_i}, C_{Var_i}, C_{T_i})$
1: Compute $C_{T_i} = \prod_{j=1}^{n} C_{m_{i,j}}$
2: **for** $j = 1 \rightarrow n$ **do**
3: Compute $C_{i,j} = e(C_{m_{i,j}}, C_{m_{i,j}})$
4: **end for**
5: Compute $C_{Var_i} = \frac{[\![\frac{1}{n}]\!] \prod_{j=1}^{n} C_{i,j}}{[\![\frac{1}{n^2}]\!]^{e(C_T, C_T)}}$
6: Return $Sum_i = (C_{t_i}, C_{Var_i}, C_{T_i})$

homomorphic encryption algorithm as follows:

$$C_{i,j} = k_1^{m_{i,j}^2} h_1^{\widetilde{r}}. \tag{6}$$

The C_{Var_i} in step 4 of Algorithm 2 represents the variance of all support degrees in V_{m_i} in ciphertext form. It can be further represented using the BGN homomorphic encryption algorithm as follows:

$$C_{Var_i} = \frac{[\![\frac{1}{n}]\!] \, k^{\sum_{j=1}^{n} m^2{}_{i,j}} \times h^{\sum_{j=1}^{n} r_j + r}}{([\![\frac{1}{n^2}]\!] \, e(k^{\sum_{j=1}^{n} m_{i,j}} \times h^{\sum_{j=1}^{n} r_j + r}, k^{\sum_{j=1}^{n} m_{i,j}} \times h^{\sum_{j=1}^{n} r_j + r})}. \tag{7}$$

This algorithm is designed to calculate the variance of support degree C_{Var_i} and the sum of support degree C_{T_i} on the ciphertext for a candidate. Importantly, the entire process is performed on ciphertext, significantly enhancing the algorithm's security.

Algorithm 3. Decryption

Input: $Sum_i = (C_{t_i}, C_{Var_i}, C_{T_i}), SK = q_1$
Output: $s_i = (t_i, score(t_i))$
1: Compute $(C_{t_i})^{q_1} = (k^{q_1})^{t_i}$
2: Compute $(C_{Var_i})^{q_1} = (k^{q_1})^{Var_i}$
3: Compute $(C_{T_i})^{q_1} = (k^{q_1})^{T_i}$
4: $t_i, Var_i, T_i \leftarrow$Using Pollard's lambda algorithm[6] to compute the discrete logarithm with base k^{q_1}.
5: Compute $score(t_i) = \frac{T_i}{1+Var_i}$
6: Return $s_i = (t_i, score(t_i))$

Algorithm 3 is designed to decrypt the variance of support degree C_{Var_i} and the sum of support degreeC_{T_i} on the ciphertext for a candidate. Then, the

variance was used as a bias term in order to calculate the sum *score* of the weighted support degree of the candidate t_i.

Algorithm 4. Privacy-preserving Support-weighted Voting Algorithm

Input: $V_a = \{\{a_{1,1}, a_{1,2}, ..., a_{1,n}\}, \{a_{2,1}, a_{2,2}, ..., a_{2,n}\}, ..., \{a_{u,1}, a_{u,2}, ..., a_{u,n}\}\}, PK = (w, G, G_1, e, h, k), SK = q_1$
Output: $S = \{(t_1, score(t_1)), (t_2, score(t_2)), ..., (t_u, score(t_u))\}$
1: $S = \{\}, V_{m_i} = \{\}, Tmp = \{\}$
2: **for** $a_{i,j} = (t_i, m_{i,j}) \in V_a, i \in [1, u], j \in [1, n]$ **do**
3: $commit_{i,j} = Encryption(a_{i,j}, PK)$
4: $Append(V_{m_i}, C_{m_{i,j}})$
5: **end for**
6: $Tmp = \{(C_{t_i}, V_{m_i}) | (C_{t_1}, V_{m_1}), (C_{t_2}, V_{m_2}), ..., (C_{t_u}, V_{m_u})\}$
7: **for** $(C_{t_i}, V_{m_i}) \in Tmp$ **do**
8: $Sum_i = Privacy\text{-}Preserving\ Sum\ and\ Variance(C_{t_i}, V_{m_i})$
9: $s_i = Decryption(Sum_i, SK)$
10: $Append(S, s_i)$
11: **end for**
12: Return S

The complete privacy-preserving support-weighted voting algorithm is shown in Algorithm 4. V_a represents the set of votes of the voters, each round of voting has only one identified candidate, the number of rounds is the number of candidates u. This means that the system needs to allow all voters to vote in u rounds, and the candidate for each round is uniquely determined.

5 Security Property Analysis

According to our security requirements, we discuss how the proposed protocol PSWS ensures the privacy protection of the voters during the weighted aggregation of their votes.

Theorem 1. *The proposed protocol PSWS enables the privacy protection of voting data for the voters.*

Proof. As the voters first invoke the decryption server to generate a public key PK and encrypt their voting data with it, they subsequently send the encrypted ciphertext data to the encryption server for homomorphic weighted aggregation operations. Other nodes on the blockchain cannot access the voting data of the voters because they are unaware of the decryption server's private key SK. After aggregating the ciphertexts of the voters' support degrees using the Privacy-Preserving Support-Weighted Voting Algorithm, the encryption server sends the aggregated ciphertext to the decryption server. The voting information is encrypted under the BGN cryptosystem. The BGN cryptosystem based on compound-order bilinear groups is proven to be semantically safe against selected

plaintext attacks. In the computation on encryption server, many homomorphic operations are also encrypted under the BGN cryptosystem, so it is impossible to deduce any meaningful content.

Consequently, the proposed Privacy-Preserving Support-Weighted Voting Algorithm accomplishes the privacy protection of the voters' voting data.

6 Experiments

In this chapter, we empirically discussed the feasibility and practicality of the PSWS protocol through experimental analysis. In our simulation experiments, we employed BGN homomorphic encryption to cryptographically secure the support degrees obtained for each candidate by the encryption server. In other words, the ciphertext utilized in the Privacy-Preserving Support-Weighted Voting Algorithm was of the BGN cryptographic form. We assumed the existence of u candidates, with each candidate receiving n randomly generated votes of support. $Time_1$ and $Time_2$ represented the number of encryption and decryption operations in the BGN encryption scheme, respectively. We used Z_n to denote the computation time for weighted aggregation operations.

We randomly generated voting data for 5 candidates, with each candidate receiving 5 votes, each containing a support degree, for the purpose of conducting simulation experiments. We conducted the simulations on a computer equipped with a 3.40GHz 13th Gen Intel(R) Core(TM) i7-13700KF processor, 32GB of RAM, and a 64-bit operating system. For the BGN encryption scheme, we adopted a security parameter length of $256bits$. The experimental results indicated that $Time_1$ was 595ms, $Time_2$ was 69ms, and the Privacy-Preserving Support-Weighted Voting Algorithm could be completed within 1823ms. Among these, the running time for the aggregation operation, denoted as Z_n, was approximately 1210 ms. The support degrees obtained by the 5 candidates in this experiment are shown in Table 2. The voting results, as depicted in Fig. 3, indicate that Candidate 5 receives the highest total support degree, making Candidate 5 the winner of this vote.

Table 2. Case

candidate	support degree				
	supportdegree1	supportdegree2	supportdegree3	supportdegree4	supportdegree5
candidate1	7	9	2	4	9
candidate2	5	4	7	10	5
candidate3	8	3	7	8	1
candidate4	4	6	1	3	5
candidate5	5	7	7	4	8

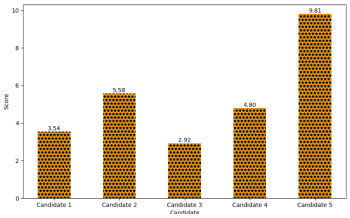

Fig. 3. Score for each candidate.

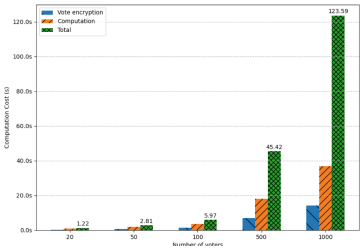

Fig. 4. Time overhead analysis of our protocol deployment ($t = 256bit$).

Figure 4 reflects the approximate linear correlation between the number of voters and the time overhead of communication. It is attributed to the number of voters will increase the number of operations and ciphertext.

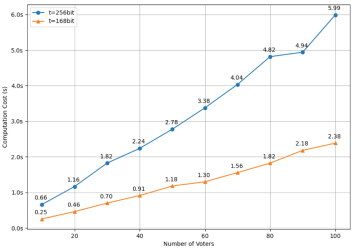

Fig. 5. Time overhead with different number of voters in different security parameter.

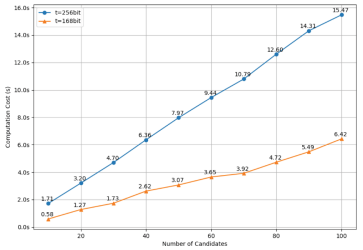

Fig. 6. Time overhead with different number of candidates in different security parameter.

The analysis of the time overhead is shown in Fig. 5 and Fig. 6. In Fig. 5, the number of candidates $n_c = 1$. In Fig. 6, the number of voters $n_v = 1$. Through the comparison of Fig. 5 and Fig. 6, it can be found that, since the impact of the security parameters on the encryption operation (i.e., complex encryption and increased ciphertext length), the time overhead is positively correlated with the security parameters.

7 Related Work

In [14], G. Revathy *et al.* proposed an electronic voting scheme for face recognition based on deep learning technology.Daria Golnarian *et al.* [7]proposed a novel trustless e-voting scheme based on blockchain technology that minimizes the intervention of authorities in the process, and voters can vote remotely using

their own mobile devices. Hemlata Kohad *et al.* [8]proposed the multiobjective genetic algorithm-based creation of a side-chain to enhance the scalability and performance of the blockchain-based e-voting system. Blockchain employs cryptographic techniques to secure data and transactions [17]. Information is stored in blocks, which are cryptographically linked in a chain. Once a block is added to the chain, altering its contents would require altering all subsequent blocks, making it nearly impossible to manipulate past data [20]. Based on the requirements analysis, Md Jobair Hossain Faruk *et al.* [3] proposed a biometric-enabled and hyperledger fabric-based voting framework to automate identity verification that will ensure transparency and security of electronic voting.

In contrast to the aforementioned work, our framework places a strong emphasis on conducting elections effectively and fairly while ensuring privacy protection throughout the entire process. The highlight of our work is the proposal of a voting algorithm framework that preserve the privacy of voting data while using weighted support degrees as the measurement criterion, and we implement this framework using homomorphic encryption methods.

8 Conclusion

This article proposes a privacy-preserving support-weighted sum protocol (PSWS). This protocol allows voters to cast their votes for the candidates they support. Voters can express their support for multiple candidates by assigning different support degrees to each candidate. The protocol employs the Privacy-Preserving Support-Weighted Voting Algorithm to calculate the votes, resulting in a fair and effective election outcome. Importantly, it ensures full privacy protection throughout the entire process, safeguarding the confidentiality of voters' voting data and preliminary results until the election results are allowed to be made public. To implement the privacy-preserving support-weighted voting algorithm, we utilize BGN homomorphic encryption technology. We prove that our proposed privacy-preserving support-weighted voting algorithm meets the security requirements within the threat model. Allowing supporters to vote for multiple candidates not only provides them with more choices but also enhances the fairness and usability of the voting system. However, it also results in increased computational overhead for privacy-preserving support-weighted voting algorithms. Future research can explore methods to enhance the computational efficiency of the current system, reduce computational costs, and mitigate the mentioned weaknesses. Furthermore, we plan to explore weighted voting with multidimensional data in more complex models, such as scenarios involving multiple evaluation criteria.

References

1. Chatterjee, U., Ray, S., Adhikari, S., Khan, M.K., Dasgupta, M.: Efficient and secure e-voting scheme using elliptic curve cryptography. Secur. Priv. **6**(3), e283 (2023)
2. Decker, C., Wattenhofer, R.: Information propagation in the bitcoin network. In: IEEE P2P 2013 Proceedings, pp. 1–10. IEEE (2013)
3. Faruk, M.J.H., et al.: Development of blockchain-based e-voting system: Requirements, design and security perspective. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 959–967. IEEE (2022)
4. Freeman, D.: Homomorphic encryption and the BGN cryptosystem (2011)
5. Gadekallu, T.R., et al.: Blockchain for the metaverse: a review. arXiv preprint.arXiv:2203.09738 (2022)
6. Gallant, R., Lambert, R., Vanstone, S.: Improving the parallelized pollard lambda search on anomalous binary curves. Math. Comput. **69**(232), 1699–1705 (2000)
7. Golnarian, D., Saedi, K., Bahrak, B.: A decentralized and trustless e-voting system based on blockchain technology. In: 2022 27th International Computer Conference, Computer Society of Iran (CSICC), pp. 1–7. IEEE (2022)
8. Kohad, H., Kumar, S., Ambhaikar, A.: Scalability of blockchain based e-voting system using multiobjective genetic algorithm with sharding. In: 2022 IEEE Delhi Section Conference (DELCON), pp. 1–4. IEEE (2022)
9. Kumar, D., Dwivedi, R.K.: Designing a secure e voting system using blockchain with efficient smart contract and consensus mechanism. In: International Conference on Advanced Network Technologies and Intelligent Computing, pp. 452–469. Springer (2022)
10. Monrat, A.A., Schelén, O., Andersson, K.: A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access **7**, 117134–117151 (2019)
11. Niranjanamurthy, M., Nithya, B., Jagannatha, S.: Analysis of blockchain technology: pros, cons and swot. Clust. Comput. **22**, 14743–14757 (2019)
12. Omran, Y., Henke, M., Heines, R., Hofmann, E.: Blockchain-driven supply chain finance: Towards a conceptual framework from a buyer perspective. Int. Purchas Supply Educ. Res. Assoc. **2017**, 15–15 (2017)
13. Rauchs, M., et al.: Distributed ledger technology systems: a conceptual framework. Available at SSRN 3230013 (2018)
14. Revathy, G., Raj, K.B., Kumar, A., Adibatti, S., Dahiya, P., Latha, T.: Investigation of e-voting system using face recognition using convolutional neural network (cnn). Theoret. Comput. Sci. **925**, 61–67 (2022)
15. Vijayakumar, K., Sriramasivam, T., Vigneshkumar, G., Senthil Kumar, G., Angel, T., Snehalatha, N.: Secure e-voting system using blockchain based on solidity technology. In: AIP Conference Proceedings. vol. 2516. AIP Publishing (2022)
16. Wahab, Y., : A framework for blockchain based e-voting system for Iraq. Int. J. Interact Mobile Technol. **16**(10) (2022)
17. Zhai, S., Yang, Y., Li, J., Qiu, C., Zhao, J.: Research on the application of cryptography on the blockchain. In: J. Phys.: Conference Series. vol. 1168, p. 032077. IOP Publishing (2019)
18. Zhang, M., Yang, M., Shen, G.: Ssbas-fa: A secure sealed-bid e-auction scheme with fair arbitration based on time-released blockchain. J. Syst. Architect. **129**, 102619 (2022) 10.1016/j.sysarc.2022.102619, https://www.sciencedirect.com/science/article/pii/S1383762122001503

19. Zhang, M., Yang, M., Shen, G., Xia, Z., Wang, Y.: A verifiable and privacy-preserving cloud mining pool selection scheme in blockchain of things. Inf. Sci. **623**, 293–310 (2023) 10.1016/j.ins.2022.11.169, https://www.sciencedirect.com/science/article/pii/S0020025522015225
20. Zhang, R., Xue, R., Liu, L.: Security and privacy on blockchain. ACM Comput. Surv. (CSUR) **52**(3), 1–34 (2019)