Electronic Voting Machine: Security Risks and Solutions

Authors: Dheeraj Deep Baluvuri
Co-authors: Prajwal Deep Baluvuri, Jayadeep Baluvuri
Reviewers: Won Park, Darrell. R

Abstract

This research is to study security risks and vulnerabilities of Electronic Voting Machine (EVM) by simulating an optical scan EVM. The research also provides potential solutions for more secure, transparent voting procedure. EVM became a pandora box in the countries where the public voting conducted using EVMs. As it's machine-based voting, voters don't have transparency on how the EVM system works. People and media are very much curious on functionality and speculations around potential risks of EVMs. As concerns and protests increase about EVMs being hacked using artificial intelligence or due to tampering of the EVM machines, this research paper attempts to provide analysis on possibilities of EVM tampering, security risks and potential solutions. The goal is to analyze security flaws, test patches, and suggest changes that could be applied on real-world EVMs. Through research, design, and experimentation on a software-based system, the research discusses how minor improvements like encryption, paper records, and audit features can significantly strengthen transparency and confidence in electronic voting machines.

Introduction

EVMs are increasingly being used to hold elections around the world due to their speed and efficiency but fears of them being hacked and tampered with have raised alarm. There have been claims that outsiders or AI can breach EVMs by accessing the mainboard and altering the database or voting records [4]. There are arguments that AI would help in carrying out cyberattacks like brute-force or software vulnerability detection more swiftly than human capabilities, while others warn against AI applications in phishing or social engineering to grant access to election infrastructure [4]. Though there is no tangible evidence to confirm these charges, these still create controversy and suspicion [4]. Well-known organizations for EVM production are Bharat Electronics Limited (BEL) and Electronics Corporation of India Limited (ECIL) in India, who are both responsible for producing safe voting machines that are used in national elections [10][11]. By working on this project, I aim to show how the vulnerabilities may be present and can be fixed in an operational, small-scale model [6].

The study is important because many countries have had issues with suspicious elections, and making voting machines more secure might lead to fairer, more credible results [5]. Certain countries like India and Brazil still rely heavily on EVMs, whereas others have dropped using them due to these problems [2][3]. In India's 2019 general election, for example, some political parties alleged EVM tampering, even though the Election Commission denied the claims and no substantial evidence was found [2]. In Venezuela, the Smartmatic company, which helped in managing the country's 2017 constituent assembly election, accused the counting of votes of

being hacked even when the machines were used as planned, raising alarm internationally [9]. Germany and the Netherlands, on their end, have abandoned the use of electronic voting due to security and transparency concerns [5]. In this project, I aim to simulate a secure EVM and discover the impact of minor changes.

Results

To solve these issues, I researched how real-world EVMs are secured. Encryption, version control, audit trails, and tamper-evident hardware are critical [1][5][6]. In my system, I added encryption to safeguard vote data, backups to safely store results, and paper printouts for finalizing choices. I also added scanning checks to reduce the amount of errors and utilized logs to monitor every vote for transparency. While my system is not perfect, these extensions illustrate how uncluttered design decisions can greatly improve the security of voting [6][7].

I tested the simulation by creating an eligible voters database and limiting each to cast one vote. The system encrypted personal information, stored all of the votes in a publicly verifiable log, and displayed live results. Upon casting, a simulated receipt confirmed the selection in a VVPAT-like fashion. In comparison with other research work, my project aligns with core recommendations by institutions like Princeton University [6] and DEFCON's Voting Village [7], which have been robust advocates of voter-verifiable paper trails, vote storage in secure ways, and auditability. While their research seeks to address more advanced attacks like firmware tampering or electromagnetic radiation leakage, my simulation is focused on demonstrating basic vulnerabilities and real-world protection at a reduced scale [6][7]. This research proves that even with limited resources, dramatic changes can be made to the design of EVM in the lines of transparency and integrity [1][2][3][5][6].

Background on EVMS

Electronic Voting Machines are used in casting and counting votes electronically in elections. They are usually composed of a Control Unit (for counting and storage of the vote) and a Ballot Unit (the place where the voter makes his or her choice) [2]. Some advanced EVMs also incorporate a Voter Verifiable Paper Audit Trail (VVPAT), which displays a paper receipt of the choice made by the voter to provide a tangible audit trail [2]. There are numerous various systems utilized

In the United States, including DREs with or without VVPATs, optical scanners of paper ballots, and ballot-marking devices assisting voters to markup ballots that are subsequently read [3]. In India, the system is much more standardized practically solely reliant on stand-alone, customized EVMs developed by BEL [10] and ECIL [11], topped up with VVPATs for verification. These are not networked and are implemented with tamper resistance in mind and simplicity. Brazil utilizes completely digital DRE-type voting machines that contain touchscreen interfaces along with biometric verification but has come under criticism for failing to leave a paper audit trail [3].
There is no international ranking list of EVMs since each one is tailored to address the

requirements of each nation, but systems providing electronic convenience with paper verifiability those with VVPATs are widely considered to be more secure and dependable [2][3][5]. Being aware of the differences is important for my research since I would like to design a secure, auditable, and convenient EVM model by learning best practices from nations [6].

In this project, I have emulated an Optical Scan Electronic Voting Machine [3]. Optical Scan Electronic Voting Machines allow voters to mark choices on paper ballots, which are then read electronically and tallied [3]. This method takes advantage of the transparency of paper voting and the efficiency of electronic counting [3]. Even though optical scan systems are widespread and more secure, thanks to the paper record, they are not resistant to issues like misread ballots, scanner sensitivity problems, and ballot design flaws [3]. Still, their speed-security balance makes them the most in-demand system in most places throughout the world, especially where auditability is mandatory by law [2][3].

Materials and Methods

Component	Description	Estimated Price (USD)
Monitor or PC	Display interface for Ballot simulation	$100.00 - $150.00
USD Keyboard	For input and testing	$15.00
Thermal Printer (58mm or 80 mm	Simulates VVPAT paper trail	$40.00 - $70.00
USB Wi-Fi/Bluetooth Module	For network simulation or optional features	$10.00 - $20.00

These materials allow me to simulate the functions of an optical scan voting machine. The computer runs the vote-processing software, while the monitor displays the interface. The keyboard simulates voter input, and the printer can be used to produce a paper trail or audit copy. Results are also stored in files for future retrieval and verification

Methodology

I designed the system to focus on simplicity and accessibility. The voter interface is easy and intuitive voters simply select a candidate, view their ballot, and cast their vote. I employed easy encryption to protect vote information during and after voting [6]. Files are saved locally, and backups are created to prevent data from being lost. I compared different types of ballots and scan options to reduce reading mistakes and make the simulation comparable to the optical scan technique [3]. I also added simple audit features like electronic log and paper printout to emulate VVPAT capability [2].

Vulnerabilities and Impact

Despite their advantages, EVM simulation being no exception is vulnerable to a variety of

risks. In the absence of strong physical security, machines are vulnerable to tampering [1][5]. Software models only can be vulnerable to hacking, especially if not properly encrypted or updated [4][6]. Weak authentication can allow unauthorized individuals to access the system [4].

These flaws can lead to incorrect voting counts, diminished public trust, and even public disorder when the legitimacy of election outcomes is challenged [5]. Even though most hacking claims are unfounded, the threat alone can destabilize democracy faith [4].

Conclusion

This project helped me understand how EVMs work, why they can be cheated, and how to improve them. Simple tools like encryption and paper records can secure votes and increase trust [1][2][3][5]. In the future, better tools like blockchain, biometric authentication, and open-source voting software may make EVMs even more secure and transparent [6][7].

Appendix

```
def create_tables(self):
    def __init__(self, db_path='../voter_info.db'):
      self.db_path = db_path
      cursor = sqlite3.connect(self.db_path).cursor()

      # Create table for voter information (with has_voted field)
      cursor.execute('''
          CREATE TABLE IF NOT EXISTS voters (
              id INTEGER PRIMARY KEY AUTOINCREMENT,
              name TEXT NOT NULL,
              age INTEGER NOT NULL,
              state TEXT NOT NULL,
              registration_date TEXT NOT NULL,
              has_voted BOOLEAN DEFAULT 0
          )
      ''')

      # Create table for voting records
      cursor.execute('''
          CREATE TABLE IF NOT EXISTS voting_records (
              id INTEGER PRIMARY KEY AUTOINCREMENT,
              voter_id INTEGER,
              election_date TEXT NOT NULL,
              candidate TEXT NOT NULL,
              FOREIGN KEY (voter_id) REFERENCES voters(id)
          )
      ''')

      sqlite3.connect(self.db_path).commit()
      sqlite3.connect(self.db_path).close()
```

Figure 1: Database schema for storing registered voters. Each voter has a has_voted flag to prevent double-voting, ensuring election integrity.

```python
def cast_vote(voter_id, candidate):
    conn = connect_database(DB_PATH)
    cursor = conn.cursor()

    # Insert the vote
    cursor.execute('''
        INSERT INTO voting_records (voter_id, election_date, candidate)
        VALUES (?, ?, ?)
    ''', (voter_id, datetime.today().date(), candidate))

    # Update voter's has_voted flag
    cursor.execute('''
        UPDATE voters SET has_voted = 1 WHERE id = ?
    ''', (voter_id,))

    conn.commit()
    conn.close()
```

Figure 2: Code that records a vote and updates the database so the same voter cannot vote twice. This enforces the principle of “one person, one vote.”

```python
# Generate a new key (only once — store securely in production!)
def generate_key():
    return Fernet.generate_key()

# Encrypt text data using Fernet
def encrypt_data(data, key):
    fernet = Fernet(key)
    encrypted = fernet.encrypt(data.encode())
    return encrypted

# Decrypt encrypted data
def decrypt_data(encrypted_data, key):
    fernet = Fernet(key)
    decrypted = fernet.decrypt(encrypted_data).decode()
    return decrypted

# Testing the functions with mock voter data
    if __name__ == '__main__':
      key = generate_key()
      print(f"Generated Key (store this!): {key.decode()}")

      mock_voter_name = "Alice Johnson"
      encrypted_name = encrypt_data(mock_voter_name, key)
      print(f"Encrypted: {encrypted_name}")

      decrypted_name = decrypt_data(encrypted_name, key)
      print(f"Decrypted: {decrypted_name}")
```

Figure 3: Encryption and decryption using Fernet. Votes and voter identities can be encrypted before storage, ensuring that sensitive data is secure.

```
for row in rows:
    if anonymize:
        print({
            'id': row[0],
            'age': row[2],
            'state': row[3],
            'registration_date': row[4],
            'has_voted': bool(row[5])
        })
    else:
        print({
            'id': row[0],
            'name': row[1],
            'age': row[2],
            'state': row[3],
            'registration_date': row[4],
            'has_voted': bool(row[5])
        })

conn.close()
```

Figure 4: Example of anonymized voter data output. Personal identifiers are hidden, but age, state, and status can still be analyzed for election audits

```
Label(self.master, text="Choose your name:").pack()
OptionMenu(self.master, self.selected_voter, *options).pack()

Label(self.master, text="Vote for a candidate:").pack(pady=10)
Button(self.master, text="Alice", command=lambda: self.submit_vote("Alice")).pack(pady=5)
Button(self.master, text="Bob", command=lambda: self.submit_vote("Bob")).pack(pady=5)

self.result_label = Label(self.master, text="")
self.result_label.pack(pady=10)
```

Figure 5: Graphical user interface (GUI) of the EVM simulation. Voters select their name from a dropdown, then cast their vote by pressing a candidate button.

```
def get_results():
    conn = connect_database(DB_PATH)
    cursor = conn.cursor()
    cursor.execute('''
        SELECT candidate, COUNT(*) FROM voting_records GROUP BY candidate
    ''')
    results = cursor.fetchall()
    conn.close()
    return results
```

Figure 6: Query to tally election results in real time. This demonstrates how electronic voting can provide immediate transparency after polls close.

Acknowledgements

I would like to thank my mentors and Polygence for providing guidance and access to resources that made this project possible. All simulations, analysis, and writing were completed independently.

References

1. *Voluntary Voting System guidelines*. (2025, January 31). U.S. Election Assistance Commission. Retrieved November 26, 2025, from https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines

2. Desk, H. N. (2024, March 30). *All you need to know about VVPAT*. The Hindu. https://www.thehindu.com/news/national/all-you-need-to-know-about-vvpat/article617950 74.ece

3. *Verifier*. (n.d.). Verified Voting. https://verifiedvoting.org/verifier/#mode/navigate/map/voteEquip/mapType/ppEquip/year/ 202

4. Livemint. (2024, June 16). Elon Musk's "anything can be hacked" VS 'Rajeev Chandrasekhar's "technically you are right" on use of EVMs. *Mint*. https://www.livemint.com/news/india/elon-musks-anything-can-be-hacked-vs-rajeev-chan drasekhars-technically-you-are-right-on-use-of-evms-11718523721133.html

5. *Securing the vote*. (n.d.). National Academies. https://www.nap.edu/catalog/25120/securing-the-vote-protecting-american-democrac

6. Felten, E. (2010, April 29). *India's Electronic Voting Machines Have Security Problems*. CITP Blog. https://blog.citp.princeton.edu/2010/04/29/indias-electronic-voting-machines-have-securit y-problems/

7. Blaze, M., Braun, J., Hursti, H., Hall, J. L., MacAlpine, M., & DEFCON. (2017). *DEFCON 25 Voting Machine Hacking Village report on cyber vulnerabilities in U.S. election equipment, databases, and infrastructure*. https://www.defcon.org/images/defcon-25/DEF%20CON%2025%20voting%20village%20 report.pdf

8. Smartmatic statement on the recent constituent Assembly election in Venezuela. (2017, August 2). *Smartmatic*. https://www.smartmatic.com/media/smartmatic-statement-on-the-recent-constituent-asse mbly-election-in-venezuela/

9. Prasad, H. K., Halderman, J. A., Rop Gonggrijp, Wolchok, S., Wustrow, E., Kankipati, A., Sakhamuri, S. K., Yagati, V., NetIndia, (P) Ltd., & The University of Michigan. (2010). Security analysis of India's electronic voting machines. *Indiaevm*. https://indiaevm.org/evm_tr2010.pdf

10. BEL. (2024, February 29). *Electronic Voting Machine -*

BEL. https://bel-india.in/product/electronic-voting-machine/