Development of a Web Framework for E-Voting System

Roopa K,
Department of Electronics and Communication Engineering
The National Institute of Engineering
Mysuru-570008, Karnataka, India
roopakodnad@gmail.com

Gokul B S
L & T Technology Services Ltd.
Mysuru-570018, Karnataka, India
gokul.bs@ltts.com

Kaushik S Arakalgud
BitClass
Bengaluru- 560102, Karnataka, India
ska.kaushik@gmail.com

***Abstract*—In modern age, there are huge requirements for web applications like ticket booking for shows, reservation for travelling, ordering food, booking a cab, online payment transactions and so on. This paper discusses a web application designed for e-voting system using open source tools like Django web framework and MySQL database. Python language is used for programming. Voting confidentiality is provided through Paillier homomorphic algorithm. The web application is designed to provide voter, administrative and electoral support. In the implemented system, the electoral head (EH) controls the voting process and is the trusted entity. The Paillier algorithm along with its web interface was simulated and tested successfully.**

Keywords—E-voting, Web framework, Paillier algorithm, Confidentiality, Django, MySQL.

I. INTRODUCTION

Recent developments in communication and cryptosystems are leading to online voting techniques as an alternative to conventional elections. E-Voting solutions usually aim at increasing participation and improving the turnout during elections by addressing challenges associated with traditional voting systems like transportation. The main objective is to provide easier access to electronic-voting system (EVS) without compromising confidentiality and security of votes casted. This can be done by Paillier algorithm. Paillier cryptosystem [[1], [2], [3], [4], [5]] provides a homomorphic encryption technique to maintain confidentiality in privacy-preserving e-voting systems. Voting data involves the identity of the candidate that is chosen by a voter. The voter identity details like name, gender, address and age are also collected for the purpose of carrying out voting analytics. The software solution is implemented using Django framework [6].

This paper mainly describes the web application development for an EVS. Only a brief description of Paillier algorithm and its homomorphic properties are considered here for the sake of clarity and continuity. The algorithmic implementation is prepared as a separate article [7]. An election specific public-private key pair is generated by the electoral head (EH). Decryption can be done by the EH alone as he possesses the secret private key.

II. RELATED WORK

The e-voting framework described in [8] focuses on the national elections held in Estonia. Estonia has been using e-voting since 2005 [9]. Each person who has the right to vote in Estonia can cast his vote in a secured way through internet [10]. A voter uses the voter application installed on his computer for voting. Voting process takes place in two stages, viz., the identification and the voting. In the identification stage, the voter is identified and voting options are sent to the voter. In the voting stage, the voter makes his choice among the candidates displayed in the list. The voter application encrypts voter's choice along with a random number with the help of the public key. Thus the secrecy of voting is ensured. An asymmetric crypto algorithm is used, so that votes encrypted cannot be decrypted with the same key. Adding a random number to the vote ensures the secrecy of the votes. The cipher texts of the votes cast for the same candidate are different.

[11] describes an e-voting web application system. [12] describes an e-voting system with smart phone using mobile application.

The work presented in this paper is different from other related works as it is simple and suitable for small, institutional level elections. A device with an internet connection is the requirement. There is no need for the voter to visit the polling station. Hence it is suitable for use during pandemic situations. The system presented here reduces the time and monetary resources involved in physical voting systems without compromising the security.

III. METHODOLOGY

PAILLIER HOMOMORPHIC PROPERTY FOR ADDITION:

This property is used for getting the voting results in a secured manner. According to this property, when two cipher texts are multiplied, the result decrypts to the sum of their plaintexts as shown in (1).

$$D_{priv}\left(E_{pub}(m1).E_{pub}(m2)\ \ mod n^2\right) = m1 + m2\ \ \text{mod} n \tag{1}$$

Here, 'E' stands for encryption operation, 'D' stands for decryption operation, 'priv' refers to the private key, 'pub' refers to the public key. m1, m2 are the two input messages, corresponding to two votes in this case. 'n' is a part of the Paillier algorithm [[1], [2], [3], [4], [5]].

Use of Paillier algorithm for e-voting system has the following advantage:

This is an asymmetric encryption technique. Computing the private key or the message even if the public key or the cipher text is known, is an infeasible task in terms of time required and the large computations involved. Because, this requires calculation of λ (i. e., λ = (p-1) (q-1)) which in turn requires p and q values. This requires factoring the product of two large prime numbers which is a difficult task.

DOI: 10.5281/zenodo.6136660
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

Fig.1 shows the block diagram of use cases of Voter, EH and Administrator (Admin). Fig.2, Fig.3 and Fig.4 show the flow of steps for the Voter, EH and Admin respectively. This model version is suitable for small elections. It can be extended to bigger elections using additional authentication procedures.

Each user registers in the system with their credentials before the election. Admin adds the candidates contesting for the election, and the EH with their details into the database which will be used for verification at the time of voting as shown in Fig. 2.

For each contesting candidate in the system, there will be a message or value assigned as shown in Fig. 3. As each voter votes, a unique cipher text will be generated against the candidate to whom the vote is cast. Thus, the confidentiality is maintained as the cipher text is never repeated across voters (even if voted for the same candidate) due to the random number generated in Paillier algorithm. The EH alone is the authorized entity to decrypt the number of votes. The authentication system allows only verified access to the voting panel and eliminates security vulnerabilities like multiple voting and masquerade attacks [13].

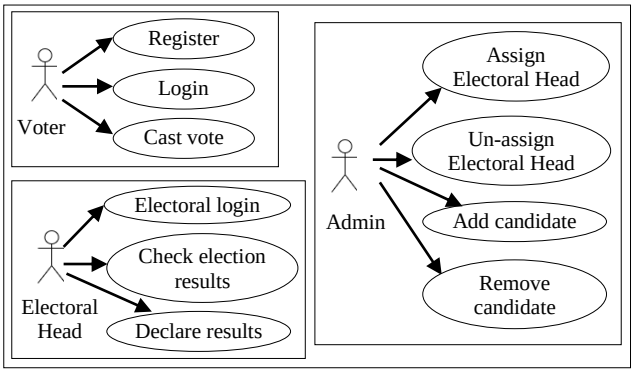


Fig. 1. Block diagram of use cases of voter, Electoral Head and Admin.

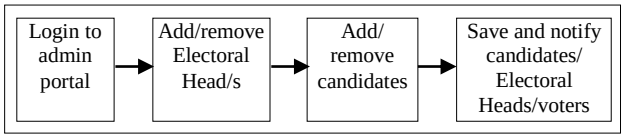


Fig. 2. Block diagram of Admin's journey

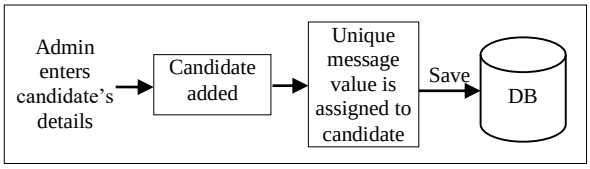


Fig. 3. Candidate message allocation

Fig. 4 shows the workflow for voter's journey. The voter logs-in or registers (first time) in the system. A voting panel will be displayed to the voter with the contesting candidates' details. The voter can then choose the candidate and input his/her preference. The candidate's message then gets encrypted and a unique cipher text is generated. This cipher text is passed on to the server where the voter count of candidate is updated by multiplying the received cipher text with candidate's existing cipher text. The voter is then shown a success message and is disallowed to vote further. This mechanism is shown in Fig. 5.

At the end of election, the EH will login to the portal. An option to freeze and declare results will be shown to the EH. On choosing this option, voting will be disabled and number of votes received by each candidate will be decrypted and displayed as shown in Fig. 6.

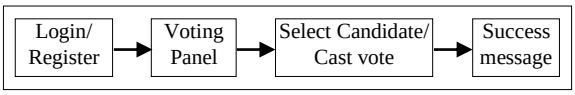


Fig. 4. Block diagram of voter's journey.

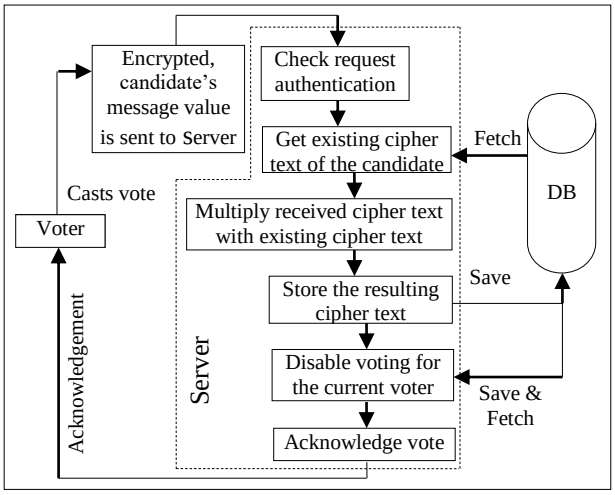


Fig. 5. Voting mechanism

Fig. 6. Block diagram of EH's journey

IV. IMPLEMENTATION

Using the Paillier cryptosystem since the cipher text is different for the same candidate it avoids any third party from knowing the votes before the election results are declared. Even admin will not know which voter has voted for which candidate.

To access the implemented web portal of e-voting system, the user is required to login from a common sign in page. The users will be redirected to their respective portals based on their role once authenticated successfully. A voter is allowed to login to the system at any point during the election, post which the login will be disabled.

The web application is built using Django, which is a python based free and open-source web framework. MySQL [14], an open-source relational database management system is used for storing and retrieving user's data.

Django follows the model-template-views architectural pattern and its primary goal is to ease the process involved in creation of complex database driven websites.

DOI: 10.5281/zenodo.6136660
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

For the development phase, a lightweight web server provided by Django was used. A web server like Apache [[15], [16]]/Gunicorn [[17], [18]] is recommended for production.

The frontend is designed using hyper text markup language (HTML), cascading style sheets (CSS), JavaScript and with the help of Django's template language.

The database of the voters will be stored apriori in the server. The identity of the voter will be verified and access to the voting panel will be enabled on his system. The voter can then cast his vote.

The encryption/decryption mechanism implemented in the system will ensure privacy and data integrity. Analytics are provided without revealing the voter's identity.

V. RESULTS

The output of the implementation is described in this section.

Fig. 7 shows the screen shot of the common sign in page. Fig. 8 shows the design of web front-end for the voter. On the day of election, voting link will be shared to the voter. On completion of voting, a success message will be displayed to the voter.

Voter can change his password during election process by visiting 'My Profile' page.

Fig. 9 shows the screenshot of web-administrative controls which appear in the Admin pageview. Admin is the authority to add and remove voters, candidates and EH to the system. Pie charts showing number of male and female candidates and voters along with total number of voters and candidates in the election are displayed in admin portal for analytics.

Fig. 10 shows the EH pageview. EH will have permissions to check the vote count of a candidate after the election process. The votes are decrypted only when 'Results' option is selected by EH. Updated vote count of candidates will be displayed in 'Results' page. Similar to voter, EH will be allowed to change his password in 'My Profile' page during election process. If necessary, two electoral heads may be allocated as shown in the screenshot. Both of them will have equal privilege.

To check the updated results, EH selects 'Details' for a candidate. Then it opens a page to 'update vote count' as shown in Fig. 11. The cipher texts are multiplied and saved in the cipher text column of that candidate as shown in Fig. 12.

When EH clicks on "update vote count", the decryption process starts. Then EH gets the sum of the votes using Paillier homomorphic property. This sum is divided by the candidate's message to get the number of votes for that candidate. In Fig. 12, the message value 67 is assigned to the candidate 1 as an example.

VI. CONCLUSION

This paper discussed the development of a web framework for e-voting system using open source softwares like Django and MySQL database. Using the web portal, voter can cast his vote. The web framework is interfaced to the Paillier algorithm that is used to obtain the polling result in a confidential manner through additive homomorphic property. Results and voting analytics are declared by the electoral head on processing of the votes. Future scope of this work can be to extend the web application for bigger elections and to provide additional authentication features.

REFERENCES

[1] https://en.wikipedia.org/wiki/Paillier_cryptosystem, Accessed on Sep. 2021.

[2] Bhumika Patel and Dharmendra Bhatti, "Homomorphic Encryption: Privacy Preserving Amicable E-voting System", International Journal of Computer Sciences and Engineering, Vol.-7, Issue-12, Dec 2019, E-ISSN: 2347-2693.

[3] T. Sridokmai and S. Prakancharoen, "The homomorphic other property of Paillier cryptosystem", 2015 International Conference on Science and Technology (TICST), 2015, pp. 356-359, doi: 10.1109/TICST.2015.7369385.

[4] S. M. Anggriane, S. M. Nasution, F. Azmi, "Advanced e-voting system using Paillier homomorphic encryption algorithm" 2016 International Conference on Informatics and Computing (ICIC), 2016, pp. 338-342, doi: 10.1109/IAC.2016.7905741.

[5] T. V. J. Shihab and P. I. Liji, "Simple and secure internet voting scheme using generalized Paillier cryptosystem", 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2017, pp. 551-557, doi: 10.1109/ICICICT1.2017.8342623.

[6] https://www.djangoproject.com/, Accessed on August 2021.

[7] Roopa K, Gokul B S, S Kaushik Arakalgud, "Use case of Paillier homomorphic algorithm for electronic-voting systems", 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques, GSSSIETW, Mysuru, 10th and 11th December 2021, in press.

[8] General Framework of Electronic Voting and Implementation thereof at National Elections in Estonia, Tallinn 2016, Available at, https://www.venice.coe.int/files/13EMB/13EMB_Priit_Vinkel.pdf, Accessed on Oct 2021.

[9] https://e-estonia.com/solutions/e-governance/i-voting/, Accessed on Oct. 2021.

[10] https://www.valimised.ee/, Accessed on Oct 2021.

[11] R. Chhabra, U. Vohra, V. Khanna, A. Verman, P. Tanwar and B. Kumar, "The Next Gen Election: Design and Development of E-Voting Web Application," 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020, pp. 536-541, doi: 10.1109/ICCES48766.2020.9138050.

[12] G. Kalaiyarasi, K. Balaji, T. Narmadha and V. Naveen, "E-Voting System In Smart Phone Using Mobile Application," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 1466-1469, doi: 10.1109/ICACCS48705.2020.9074289.

[13] William Stallings, Cryptography and Network Security, 4th ed., Pearson Education, 2011.

[14] https://www.mysql.com/, Accessed on Oct. 2021.

[15] https://www.apachefriends.org/index.html, Accessed on Oct. 2021.

[16] https://httpd.apache.org/, Accessed on Oct. 2021.

[17] https://gunicorn.org/

[18] https://docs.gunicorn.org/

DOI: 10.5281/zenodo.6136660
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

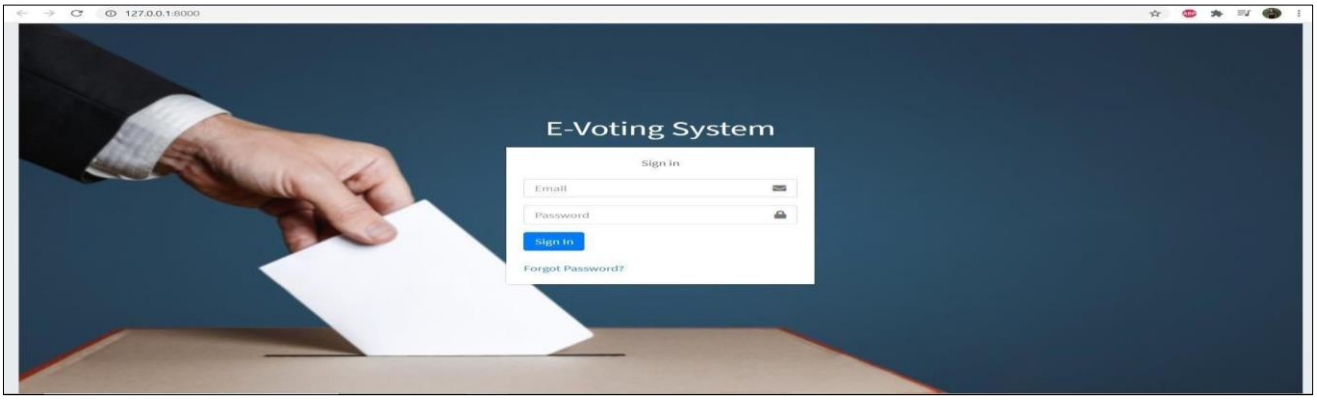


Fig. 7. Screenshot of common sign in page.

Fig. 8. Screenshot of Voter page

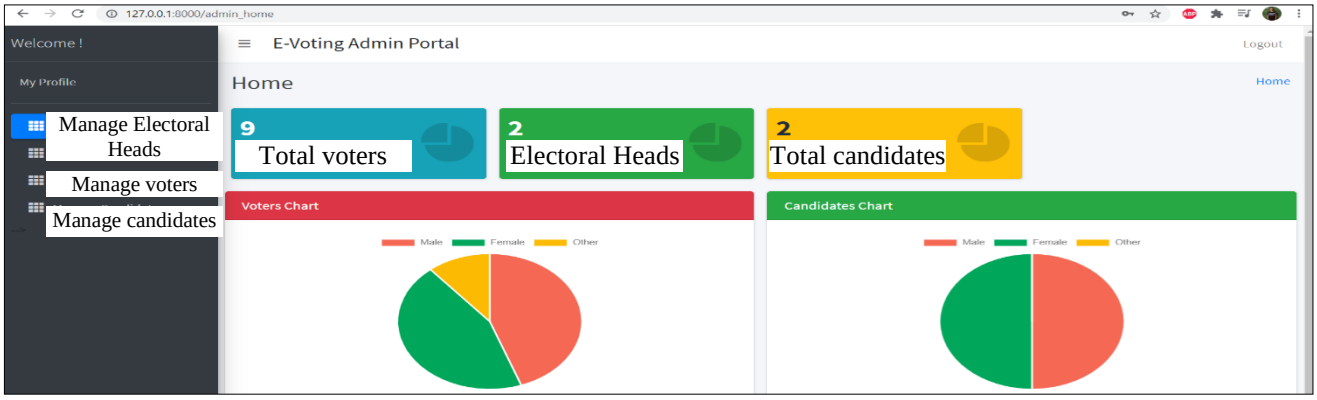
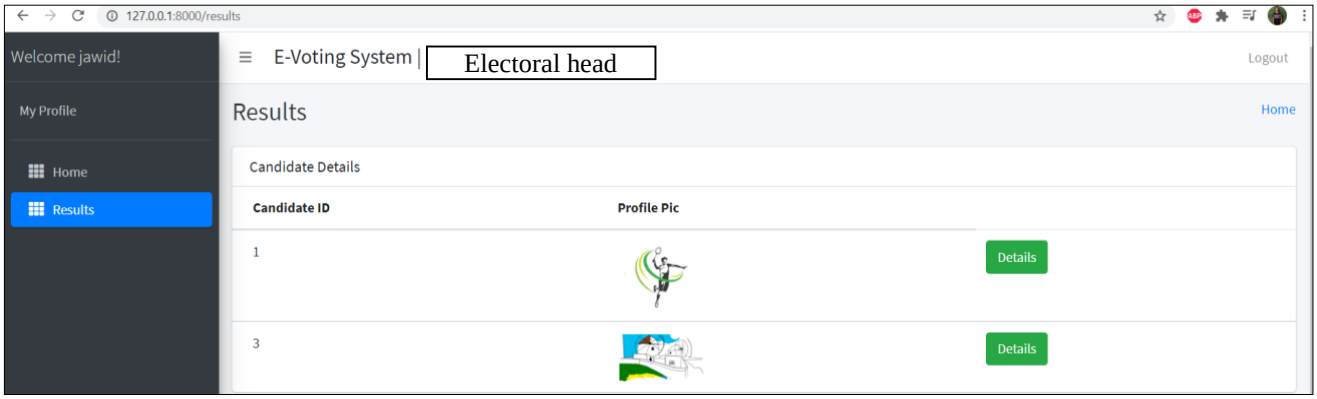


Fig. 9. Screenshot of Admin page



DOI: 10.5281/zenodo.6136660
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

Fig. 10. Screenshot of electoral head page

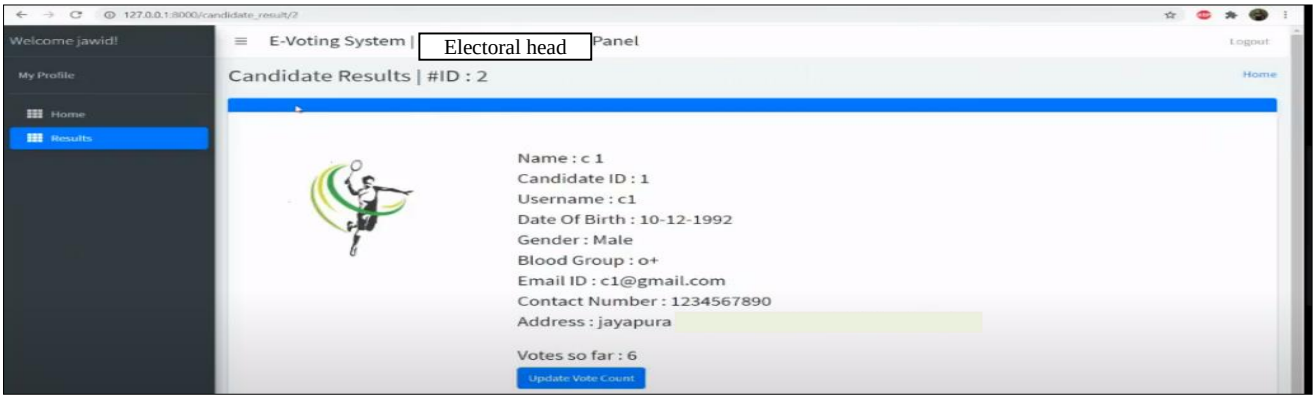


Fig. 11. Example of updating vote count of a candidate

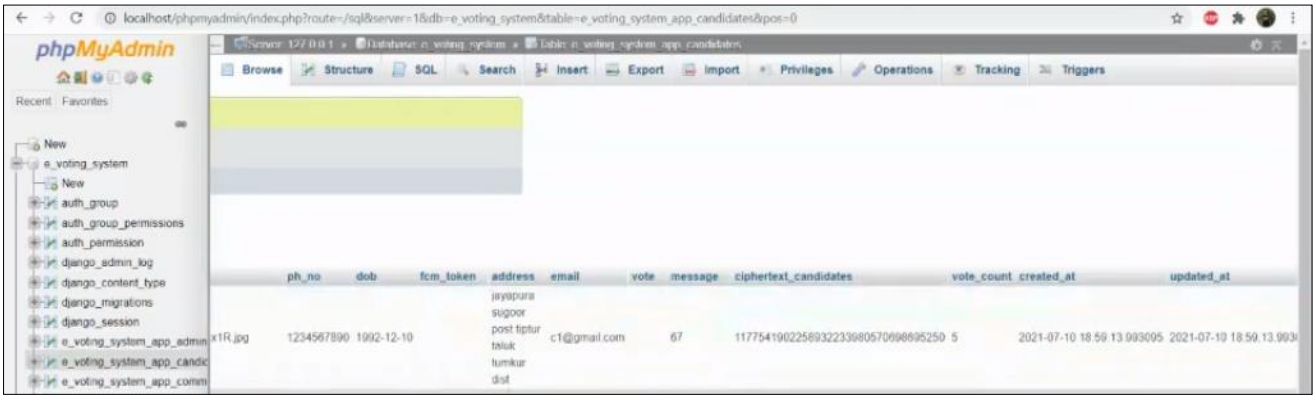


Fig. 12. Screenshot showing the cipher text of the total votes cast for a candidate

DOI: 10.5281/zenodo.6136660
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam