Secure Electronic voting protocol with partial Homomorphic encryption with Paillier algorithm

Vanipenta Pavan Kumar Reddy
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21215@bl.students.amrita.edu

Revanth Krishna Verma
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21155@bl.students.amrita.edu

Peta Sandeep
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21156@bl.students.amrita.edu

Rejeti Kartik
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21170@bl.students.amrita.edu

Kavitha C.R.*
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
cr_kavitha@blr.amrita.edu

***Abstract*—Voting on online platforms can cause many issues in the entire process. The choices given may not be great if they are recommended by the audience. The votes are displayed, sometimes with the names of the voters and with neither anonymity nor confidentiality kept in mind. The creation of a secure protocol for electronic voting is important for these reasons. The application of a homomorphic encryption system integrated into a e-voting software, where the voters can vote to which even choice they want and the vote is anonymously given and the admin will not know who gave the vote. The final tally will be displayed at the end of voting time and hence no live tracking is possible. The identity will be hidden for all, while the vote will be encrypted and sent to a tally server which aggregates them and finally decrypts the final results. This approach readily addresses challenges regarding security and transparency.**

***Index Terms*—Anonymity, Confidentiality, Homomorphic, Encryption, Decrypt**

I. Introduction

The electoral system in most countries uses a paper ballot system in some form or the other for entering the voter's choice and for tallying the final count to show the results. This method is very tedious and requires a lot of manpower and due to human error can cause flaws in the system. Hence an automatic electronic system can be useful to counter that issue. But, here too there are multiple issues. These include the integrity and confidentiality of the votes. Confidentiality is to ensure the anonymity of the person casting the vote. Integrity is to ensure the correct tally of the system and the proper count of votes. To ensure this, a proper system has to be designed to improve on these issues and build a smooth system.

This study ensures that the voting process in modern democratic systems is free from compromises and interferences by implementing the cryptographic technique known as Partial Homomorphic Encryption (PHE) [1] in the encryption and tallying of the votes in elections. The study applies multiple layers of Paillier homomorphic encryption for its efficiency as it has additive homomorphic nature well suited for this use. Through implementation of many layers of encryption and decryption of keys of different bit sizes, the system contributes to the increase of security and also guarantees that the votes are inclined with the highest level of security.

Analyzing the concepts that may influence voters' decisions, the study aims to accomplish the following objectives: secure vote encryption, homomorphic tallying, multi-round encryption, and parallel processing. That is, Paillier cryptosystem handles the additive homomorphic property, multiple encryption with different key pairs, parallel encryption decryption in Python multiprocessing and encrypted result compendium. This makes it possible for the sum of two encrypted numbers to have the decrypted result equal to the sum of the two normal numbers, thus making individual votes retrieved during the tallying stage to remain concealed.

II. Literature Survey

There have been many methods and protocols on electronic voting based on electronic voting based on the homomorphic system. Yuan et. al [2] addresses on how tampering and vote privacy violations in electronic voting systems. A new

*Corresponding Author
Dataset: https://catalog.data.gov/dataset/allegheny-county-election-results

scheme hat combines homomorphic encryption and blockchain technology to increase the performance of security and maintain the vote anomaly. The proposes method uses Pallier homomorphic encryption algorithm to maintain votes and block chain to decentralise the voting process and this also reduces the manipulation risks. The proposed methods shows 66.7% improvement in computational efficiency compared to other methods. In future more cryptographic measures are to added to make it possible to deploy in real world and observer the performance under different electoral conditions.

Wenlei et. al [3] propose a protocol for electronic voting on the basis of block chain. The protocol uses a homomorphic signature along with encryption where the voter generates his own public and private key on the Hyperledger platform using block chain system. The transaction involves a certification service and compared on the Ethereum block chain. It shows that the protocol is less efficient and uses higher data in Ethereum compared to the proposed system.

Xuechao et. al [4] propose a system for ranked choice voting based on homomorphism. The system applies ElGamal cryptography and assigns each rank of the candidates to a binary number and the vote is encrypted together and a verification is done to confirm submissions. The system has been compared to other systems, based on the time for casting votes, submission size and verification times. The proposed ranked choice systems performs better than other systems. The limitation of the system is being assumed that the authority is honest and it may not be true.

Ahmed et. al [5] propose a system for electronic voting using homomorphic cryptography. The homomorphic system consists of a voter authentication, key generation, verification and results decryption along with channels security [6]. This system has been compared at every stage of encryption, tallying and decryption using the HELib library and the implementation is applied on the servers of authentication, voting and voters. The results thus shows that mask calculation takes longer time than encryption or decryption and generation of votes takes longer tallying them. The intention in the future is to provide better results using cloud as the main data storage [7].

Saba et. al [8] proposes a new hybrid method for an electronic voting system with the help of a blockchain cryptography and homomorphic encryption to main the confidentiality. The proposed method uses a algorithm which generates digital signatures to maintain the vote confidentiality. It uses a homomorphic encryption to compute encypted votes and making the votes remain tamper-proof during aggregation. The results shows that the proposed system has made the security higher. Metrics such as number of successfully authenticated votes, vote tally accuracy are used. In future real world pilot testing is recommended to find how effective the system will be in different scenarios.

Zhan et. al [9] proposes an improved system that enhances security while giving efficient output. The proposed method develops a new crypt-analytic attack to identify the flaws in the existing system and combine proof of partial-knowledge protocols for added security during authentication. The method adopts different protocols to boost the vote's integrity and maintaining confidentiality. The results shows that proposed system reduces computation costs on the server side and increases the security measures against potential threats and also make sure voter anomaly is safe. In future additional cryptographic layers to be added to make more secure and implement the systems in real world to find the effectiveness under diverse voting conditions.

Zaid et. al [10] proposes a electronic voting system using homomorphic encryption to maintain vote secrecy and enable computation of vote total without decryption. The method leverages the cloud for purposes like storage and processing and maintains the scalability. The method performed well and handle multiple voting scenarios more efficiently. The paper also explains on how to validate the system practicality and robustness in real life electoral processes. In future the encryption should made more efficient and also reduces the computational cost.

Fan et. al [11] present a comprehensive analysis of electronic voting, highlighting and its benefits such as higher participation rates for more accurate representation of public opinion. It presents HSE-voting, a workable electronic voting system supported by homomorphic sign-cryption and the cryptographic tools. The technique is based on security technologies such as mix-net encryption, blind signature method, and homomorphic encryption, and is further validated by a comparative performance analysis. The four main components of the HSE-voting architecture are the Auditor, Voter, Election Process, and B Board. The study highlights the importance of election security and privacy [7], stating that e-voting has been used in at least 15 nations. By the integrating signature and encryption into a single step, the proposed approach streamlines the voting process and enables voters to confirm their votes on the B Board. The robustness of the method in protecting security and privacy is confirmed by a theoretical security analysis and that comes with theorems and proofs.

A systematic review methodology [12] is being used, to close the main knowledge gap in the homomorphic encryption which resulted in the methodical collection, evaluation, and integration of previous research. Merely 59 out of 418 peer-reviewed articles published from 2014 onwards that were found in databases such as ProQuest Central and IEEE journals met the requirements. The majority of the study focused on data security using homomorphic encryption, with the US and UK accounting for a significant portion of the focus. However, fewer than 38% of the studies provided explicit financing and research objectives. The 2014 was the peak year for publications, even though they were distributed geographically, with China, India, the US, and the UK leading the way. Due to the enormous public-key sizes some of ranging from 69 MB to 2.25 GB a decrease in efficiency was seen. However, the significant development of homomorphic approaches keeps driving progress in the field, particularly with regard to large data applications.

Bharati Raut et al. [13] suggest a novel solution to the

Votes	Prec. Count	Party	Dist. Type	Dist. Code	Contest Name	Cand. Num	Cand. Name	Vote for	Ref
90214	1319	DEM			Justice of the Supreme Court	1	David Wecht	3	
64654	1319	DEM			Justice of the Supreme Court	2	Christine Donohue	3	
24100	1319	DEM			Justice of the Supreme Court	3	Kevin M. Dougherty	3	
8583	1319	DEM			Justice of the Supreme Court	4	John Henry Foradora	3	
16571	1319	DEM			Justice of the Supreme Court	5	Anne E. Lazarus	3	
60606	1319	DEM			Justice of the Supreme Court	6	Dwayne D. Woodruff	3	
758	1319	DEM			Justice of the Supreme Court	7	WRITE-IN	3	
18552	1319	DEM			Judge of the Superior Court	1	Alice Beck Dubow	1	
78212	1319	DEM			Judge of the Superior Court	2	Robert Colville	1	
282	1319	DEM			Judge of the Superior Court	3	WRITE-IN	1	
9561	1319	DEM			Judge of the Commonwealth Court	1	Todd Eagen	1	
82451	1319	DEM			Judge of the Commonwealth Court	2	Michael Wojcik	1	
341	1319	DEM			Judge of the Commonwealth Court	3	WRITE-IN	1	
54965	1319	DEM			Judge of the Court of Common Pleas	1	Jennifer Staley McCrady	3	
23903	1319	DEM			Judge of the Court of Common Pleas	2	Richard Schubert	3	
25285	1319	DEM			Judge of the Court of Common Pleas	3	Rosemary Crawford	3	
37762	1319	DEM			Judge of the Court of Common Pleas	4	Dan Regan	3	
34711	1319	DEM			Judge of the Court of Common Pleas	5	Hugh Fitzpatrick McGough	3	
20270	1319	DEM			Judge of the Court of Common Pleas	6	Pauline Calabrese	3	
28525	1319	DEM			Judge of the Court of Common Pleas	7	P. J. Murray	3	
28428	1319	DEM			Judge of the Court of Common Pleas	8	William F. Caye, II	3	
328	1319	DEM			Judge of the Court of Common Pleas	9	WRITE-IN	3	

Fig. 1. Overview of the first dataset

security issues with online voting platforms. Voters' confidentiality and integrity are protected by the system, which uses homomorphic encryption. The final vote counting, decryption, and encryption of encrypted data [14] are made possible by the Paillier cryptosystem, which improves voting security. The entire framework for secure online voting is provided by the system architecture, which includes modules for voter verification, login, voter registration, and voting. The suggested approach seeks to improve online voting processes' dependability and trustworthiness by putting strong encryption and safe authentication procedures into place.

III. Methodology

A. Dataset Description

This study uses two datasets for electronic voting. The first is a 2015 primary election voting dataset, which consists of the vote id, party affiliation, contest name, rank of candidate, candidate name, and the vote given for a total of 2600 voters . This data is originally sorted with party affiliation and so the data is newly sorted based on the contest name. This is used to tally votes based on the contest name and hence per chunk the homomorphic encryption is done. Since the dataset is very large to compute all at once, a second dataset has been taken for testing purposes.

The second dataset is a synthetic dataset that consists is randomly generated voters for one contest such that it captures the maximum data of the voters and the tallying becomes faster. A fixed number of candidates has also been given namely Alice, Bob and Charlie.

The data is originally removed of any attributes that do not contribute to the vote tallying and hence needing any encryption. The data has been sorted based on the contest name and among all the attributes, the voter id the candidate order, its rank and the vote being cast are the immediately most important data to be encrypted.

B. Public Key Cryptography

Public key cryptography is a cybersecurity concept where there is a use of two different keys for encryption and decryption rather than using a single key shared across the sender and recipient. When trying to prove confidentiality, a public key of the recipient is used for encryption of the plain text and then during decryption only the intended receiver uses their private key on the cipher text. The public key can be accessible by everyone in the network [15], while the private key remains within the user. When trying to prove authentication using digital signature, the sender does encryption of the plain text with their private key and during decryption the recipient uses the sender's public key. The order of encryption from the sender is:

$$C = E_{Pu_{recv}}(E_{Pr_{send}}(P)) \tag{1}$$

and the order of decryption of the receiver is:

$$P = E_{Pu_{send}}(E_{Pr_{recv}}(C)) \tag{2}$$

where P is the plain text, C is the encrypted cipher text, (Pu_{send}, Pr_{send}) is the public, private key pair of the sender, and (Pu_{recv}, Pr_{recv}) is the public, private key pair of the receiver.

C. Partial Homomorphic encryption

Homomorphic encryption is a more of encryption where any computation is being carried out on the encrypted data to give an encrypted solution which later can be decrypted to prove the calculations that were made with the plain text data. The decryption is not a necessary step as the encrypted cipher text has all the information required to do any operation. This ensures the confidentiality of the data and hence during tallying the results are shown with plain text tallying. In homomorphic encryption, the tally is always encrypted as the decryption has to be done with the private key of the receiver. The homomorphic system can be done by many different ways. Fully homomorphic encryption can apply many different computations which may be arbitrary in nature. Example: Addition, Multiplication, Exclusive OR etc. are some of the operations or gates that can be applied all at once on the cipher text. Somewhat homomorphic encryption can apply upto a certain limit of operations depending on the criteria required. Partial homomorphic encryption can only apply one operation on the cipher text. Some of the opeartions that can be done are addition, multiplication, exclusivity and regeneration of text and each public key cryptography algorithm can apply certain homomorphic operations.

Paillier Homomorphic Encryption [16] is an asymmetric cryptography that is helpful for handling multiple classes using an additive schema to prove confidentiality and show an encrypted sum as the solution. It is also extremely helpful for the key generation for the public and private key pairs. The public key is applied as soon the vote is entered into the system and the tallying is done only after the homomorphic addition is allowed to do so. The useful part of homomorphic encryption is that it enables a secure addition without needing to decyrpt the data. This ensures the confidentiality of the votes throughout the process. The key generation is done so with the following steps:

Two large prime numbers p and q are chosen at random such that:

$$gcd(pq, (p-1)(q-1)) = 1 \tag{3}$$

where gcd stands for greatest common divisor, ensuring the equal length of the key. Then compute n and λ values such that:

$$n = pq \tag{4}$$

$$\lambda = lcm((p-1), (q-1)) \tag{5}$$

where lcm stands for the least common multiple of $p-1$ and $q-1$. Then select a random number g where $g \epsilon Z^*_{n^2}$ where the $Z^*_{n^2}$ indicates the set of integers that have a multiplicative inverse with n^2. Then the multiplicative inverse of n for g is verified using:

$$\mu = (\frac{x-1}{n}(g^{\lambda} mod(n^2)))^{-1} mod(n) \tag{6}$$

as μ is the modular multiplicative inverse and x is a random number. Hence we get the public key pair as (n, g) and the private key pair as (λ, μ).

D. Proposed System

The Paillier encryption generates the public private key pairs of sizes 1024, 2048 and 3072 bits. The 1024 bits provide a basic level of security and are sufficient for less critical applications. The 2048 bits are commonly used and offer a higher level of security and computational efficiency. The 3072 bits gives the security in high-stakes environments like electronic voting. The uses of each different key sizes allows the system to have a higher level of security. The public key is used for encryption and the private key is used for decryption. The different key sizes provides additional layers of security by applying encryption in that order. The votes are structured in order of contest name and this is helpful for analysis, tallying, encryption and decryption of votes. The dataset is split into smaller chunks of varying sizes based on the votes per contest, which can applied using a chunking function. This enables parallel processing enabling efficiency and scalability to tackle large datasets. The votes are encrypted parallel using the multiprocessing one chunk at a time. The system undergoes three rounds of encryption and decryption. The first round encrypts the data with public key of size 1024 bits. The second round, encryption is done with the public key of size 2048 bits. In the third round, encryption is done with the public key of size 3072 bits. This parallel encryption ensures large volume of votes can be processed quickly as it helps maintain confidentiality of the votes. The encrypted votes are then homomorphically tallyed and summation for each candidate without any decryption. The additive properties of the homomorphic system ensures that the tally is accurate and maintains the confidentiality throughout the counting process. The encrypted tally is then decrypted to show the final tally of the candidates. The decryption is done in the reverse order of the keys with 3072, 2048 and 1024 sizes in that order. A nested dictionary is used to store the vote counts of each candidate for each contest. This helps for easy retrieval and further analysis is possible. Finally the comparison is done among the candidates and winner is declared. The same is then used to verify the test data to verify the authenticity of the algorithm. The Fig. 2 shows the entire system workflow for secure online voting.

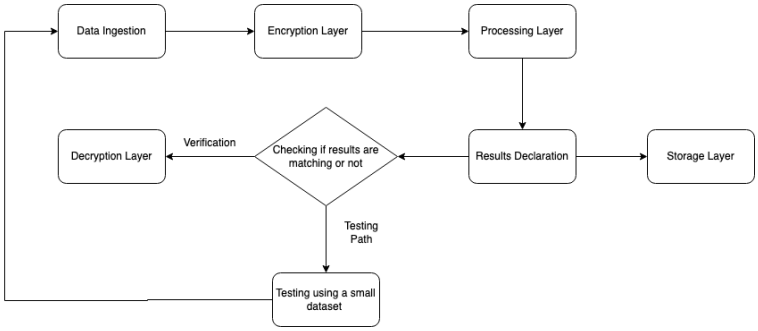

Fig. 2. System architecture of the system

The winner of the Auditor BELLEVUE contest is WRITE-IN with 3 votes.
The winner of the Auditor EAST DEER contest is WRITE-IN with 9 votes.
The winner of the Auditor ETNA contest is WRITE-IN with 0 votes.
The winner of the Auditor FAWN contest is WRITE-IN with 6 votes.
The winner of the Auditor FINDLAY contest is WRITE-IN with 2 votes.
The winner of the Auditor FORWARD contest is WRITE-IN with 6 votes.
The winner of the Auditor FRAZER contest is WRITE-IN with 2 votes.
The winner of the Auditor GLENFIELD contest is WRITE-IN with 6 votes.
The winner of the Auditor HARMAR contest is WRITE-IN with 2 votes.
The winner of the Auditor INDIANA contest is WRITE-IN with 4 votes.
The winner of the Auditor KILBUCK contest is WRITE-IN with 2 votes.
The winner of the Auditor MARSHALL contest is WRITE-IN with 2 votes.
The winner of the Auditor MOON contest is WRITE-IN with 2 votes.
The winner of the Auditor MT OLIVER contest is WRITE-IN with 5 votes.
The winner of the Auditor NORTH FAYETTE contest is WRITE-IN with 2 votes.
The winner of the Auditor OHARA contest is WRITE-IN with 4 votes.
The winner of the Auditor SOUTH PARK contest is WRITE-IN with 6 votes.
The winner of the Auditor TURTLE CREEK contest is WRITE-IN with 3 votes.
The winner of the Auditor WEST DEER contest is WRITE-IN with 4 votes.
The winner of the Auditor WEST ELIZABETH contest is WRITE-IN with 4 votes.

Fig. 3. Results of the first dataset

IV. RESULTS

The system then undergoes testing on both datasets. The first dataset consists of 600 contest names and hence the results of each contest along with the encrypted votes. The results of some of the contests are given as below in Fig. 3: for each of these contests the encrypted votes are as follows:

1) Encrypted total votes for WRITE-IN in Auditor BELLEVUE: 1014897762...
2) Encrypted total votes for WRITE-IN in Auditor EAST DEER: 1276823223...
3) Encrypted total votes for WRITE-IN in Auditor ETNA: 8165440776...
4) Encrypted total votes for WRITE-IN in Auditor FAWN: 8772786002
5) Encrypted total votes for WRITE-IN in Auditor FINDLAY: 1246218388
6) Encrypted total votes for WRITE-IN in Auditor FORWARD: 1159598443
7) Encrypted total votes for WRITE-IN in Auditor FRAZER: 1171772803
8) Encrypted total votes for WRITE-IN in Auditor GLENFIELD: 9017935122

	Votes	Prec. Count	Contest Name	Cand. Num	Cand. Name	Vote for
0	9	2	Contest A	2	Bob	2
1	16	1	Contest A	1	Alice	1
2	20	3	Contest A	3	Charlie	3
3	2	1	Contest A	1	Alice	1
4	13	1	Contest A	1	Alice	1
5	17	2	Contest A	2	Bob	2
6	3	2	Contest A	2	Bob	2
7	1	1	Contest A	1	Alice	1
8	5	3	Contest A	3	Charlie	3
9	15	1	Contest A	1	Alice	1
10	7	3	Contest A	3	Charlie	3
11	5	2	Contest A	2	Bob	2
12	19	2	Contest A	2	Bob	2
13	13	2	Contest A	2	Bob	2
14	20	1	Contest A	1	Alice	1
15	2	1	Contest A	1	Alice	1
16	14	3	Contest A	3	Charlie	3
17	19	1	Contest A	1	Alice	1
18	2	3	Contest A	3	Charlie	3
19	4	1	Contest A	1	Alice	1
20	2	2	Contest A	2	Bob	2
21	18	1	Contest A	1	Alice	1
22	8	2	Contest A	2	Bob	2
23	17	1	Contest A	1	Alice	1
24	20	2	Contest A	2	Bob	2

Fig. 4. Second dataset of the first simulation

	Votes	Prec. Count	Contest Name	Cand. Num	Cand. Name	Vote for
0	12	3	Contest A	3	Charlie	3
1	2	3	Contest A	3	Charlie	3
2	14	3	Contest A	3	Charlie	3
3	4	2	Contest A	2	Bob	2
4	2	2	Contest A	2	Bob	2
5	6	3	Contest A	3	Charlie	3
6	14	2	Contest A	2	Bob	2
7	7	1	Contest A	1	Alice	1
8	19	1	Contest A	1	Alice	1
9	5	3	Contest A	3	Charlie	3
10	18	2	Contest A	2	Bob	2
11	4	2	Contest A	2	Bob	2
12	4	1	Contest A	1	Alice	1
13	9	2	Contest A	2	Bob	2
14	20	3	Contest A	3	Charlie	3
15	9	2	Contest A	2	Bob	2
16	12	1	Contest A	1	Alice	1
17	9	3	Contest A	3	Charlie	3
18	10	3	Contest A	3	Charlie	3
19	20	1	Contest A	1	Alice	1
20	12	1	Contest A	1	Alice	1
21	3	2	Contest A	2	Bob	2
22	15	1	Contest A	1	Alice	1
23	4	2	Contest A	2	Bob	2
24	19	3	Contest A	3	Charlie	3

Fig. 5. Second dataset of the second simulation

These encrypted votes are the final tally of each candidate for what they have received in the election. The vote is a 3072 bits encrypted result represented in decimal form and each of the votes are unique to each other.

The second dataset has undergone two separate simulations to get the best randomization possible and to ensure that the system gives accurate results. Fig. 4 shows the randomized dataset in the first simulation. A manual counting of the votes is done to validate the results. It shows that Alice received 11 votes, Bob received 9 votes and Charlie received 5 votes. So, manually we can tell that Alice is the winner of this simulation. When applying the proposed system, we get result as "The winner of the Contest A contest is Alice with 11 votes." Along with this result the encrypted votes for each candidate as given below as:

1) Encrypted total votes for Bob in Contest A: 4522103766...
2) Encrypted total votes for Alice in Contest A: 1778470637...
3) Encrypted total votes for Charlie in Contest A: 5987806168...

The first simulation does give the accurate result of the votes. The encrypted total using paillier homomorphic system without the need for decryption shows the votes of Bob, Charlie and Alice are protected and it shows the confidentiality at work.

Fig. 5 shows the randomized dataset in the second simulation. A manual counting of the votes is done to validate the results. It shows that Alice received 7 votes, Bob received 9 votes and Charlie received 9 votes. So, manually we can tell that Bob and Charlie have tied as the winner of this simulation. When applying the proposed system, we get result as "The winner of the Contest A contest is Charlie with 9 votes." Since Charlie was first in the dataset, the system prints it as the winner, as in real world scenarios, ties are exceedingly rare and hence this can be taken as a special case. Along with this result the encrypted votes for each candidate as given below as:

1) Encrypted total votes for Bob in Contest A: 1110421452...
2) Encrypted total votes for Alice in Contest A: 4543552429...
3) Encrypted total votes for Charlie in Contest A: 1060419098...

The second simulation does give the relatively accurate result of the votes. The encrypted votes in the second simulation also differs from the first simulation showing the correctness of the encryption. It builds on this system to show that Bob and Charlie have gotten tie votes despite having same encryption formula. This gives the valid encryption for integrity as there is no scope of tampering with the votes as having showed in the simulation. This proves the validity of the system being tested on both datasets showing that it can prove confidentiality as all the people who have voted remain anonymous and only the tally is shown with the pin point accuracy. The protocol provides confidentiality to the votes of each individual vote, integrity of allowing one vote to be cast for the category/race, transparency allowing the public reveal of the tally without individuals worry about their own vote, non-repudiation which is participants cannot deny their participation and finally, verifiability so that voters can verify their vote has been counted.

V. Conclusion

This study demonstrates how the decryption scheme based on Paillier homomorphic encryption can be used for securely summing the votes in elections. It encompasses data preprocessing, data encryption, and parallelism, homomorphic computations, decryption, and final results computation. The main idea is to provide an example of homomorphic encryption that would preserve the confidentiality and the integrity of the elections, yet simultaneously enable computations on the encrypted data. The novel and the software have avenues for future improvements, such as extending the keyspace which is proportionate to the overall system scalability, key management and security, GUI interphase, compatibility with

other systems, legal and regulatory concerns, performance comparison, and evaluation with real-world datasets. Through these future directions, it is possible to further enhance the proposed system to ensure that the delivery of electronic voting system of different electoral context to be secure, effective and feasible. It is for this reason that research and development has to go on for the purpose of fine tuning the system for its use in the real world. The future work for the system can be an overall working system based on this protocol which can be tested in the real world.

REFERENCES

[1] H. M. Reddy, P. Sajimon, and S. Sankaran, "On the feasibility of homomorphic encryption for internet of things," in *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pp. 1–6, IEEE, 2022.

[2] K. Yuan, P. Sang, S. Zhang, X. Chen, W. Yang, and C. Jia, "An electronic voting scheme based on homomorphic encryption and decentralization," *PeerJ Computer Science*, vol. 9, p. e1649, 2023.

[3] W. Qu, L. Wu, W. Wang, Z. Liu, and H. Wang, "A electronic voting protocol based on blockchain and homomorphic signcryption," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 16, p. e5817, 2022.

[4] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han, "A secure verifiable ranked choice online voting system based on homomorphic encryption," *IEEE Access*, vol. 6, pp. 20506–20519, 2018.

[5] A. A. A. Aziz, H. N. Qunoo, and A. A. A. Samra, "Using homomorphic cryptographic solutions on e-voting systems," *International Journal of Computer Network and Information Security*, vol. 14, no. 1, p. 44, 2018.

[6] S. Chethana, S. S. Charan, V. Srihitha, D. Radha, and C. Kavitha, "Comparative analysis of password storage security using double secure hash algorithm," in *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, pp. 1–5, IEEE, 2022.

[7] S. Murthy and C. Kavitha, "Preserving data privacy in cloud using homomorphic encryption," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1131–1135, IEEE, 2019.

[8] S. A.-B. Salman, S. Al-Janabi, and A. M. Sagheer, "Valid blockchain-based e-voting using elliptic curve and homomorphic encryption.," *International Journal of Interactive Mobile Technologies*, vol. 16, no. 20, 2022.

[9] Y. Zhan, W. Zhao, C. Zhu, Z. Zhao, N. Yang, and B. Wang, "Efficient electronic voting system based on homomorphic encryption," *Electronics*, vol. 13, no. 2, p. 286, 2024.

[10] Z. Kartit, M. El Marraki, A. Azougaghe, and M. Belkasmi, "Towards a secure electronic voting in cloud computing environment using homomorphic encryption algorithm," *International Journal of Applied Engineering Research*, vol. 10, no. 16, pp. 37403–37408, 2015.

[11] X. Fan, T. Wu, Q. Zheng, Y. Chen, M. Alam, and X. Xiao, "Hse-voting: A secure high-efficiency electronic voting scheme based on homomorphic signcryption," *Future Generation Computer Systems*, vol. 111, pp. 754–762, 2020.

[12] M. Alloghani, M. M. Alani, D. Al-Jumeily, T. Baker, J. Mustafina, A. Hussain, and A. J. Aljaaf, "A systematic review on the status and progress of homomorphic encryption technologies," *Journal of Information Security and Applications*, vol. 48, p. 102362, 2019.

[13] B. Raut, M. Jagtap, S. Ghule, K. Jadhav, and S. Aundhakar, "Homomorphic encryption based online voting system," 2019.

[14] B. Kruthika, S. M. Rajagopal, C. Kavitha, *et al.*, "Homomorphic encryption for secure data analysis: A hybrid approach using pkcs1_oaep padding," in *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp. 481–485, IEEE, 2024.

[15] V. Sidharth and C. Kavitha, "Network intrusion detection system using stacking and boosting ensemble methods," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 357–363, IEEE, 2021.

[16] M. Bhavitha, K. Rakshitha, and S. M. Rajagopal, "Performance evaluation of aes, des, rsa, and paillier homomorphic for image security," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, pp. 1–5, IEEE, 2024.