PROCEEDINGS OF
INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS
https://proceedings.icisna.org/
3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Secure Voting System Using FPGA

Kiruthick K M[1], Rushindra K R[2], Karthikeyan T[3], Kaveri Hatti[4]

[1-4] Department of Electronics and Communication Engineering
Amrita School of Engineering, Bengaluru – 560035
Amrita Vishwa Vidyapeetham, India

[1]bl.en.u4ece23221@bl.studenta.amrita.edu, ORCID: 0009-0009-8256-6561
[2]bl.en.u4ece23140@bl.studenta.amrita.edu, ORCID: 0009-0009-8143-5105
[3]bl.en.u4ece23220@bl.studenta.amrita.edu, ORCID: 0009-0007-0815-8004
[4]h_kaveri@blr.amrita.edu, ORCID: 0000-0002-8973-5534

***Abstract*— The project is a proposal of a secure and reliable electronic voting system that will be made by the use of ESP32 microcontroller, keypad input module, and LCD interface. It mainly aims to make the process of voting transparent, accurate, and secure through the verification and management of the data in the clouds. Voter identification is done through the Aadhaar based credentials with password based login, providing a highly effective two-factor authentication system. The system will ensure that no votes are counted twice, as each vote will be associated with a specific Aadhaar ID and always have real-time status updates stored on the cloud. Once verified, the votes are encrypted with the aid of hardware-supported encryption chip executed on FPGA and sent safely to a cloud database such that the stored data cannot be easily changed, and can only be accessed via the authorized medium. To boost confidence among the voters, an SMS or OTP message of confirmation is automatically sent to the voter when a vote has been successfully registered. The proposed system provides end-to-end security, endures the privacy of the voter, and real-time vote counting is accurate. This solution is a modern solution to use in place of traditional methods of voting by use of IoT technology and secure cloud services to have a scalable, modern, and reliable approach to the same.**

***Keywords*—Electronic voting system, ESP32, Aadhaar verification, Cloud authentication, IoT security, Encrypted voting, Duplication vote prevention, OTP confirmation, Secure data storage.**

I. Introduction

The use of electronic systems of voting is crucial to the contemporary states of democracy, but the issue of security and reliability remains. Most of the current systems also do not have a good voter authentication, data integrity and privacy. As more people go digital, elections need to have secure, transparent and efficient voting provisions that resist fraud without compromising of voter confidence and reliability [1]–[4].

Voter impersonation is a significant problem that exists in the contemporary systems of voting because of the poor check of identity. Tampering of data also poses an additional challenge to the integrity of an election, as it allows a manipulation of stored or transmitted votes. Also, the absence of the vote confidentiality weakens the privacy of voters, which may affect voter turnout and compromise the election results credibility [2], [3], [5]. In order to overcome these problems, this paper will suggest a secure FPGA based-voting system, which uses AES encryption, password and Aadhaar-based authentication to secure confidentiality.

The rest of the paper is structured in the following way: Section II entails the literature survey, Section III details the methodology, Section IV addresses the implementation and results obtained, and Section V is a concluding part of the paper [1], [2], [4].

II. Literature survey

The recently provided research works have taken much attention on enhancing the security and reliability of electronic voting systems by employing multi-factor authentication and cryptographic security. Voter impersonation is greatly mitigated and better voter confidence is ensured due to the ability of biometric based voting systems to be undertaken using fingerprint authentication and SMS confirmation mechanism [1], [3]. Authentication is further enhanced with Aadhaar based identity verificationto ensure that there is no voting twice and unauthorized channel access [5].

Most of the researches have embraced cryptographic methods to guarantee confidentiality of votes and data integrity through the application of the Advanced encryption Standard (AES) coupled with OTP validation, which affords a high level of resistance against data corruption in transit and storage [2].

New voting systems combining machine learning, blockchain, and cryptography enhance the transparency, decentralization, and auditability of the voting systems [4], [6]. FPGA and VLSI-based AES, LFSR, and hybrid encryption algorithms provide hardware-oriented security solutions where high-speed and low power consumption software is suitable as well as differentiating a secure environment on real-time voting systems [7] to [9]. The technical research and documentation also attest to the essence of authenticated encryption, scalable system architecture, and key security in terms of reliable electronic voting systems [10], [11].

III. Methodology

A. Single-Ward Voting, Authentication, and Vote Casting Architecture

The voter enters 12-digit Aadhaar number first. Then the password is entered. Both values are checked with the voter data stored in the cloud. If the details are correct, the system allows the voter to continue. If the details are wrong, the voting process is stopped. After verification, the system shows the candidate names on the display and the voter selects one candidate using the keypad.

After candidate selection, the system checks the vote status of the voter. This status is already stored in the database. If the voter has voted earlier, the system does not allow voting again. If not voted, an OTP is sent to the registered mobile number. The vote is accepted only after the correct OTP is entered.

If the OTP entered is correct ,then the system sends back another SMS to the registered mobile number stating that the vote is successfully saved. After this process the votes come to the next block , the encryption module.Fig. 1. illustrates the functional flow of the proposed electronic voting system for a single voting booth or ward.

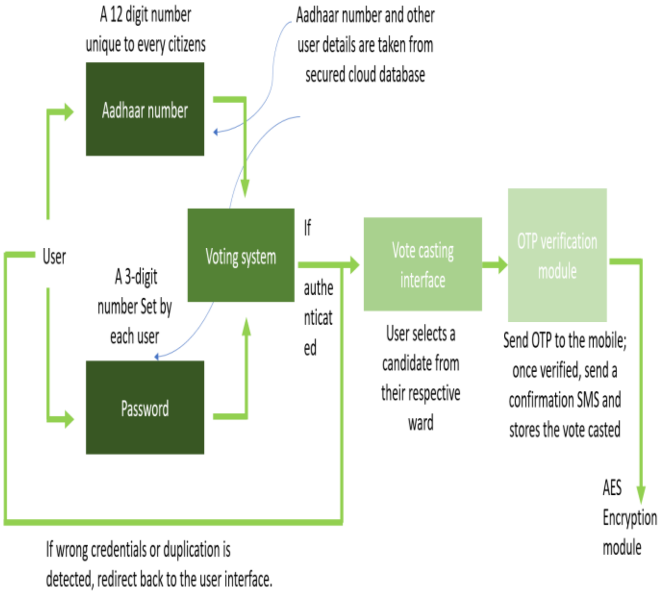

A 12 digit number unique to every citizens
Aadhaar number and other user details are taken from secured cloud database
Aadhaar number
User
Voting system
A 3-digit number Set by each user
Password
If authenticated
Vote casting interface
User selects a candidate from their respective ward
OTP verification module
Send OTP to the mobile; once verified, send a confirmation SMS and stores the vote casted
AES Encryption module
If wrong credentials or duplication is detected, redirect back to the user interface.

Fig. 1 Authentication and Vote Casting Flow of the Proposed Voting System

B. Secure Vote Encryption, Decryption, and Result Processing Architecture

The vote from the voter is taken and added with other votes, then the votes are collected together. The collected data is prepared for storage. This step is done before saving the vote.

After this, the vote data is encrypted using the AES algorithm. The encrypted data is stored. The data cannot be read directly. Only authorized persons who have the correct key can decrypt the data and see the result. This keeps the voting result safe. Fig. 2 shows what happens to the vote after it is selected.

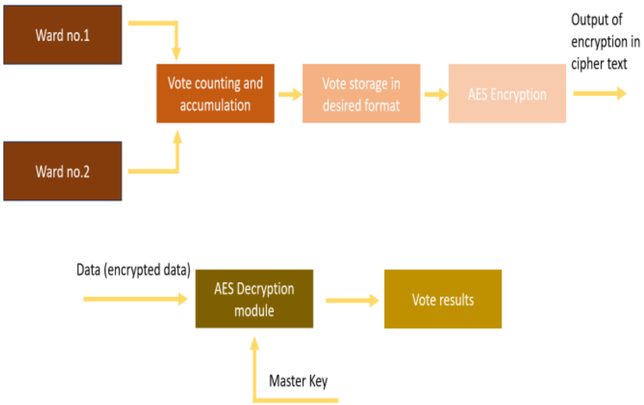


Fig. 2 Vote Accumulation, AES Encryption, and Result Decryption Process

IV. Implementation and Results

A. End-to-End Voting System in Wokwi Simulator

The ESP32 is used as the main controller. A 4X4keypad is used for input and an 16X2 LCD is used for display. The ESP32 connects to the cloud using Wi-Fi. This setup is used to test the working of the voting process. Fig. 3 shows the simulation setup of the voting system.

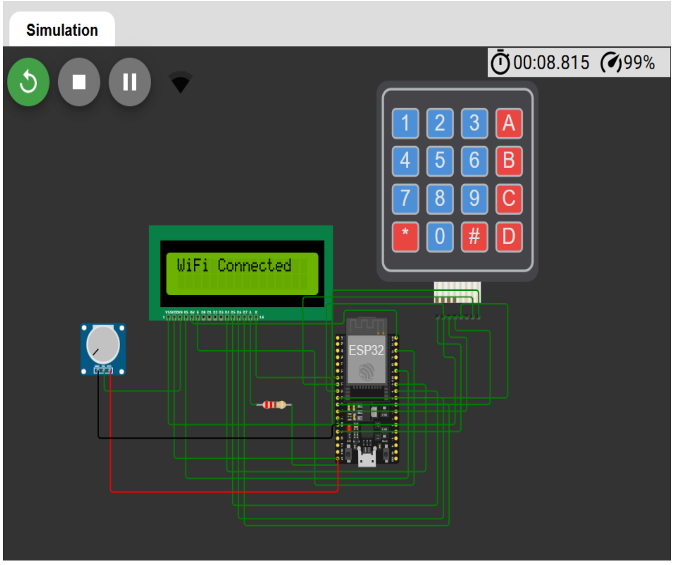
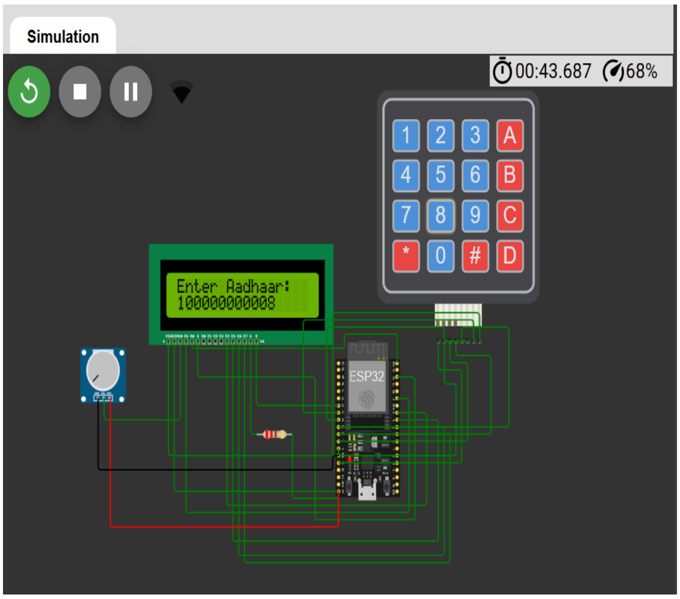


Fig. 3 ESP32-Based Hardware Simulation Setup for Secure Voting System

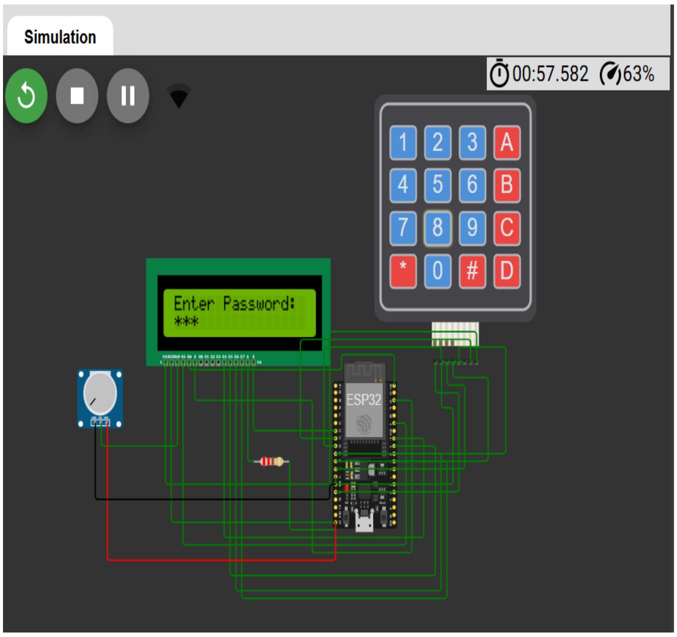
The voter enters the Aadhaar number using the keypad. The entered number is displayed on the LCD. This Aadhaar number is used to identify the voter. The system uses this value to check the voter details from the cloud. Fig. 4 shows the Aadhaar number entry stage.



Enter Aadhaar:
100000000008

Fig. 4 Aadhaar Number Entry Stage

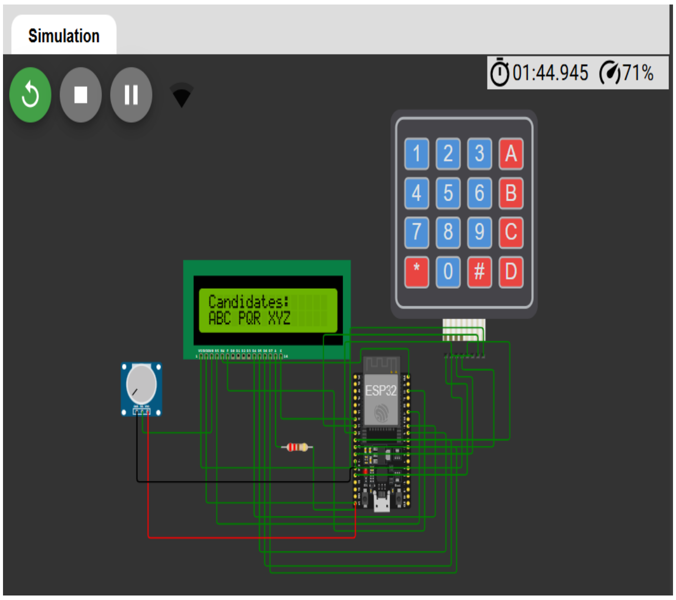
After Aadhaar entry, the voter enters the password. The password is entered using the keypad. The system checks the password along with Aadhaar details. If the password is correct, the voter is allowed to continue. These details are validated from the information which is stored in cloud (google spreadsheet) Fig. 5 shows the password entry stage.



Enter Password:
***

Fig. 5 Password Entry and Verification Stage

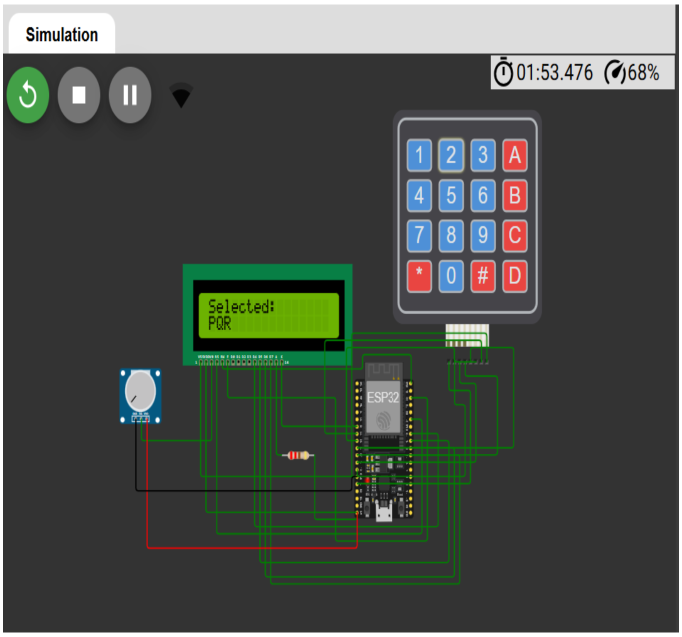
This Aadhaar and password based login prevents the voter impersonation and voter duplication as the system come again into the user login interface if the details entered is not correct or the person is trying to cast the vote again. After verification, the system displays the candidate names. The voter selects one candidate using the keypad. This is the voting stage where the vote is chosen. Fig. 6 shows the candidate selection screen.



Candidates:
ABC PQR XYZ

Fig. 6 Candidate Selection Display

The system shows that the vote is selected. This confirms that the input is received correctly before moving to the next step. After the vote confirmation a OTP is send to the mobile number, the voter is supposed to enter the OTP into the system only after which their vote will be saved. Fig. 7 shows the confirmation message after selecting the candidate.



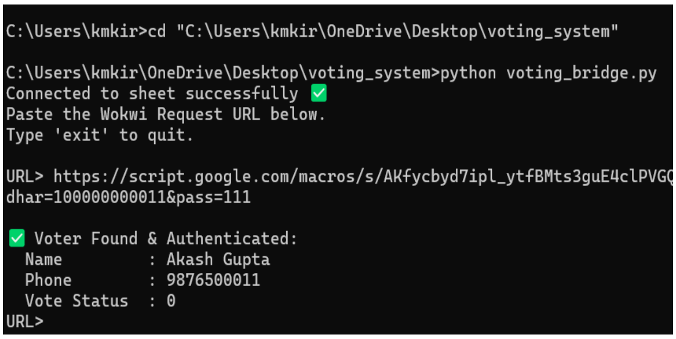
Selected:
PQR

Fig. 7 Vote Confirmation Display

After the OTP the voter will again get an SMS as an confirmation message, it states the vote is successfully saved.

B. *Wokwi–Spreadsheet Integration*

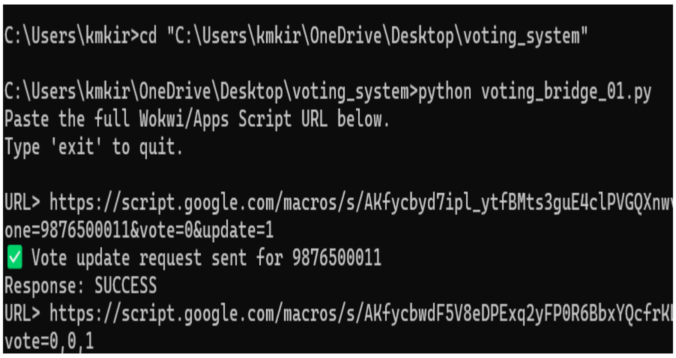
The URL generated by the ESP32 is pasted into the program. The program sends the request to the cloud spreadsheet. The voter details such as name, phone number, and vote status are returned. The message shows that the voter is found and authenticated. Fig. 8. Cloud Communication and Data Exchange Using Python Bridge.



```
C:\Users\kmkir>cd "C:\Users\kmkir\OneDrive\Desktop\voting_system"

C:\Users\kmkir\OneDrive\Desktop\voting_system>python voting_bridge.py
Connected to sheet successfully ✅
Paste the Wokwi Request URL below.
Type 'exit' to quit.

URL> https://script.google.com/macros/s/AKfycbyd7ipl_ytfBMts3guE4clPVGQ
dhar=100000000011&pass=111

✅ Voter Found & Authenticated:
  Name         : Akash Gupta
  Phone        : 9876500011
  Vote Status  : 0
URL>
```

Fig. 8. Cloud Communication and Data Exchange Using Python Bridge

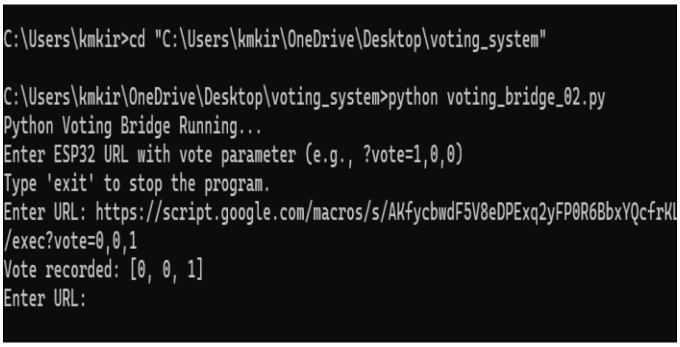
The script URL generated by the ESP32 is pasted into the terminal. The Python program sends a request to the cloud spreadsheet. The response message shows success. This confirms that the voter status is updated correctly in the cloud. Fig. 9 shows the execution of the Python bridge program used to update voter status.



```
C:\Users\kmkir>cd "C:\Users\kmkir\OneDrive\Desktop\voting_system"

C:\Users\kmkir\OneDrive\Desktop\voting_system>python voting_bridge_01.py
Paste the full Wokwi/Apps Script URL below.
Type 'exit' to quit.

URL> https://script.google.com/macros/s/AKfycbyd7ipl_ytfBMts3guE4clPVGQXnw
one=9876500011&vote=0&update=1
✅ Vote update request sent for 9876500011
Response: SUCCESS
URL> https://script.google.com/macros/s/AKfycbwdF5V8eDPExq2yFP0R6BbxYQcfrKl
vote=0,0,1
```

Fig.9.Python Bridge Execution for Voter Status Update

The ESP32 URL is given as input to the program. The vote data is received and stored in the cloud. The terminal output shows the recorded vote values. This confirms that the vote is successfully stored. Fig. 10 shows the execution of the Python bridge program for vote recording.

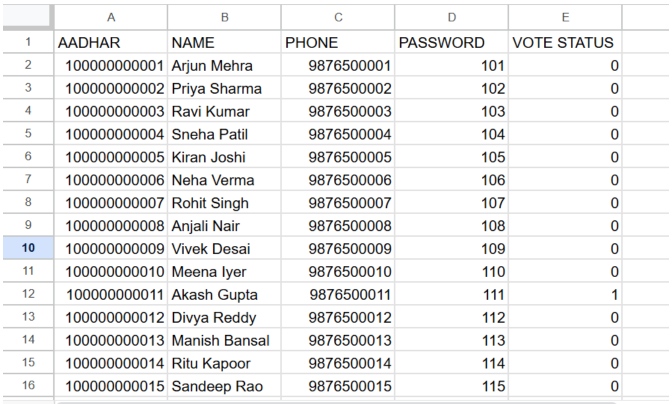


```
C:\Users\kmkir>cd "C:\Users\kmkir\OneDrive\Desktop\voting_system"

C:\Users\kmkir\OneDrive\Desktop\voting_system>python voting_bridge_02.py
Python Voting Bridge Running...
Enter ESP32 URL with vote parameter (e.g., ?vote=1,0,0)
Type 'exit' to stop the program.
Enter URL: https://script.google.com/macros/s/AKfycbwdF5V8eDPExq2yFP0R6BbxYQcfrKl
/exec?vote=0,0,1
Vote recorded: [0, 0, 1]
Enter URL:
```

Fig.10.Python Bridge Execution for Vote Recording

C. *Spreadsheet-Based Voter and Result Management*

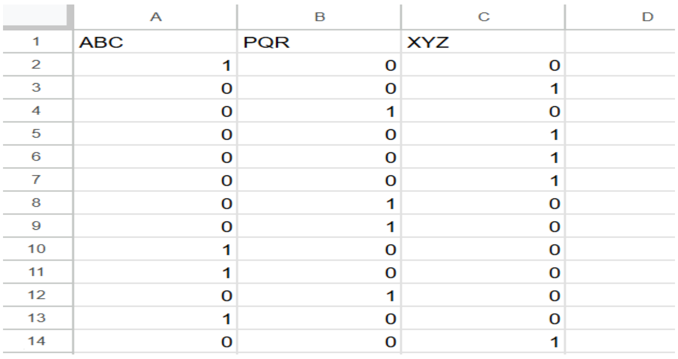
The table has Aadhaar number, name, phone number, password, and vote status. Each voter is stored in one row. The vote status value is zero before voting. After voting, this value is changed. This value is checked to avoid voting again. Fig. 11 shows the voter list stored in the cloud spreadsheet.



	A	B	C	D	E
1	AADHAR	NAME	PHONE	PASSWORD	VOTE STATUS
2	100000000001	Arjun Mehra	9876500001	101	0
3	100000000002	Priya Sharma	9876500002	102	0
4	100000000003	Ravi Kumar	9876500003	103	0
5	100000000004	Sneha Patil	9876500004	104	0
6	100000000005	Kiran Joshi	9876500005	105	0
7	100000000006	Neha Verma	9876500006	106	0
8	100000000007	Rohit Singh	9876500007	107	0
9	100000000008	Anjali Nair	9876500008	108	0
10	100000000009	Vivek Desai	9876500009	109	0
11	100000000010	Meena Iyer	9876500010	110	0
12	100000000011	Akash Gupta	9876500011	111	1
13	100000000012	Divya Reddy	9876500012	112	0
14	100000000013	Manish Bansal	9876500013	113	0
15	100000000014	Ritu Kapoor	9876500014	114	0
16	100000000015	Sandeep Rao	9876500015	115	0

Fig. 11. Cloud-Based Voter Database with Vote Status Update

The columns represent candidates ABC, PQR, and XYZ. Each row represents 1 vote. A value of '1' is entered for the selected candidate. Other values remain '0' (This is to count votes). Fig. 12 shows the vote result sheet stored in the cloud.

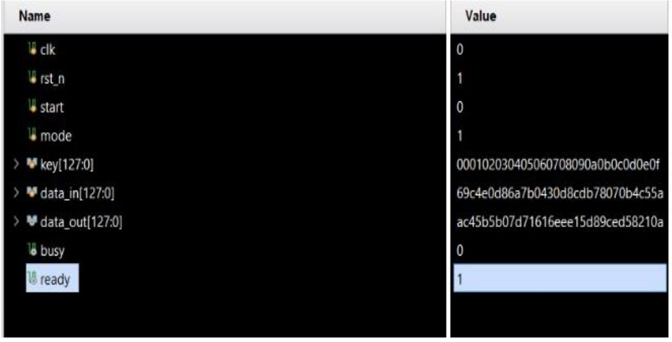


	A	B	C	D
1	ABC	PQR	XYZ	
2	1	0	0	
3	0	0	1	
4	0	1	0	
5	0	0	1	
6	0	0	1	
7	0	0	1	
8	0	1	0	
9	0	1	0	
10	1	0	0	
11	1	0	0	
12	0	1	0	
13	1	0	0	
14	0	0	1	

Fig. 12. Cloud Storage of Candidate-Wise Voting Results

D. *Secure Encryption and Decryption in Xilinx Vivado*

Signals such as clock, reset, start, key, input data, and output data are displayed. The output data value changes after encryption. The ready signal becomes high after encryption is completed.



Name	Value
clk	0
rst_n	1
start	0
mode	1
key[127:0]	000102030405060708090a0b0c0d0e0f
data_in[127:0]	69c4e0d86a7b0430d8cdb78070b4c55a
data_out[127:0]	ac45b5b07d71616eee15d89ced58210a
busy	0
ready	1

Fig. 13. AES Encryption Simulation values on FPGA

This shows that the AES module finishes the operation correctly. Fig. 13 shows the simulation values of the AES encryption module.

The signals of time along the time axis are the clock and data signals where the input data is fed first and aligned with the clock. The encryption stage entails encryption of an input data with an AES module in the use of a secret key to achieve an output in the form of transformed ciphertext. AES encryption success is assured by a variation in output values. Equally, ciphertext can be decrypted by the same key to obtain the original text and ensure successful and safe functioning. Fig. 14 shows the waveform of the timing of the entire process of encryption and decryption of AES.

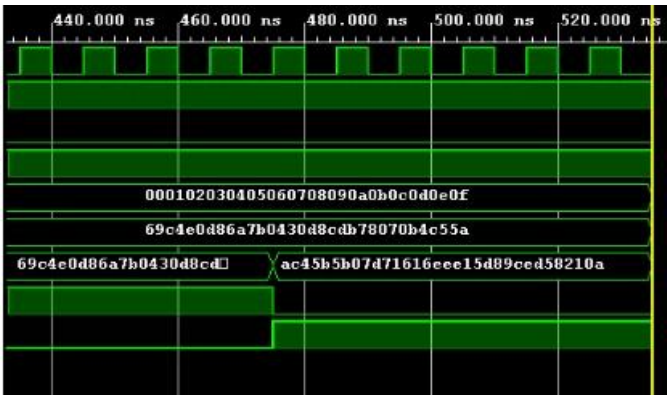

440.000 ns
460.000 ns
480.000 ns
500.000 ns
520.000 ns
000102030405060708090a0b0c0d0e0f
69c4e0d86a7b0430d8cdb78070b4c55a
69c4e0d86a7b0430d8cd□
ac45b5b07d71616eee15d89ced58210a

Fig. 14.Simulation Waveform Implemented on FPGA

V. Conclusion

The secure electronic voting system which was presented in this paper aimed at dealing with the critical challenges of voter impersonation, data manipulation and unavailability of vote secrecy. The proposed system will enable the preservation of the integrity and privacy of voting information by promoting the authentication procedure using Aadhaar, the validation of the password, and an encryption tool (AES), which should make sure that a person can cast a vote only in case he/she is an authorized voter. The architecture offers a robust and non-tamperable infrastructure that can be used in the digital elections. The findings reveal that the system can be used to provide increased security and confidence of the voters in the elections without lacking efficiency and scalability.

The future development will concentrate on improving the system by the use of sophisticated authentication and access control protocols. Fingerprint, facial, or iris recognition can be used as a part of the biometric authentication, which will further improve the identification process and enable the voter process the login process with ease. An extra impersonation security layer of using an OTP verification step can be added before voting. Moreover, authorized decryption mechanisms of biometric identification of election requires can provide a limited and safe access of voting outcomes. These upgrades will have the purpose of transforming the proposed system into a totally transparent, scalable, and safe system that can be utilized in large scale state and national elections.

References

[1] M. J, A. Periasamy, D. Kannu, S. T, and T. B, “Secure Electronic Voting System with Fingerprint Authentication and SMS Confirmation," International Journal of Progressive Research in Engineering Management and Science (IJPREMS), vol. 05, no. 04, pp. 2999-3004, Apr. 2025.

[2] V. Natarajan, S. M. S, S. Vaz J, and R. Pathi R, "Online Voting System Using AES Algorithm with OTP Validation," International Research Journal on Advanced Engineering Hub (IRJAEH), vol. 02, no. 02, pp. 57-61, Feb. 2024.

[3] T. Haripriya, Vinodkumar BG, M. Babu, G. Aswini, and R. M S, "Biometric System Based Electronic Voting Machine," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 10, no. 3, pp. 155-160, May-Jun. 2024.

[4] M. Jain, S. Raut, and S. Ghadge, "E-Voting System Using Machine Learning, Blockchain, and Cryptography," International Journal on Science and Technology (IJSAT), vol. 16, no. 2, Apr.-Jun. 2025.

[5] P. G R, S. S, C. N, S. J, and Megha, "Aadhar Based Voting System With Finger Print Authentication," IJSART, vol. 10, no. 12, pp. 26-29, Dec. 2024.

[6] S. Durga, E. Daniel, S. Seetha, and S. Deepakanmani, “Private and Secure Blockchain-Based Mechanism for an Online Voting System,” in *Big Data Innovation for Sustainable Cognitive Computing (BDCC 2021)*, Springer, Cham, Switzerland, 2022, pp. 453–464.

[7] C. Chidanandamrita, B. Ramesh, N. K. Devika, and K. A. Anantha Krishnan, “VLSI Implementation of Crypto Coprocessor Using AES and LFSR,” *Microprocessors and Microsystems*, Elsevier, vol. 88, pp. 1–10, 2022.

[8] C. Chidanandamrita, B. Ramesh, P. Mammen, A. K. N., and S. Sreehari, “Implementation of an Efficient Hybrid Encryption Technique,” *Procedia Computer Science*, Elsevier, vol. 200, pp. 123–130, 2022.

[9] C. Chidanandamrita and G. R. S. Geethu, “Design and Implementation of Reconfigurable Linear Feedback Shift Register,” *Microelectronics Journal*, Elsevier, vol. 127, pp. 1–8, 2022.

[10] TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham, “Research Contributions in Secure and Authenticated Encryption Systems,” Amritapuri Campus, India, Technical Report, 2021.

[11] Amrita Vishwa Vidyapeetham, “Research Trends in Cryptography, Blockchain and Secure Voting Systems,” School of Engineering, Amritapuri Campus, India, Technical Report, 2022.