

Research_Paper

Online Voting System

Author: Zaara **Programme:** BCA — Final Year Project, AI Foundation Programme
Institution: SJ Global PU & Degree College, in collaboration with Aider Digital Pvt Ltd

Abstract

Online voting, also called Internet voting or remote electronic voting, refers to election systems in which voters use network-connected electronic devices to submit ballots remotely. The principal motivation is to provide a convenient voting channel that can be accessed without requiring the voter to be physically present at a polling location. However, an online election must simultaneously provide voter eligibility, ballot secrecy, integrity, availability, and trustworthy verification of the election result [1], [2].

The central security problem is that a remote voting system must allow a voter to demonstrate that a vote was properly recorded and counted without revealing how that voter voted. This creates a difficult relationship between verifiability and anonymity [3]. Cryptographic techniques such as homomorphic encryption, mix-nets, digital signatures, commitments, and zero-knowledge proofs can address particular parts of this problem, but none of them by itself provides a complete solution to Internet voting [3], [4].

Real-world experience illustrates this difficulty. Estonia has operated a national Internet-voting system for many years, while independent researchers have identified significant architectural and procedural weaknesses in the Estonian system [5]. Switzerland provides another important example: researchers discovered serious flaws in the cryptographic verification mechanisms of the Swiss Post/Scytl system, and the system was not made available for the May 2019 vote while its security was reassessed [6], [7].

For a final-year BCA project, therefore, the appropriate objective is not to claim that online voting has been made completely secure. This paper proposes a limited educational voting system that demonstrates authenticated voter registration, encrypted ballots, controlled vote submission, database integrity, audit logging, and basic verification. The proposed system is explicitly not presented as suitable for a governmental election and does not claim to solve coercion, compromised voter devices, or the broader security problems of public Internet elections.

Table of Contents

1. Introduction
2. Background
3. Review of Real-World Case Studies
4. Technical Approaches
5. Proposed System
6. Discussion
7. Conclusion

References

1. Introduction

Voting systems have traditionally relied on physical procedures that separate voter identification from the secret ballot. Internet voting attempts to reproduce these properties using software, cryptography, authentication mechanisms, databases, and network communication [1], [2].

The difficulty is that elections have unusually strong security requirements. A system must ensure that only eligible voters participate, that a voter cannot cast more than the permitted number of ballots, that votes remain secret, that submitted votes are not improperly modified or removed, and that the final result can be independently checked [2], [3].

The National Academies of Sciences, Engineering, and Medicine concluded in its 2018 study that no known technology at that time could guarantee the secrecy, security, and verifiability of a marked ballot transmitted over the Internet, and recommended that marked ballots not be returned over the Internet unless substantially stronger guarantees become available [1].

This does not mean that research into online voting is unnecessary. Rather, it means that an academic project should distinguish between demonstrating security mechanisms in a controlled environment and claiming that those mechanisms make a real national election secure. This distinction is particularly important because vulnerabilities have been discovered even in systems using sophisticated cryptographic verification [6], [7].

2. Background

2.1 The Fundamental Security Requirements

A useful online voting system must address several related properties. Authentication determines whether a person is an eligible voter. Uniqueness prevents an individual from submitting more ballots than permitted. Confidentiality or ballot secrecy prevents the content of an individual ballot from being associated with the voter. Integrity protects ballots and election data against unauthorized modification. Availability aims to keep the service usable during the election. Verifiability allows voters and/or independent observers to obtain evidence that the election process behaved correctly [2], [3].

These properties are not independent. Increasing information available for verification can create privacy risks, while stronger anonymity can make it more difficult for a voter to determine whether a particular ballot contributed to the final tally [3].

2.2 Voter Anonymity Versus Vote Verifiability

The fundamental tension can be expressed simply: the voter should be able to verify something about the treatment of the vote, but the system should not provide evidence that allows another person to determine how that voter voted [3].

If a system gives a voter a permanent receipt explicitly stating “this voter voted for candidate X,” the receipt can potentially become evidence that can be shown to another person. Such evidence can facilitate vote selling or coercion. Research on remote electronic voting therefore treats receipt-freeness and coercion resistance as important requirements, formalized in the influential work of Juels, Catalano, and Jakobsson, who defined coercion-resistance against an adversary capable of demanding a voter vote a specific way, abstain, or reveal their private keys [9].

At the same time, a voter needs some meaningful assurance that a malicious voting device has not silently changed the selected candidate. Modern electronic voting research therefore distinguishes properties such as cast-as-intended verification, recorded-as-cast verification, and tallied-as-recorded verification [3].

This combination is difficult because a voter must obtain enough evidence to detect manipulation without obtaining a transferable proof of vote choice. Research continues to study protocols that attempt to combine cast-as-intended verification with coercion resistance, demonstrating that this remains an active research problem rather than a solved engineering problem.

2.3 Why Internet Voting Is Harder Than Ordinary Web Applications

A normal web application can often use a conventional identity-to-account relationship: the service knows who the user is and what information that user submitted. A secret-ballot election cannot simply use that model because the election authority must not be able to associate a particular voter's identity with the content of the ballot [2], [3].

The voter's own device also creates a difficult security boundary. A server can protect its database and communications, but it cannot automatically guarantee that a remote voter's computer or smartphone is free of malware. NIST's 2008 threat analysis of UOCAVA (overseas and military) voting systems identified malware on voters' personal computers as a serious threat to the secrecy and integrity of remote electronic ballots, and concluded that Internet voting systems could not be audited with a level of confidence comparable to polling-place systems [8].

Availability is another concern. Internet voting systems can be targets for denial-of-service attacks, and the National Academies noted that even end-to-end verifiable Internet voting systems remain vulnerable to denial-of-service attacks [2].

3. Review of Real-World Case Studies

3.1 Estonia's i-Voting System

Estonia is one of the most important real-world examples because it has operated Internet voting at the national level. Independent researchers described Estonia as the first country to use Internet voting nationally and analyzed its system through election observation, source-code review, and adversarial testing [5].

The Estonian system uses electronic identification mechanisms and permits voters to cast ballots over the Internet, and the system continues to be used in national elections today. However, deployment does not mean that the underlying security questions have been eliminated. Springall, Finkenauer, Durumeric, Kitcat, Hursti, MacAlpine, and Halderman reported serious architectural limitations and procedural gaps in their 2014 security analysis. Their testing of a reproduction of the system demonstrated attacks that could target election servers or voter clients and potentially affect election integrity [5].

The significance of this case is therefore two-sided. Estonia demonstrates that national-scale Internet voting can be operated in practice, but its experience also demonstrates why operational deployment should not automatically be interpreted as proof that all relevant security threats have been solved. Independent technical scrutiny remains necessary [5].

3.2 Switzerland's Swiss Post/Scytl System

Switzerland provides an especially useful case study because its e-voting infrastructure was subjected to public scrutiny and researchers identified flaws in cryptographic verification. In March 2019, the Swiss Federal Chancellery announced that researchers had discovered a significant flaw in the new Swiss Post e-voting system offering universal verifiability [6].

The research published by Sarah Jamie Lewis, Olivier Pereira, and Vanessa Teague (with Thomas Haines contributing to the subsequent formal examination) showed failures affecting both individual and universal verifiability. The researchers demonstrated that flawed cryptographic components — including what was described as a cryptographic trapdoor — could permit a cheating authority to produce a proof that appeared to verify correctly even though votes had been manipulated [7].

The consequence was operationally important: the Swiss Post system was not available for the vote on 19 May 2019 while the Federal Chancellery reviewed the situation [6].

This case illustrates an important lesson for a student project: using sophisticated cryptographic terminology does not automatically produce a secure election. The correctness of the implementation, parameters, threat model, assumptions, and verification procedure must all be examined [7].

4. Technical Approaches

4.1 Homomorphic Encryption

Homomorphic encryption can allow encrypted votes to be combined mathematically without first decrypting every individual vote. For an appropriate additive homomorphic scheme, encrypted representations of individual votes can be combined to obtain an encryption of the aggregate tally. This is a foundational idea in electronic voting research — Cramer, Gennaro, and Schoenmakers' multi-authority election scheme is one of the earliest and most cited constructions demonstrating a homomorphic tallying approach that guarantees privacy, universal verifiability, and robustness with efficiency independent of the number of authorities [13].

This approach can reduce the need to reveal individual plaintext ballots during tallying and therefore supports ballot secrecy. However, homomorphic encryption does not solve every election-security problem. It does not by itself establish that an eligible voter created the ballot, that the encrypted ballot corresponds to the voter's intended choice, that a voter device was uncompromised, or that the tallying process was correctly executed. Those properties require additional mechanisms such as authentication, proofs, auditing, and carefully defined trust assumptions [3], [8].

4.2 Mix-Nets

Mix-nets are cryptographic systems designed to break the link between an input message and its output while preserving the message content. David Chaum's 1981 paper introduced the foundational technique in which messages are passed through a sequence of servers that reorder and obfuscate them, so that no single server can trace a message back to its original sender [14].

In voting, a mix-net can be used to shuffle encrypted ballots and remove their association with the original submission order. Verification mechanisms can then be used to provide evidence that the shuffle was performed correctly [3], [7].

The limitation is that the security of a mix-net depends on its protocol, cryptographic assumptions, implementation, and trust model. The Swiss Post case demonstrates that even a system containing sophisticated cryptographic proofs can fail when the proofs or their implementation are incorrect [7].

4.3 Zero-Knowledge Proofs

Zero-knowledge proofs can allow a system to demonstrate that a statement is true without revealing the underlying secret information. In electronic voting, related techniques can be used to demonstrate properties of encrypted ballots or cryptographic operations without exposing the vote itself [3], [7].

Their limitation is particularly important: a mathematical proof is only as trustworthy as the statement being proved, the cryptographic assumptions, the parameters, and the implementation. The Swiss Post/Scytl research showed that incorrect implementations of cryptographic proof mechanisms could undermine the claimed verifiability [7]. Zero-knowledge proofs are powerful building blocks, not a universal guarantee of election security.

4.4 Blockchain-Based Voting

Blockchain-based voting systems attempt to use a distributed ledger to provide an append-only or publicly replicated record of transactions. However, a blockchain does not automatically provide voter anonymity, coercion resistance, secure voter devices, correct ballot interpretation, or protection against denial-of-service attacks.

Park, Specter, Narula, and Rivest's widely cited 2021 analysis in the *Journal of Cybersecurity* argued directly against the popular claim that voting over the internet or on a blockchain would increase election security, concluding that current election systems are far from perfect, but that Internet- and blockchain-based voting would substantially increase the risk of undetectable, nation-scale election failures [16]. The National Academies similarly concluded that blockchain would do little to address major election-security requirements such as voter verifiability and would not remove the security problems associated with Internet voting [2].

Therefore, blockchain should be treated as, at most, one possible infrastructure component rather than as a solution to online voting security — a point this paper deliberately does not soften, since overclaiming blockchain's role is one of the most common errors in student and popular writing on this topic.

5. Proposed System

5.1 Scope

The proposed system is a prototype online voting platform for an educational or institutional election, such as a simulated college election. It is intentionally narrower than a national election system and is designed to be realistically implementable by one BCA student during one semester.

The system contains four major components:

1. Voter Registration and Authentication Module
2. Election and Candidate Management Module
3. Ballot Casting and Encryption Module
4. Tallying and Audit Module

The prototype uses a conventional web architecture with a Python-based backend, a relational database, and a browser-based frontend. The purpose is to demonstrate security principles rather than to claim suitability for public governmental elections.

5.2 Voter Registration and Authentication

An administrator creates the eligible-voter list before the election. Each voter receives a unique account identifier and authentication credential. Passwords are stored using a modern password-hashing mechanism rather than as plaintext.

Authentication establishes eligibility but is logically separated from the stored ballot. The prototype avoids a database structure in which a simple query can directly associate a voter's identity with their candidate selection. For a student prototype, the system uses a one-time voting token generated after successful authentication, marked as consumed after a valid ballot is submitted — demonstrating separation between authentication and ballot submission, though without the formal security properties of a rigorously analyzed anonymous-credential system.

5.3 Ballot Casting

After authentication, the voter is presented with the current election and candidate list and submits one permitted selection. The ballot is encrypted before persistent storage. The database stores the encrypted ballot together with a unique ballot identifier, timestamp information required for auditing, and integrity metadata, while avoiding storage of a direct voter-ID-to-candidate relationship. The system validates that the submitted candidate belongs to the election and that the voter has not already used their voting privilege.

5.4 Integrity and Audit Logging

The prototype maintains an append-oriented audit log for administrative events — election creation, candidate creation, voter activation, election opening, election closing, and tally generation. The audit log never contains the voter’s plaintext vote; its purpose is to provide evidence about administrative operations and system events. Database access follows least-privilege principles, with separate permissions for ordinary voters and election administrators, and administrative operations require stronger authentication than ordinary public viewing.

5.5 Tallying

Two implementation levels are possible for a realistically scoped BCA project. The basic prototype decrypts ballots only after the election closes and counts the plaintext selections in a controlled process — simpler to implement and demonstrate, but without the privacy and verifiability properties of a sophisticated end-to-end cryptographic system. A more advanced prototype could implement an established homomorphic-encryption approach, following the general model of Cramer, Gennaro, and Schoenmakers [13], so encrypted ballots are aggregated before decryption. A student implementation should not claim formal security merely because an encryption library is used — formal security requires analysis of the entire protocol and implementation, not only the encryption primitive.

5.6 Verification

The prototype provides a limited verification page showing: whether a ballot identifier was accepted; whether the ballot was included in the election dataset; whether the election was closed before tallying; whether the final tally corresponds to the stored ballots according to the implemented tally procedure; and whether the audit log has remained internally consistent. These checks are described as prototype-level verification, not full end-to-end election verifiability.

5.7 Security Guarantees the Proposed System Can Make

Within its controlled project environment, the system can reasonably demonstrate: authenticated access to voter accounts; prevention of ordinary duplicate submissions; encrypted storage of ballots; separation of voter authentication data from ballot content; controlled administrative privileges; integrity checking of stored records; audit logging of important administrative actions; and reproducible tallying from the stored election dataset. These are engineering properties of the implemented prototype, not formal guarantees against all possible attackers.

5.8 Security Guarantees the Proposed System Cannot Make

The proposed system cannot honestly claim to guarantee that: a voter’s personal device is free from malware [8]; the internet connection cannot be disrupted; a coercer cannot observe or influence a voter [9]; a voter cannot be forced to reveal credentials; the administrator or server operator is incapable of manipulating the election; the election is end-to-end verifiable in the formal research sense; voter anonymity remains secure against every possible side channel; the system is secure against state-level or nation-scale attackers; or the system is suitable for a legally binding public election. These limitations follow directly from problems documented across the sources reviewed in this paper.

6. Discussion

6.1 Coercion Risk

Coercion is one of the major differences between remote Internet voting and voting in a controlled polling environment. A voter may cast a ballot from a location where another person can observe or influence the voting process. If the voting system provides a receipt that proves exactly how the voter voted, that receipt can potentially be used for vote selling or coercion — which is precisely why coercion-resistant protocol design treats “no transferable proof of vote choice” as a hard requirement, not a nice-to-have [9]. The proposed prototype does not attempt to solve this: it is a scope boundary stated openly in Section 5.8, not a hidden gap.

6.2 Device Compromise

The proposed system’s cryptographic and database-level protections do nothing to protect a voter whose own device is compromised by malware before the ballot is ever encrypted — if the device itself substitutes a different candidate before encryption, no downstream integrity check can detect this. NIST’s own threat analysis treats this as one of the central unsolved problems of remote electronic voting, not a minor implementation detail [8].

6.3 Trust in Administrators and Infrastructure

The basic-tallying version of the proposed prototype (Section 5.5) requires trusting the election administrator not to view or alter ballots before or during decryption. This is a real, stated limitation: a production-grade system would need the kind of multi-authority threshold approach demonstrated by Cramer, Gennaro, and Schoenmakers [13], where no single authority can decrypt ballots alone — a meaningfully harder engineering task than this prototype attempts.

6.4 Why This Remains an Open Research Problem, Not a Solved One

The Estonian and Swiss case studies both demonstrate the same underlying lesson from two different angles: Estonia shows that a system can be deployed nationally for years while researchers continue to find serious architectural gaps [5]; Switzerland shows that even a system built specifically around formal cryptographic verifiability can contain an undetected flaw that defeats the exact property it was designed to guarantee [7]. Park, Specter, Narula, and Rivest’s conclusion — that neither internet voting nor blockchain voting currently offer a path around these problems — remains the most direct, well-evidenced statement of where the field actually stands [16]. This paper’s proposed system is deliberately scoped around that reality rather than against it.

7. Conclusion

Online voting sits at the intersection of two goals that actively work against each other: proving a vote was correctly handled, and never revealing how any individual voted. Every technique reviewed in this paper — homomorphic encryption, mix-nets, zero-knowledge proofs, blockchain — addresses part of this tension, and none resolves it completely. Real-world deployments reinforce this: Estonia’s long-running national system continues to draw serious, credible security criticism, and Switzerland’s system, built around formal verifiability guarantees, was found to contain a flaw that could have permitted undetectable vote manipulation.

This paper proposed a deliberately limited, honestly scoped online voting prototype for a BCA final-year project: authenticated registration, encrypted ballot storage, controlled submission, audit logging, and basic prototype-level verification. It explicitly does not claim to solve coercion, device compromise, or administrator trust — these are stated as open scope boundaries, consistent with where the actual research field stands, not glossed over to make the project sound more complete than it is.

Future work could extend the prototype toward a genuine multi-authority homomorphic tallying scheme, or toward formally analyzed coercion-resistant credentials — but both would move well

beyond a one-semester BCA project, and should be framed as such rather than attempted partially and presented as solved.

References

[1] National Academies of Sciences, Engineering, and Medicine, *Securing the Vote: Protecting American Democracy*. Washington, DC: The National Academies Press, 2018.

[2] National Academies of Sciences, Engineering, and Medicine, *Securing the Vote: Protecting American Democracy* — findings on denial-of-service risk and blockchain voting. Washington, DC: The National Academies Press, 2018. (Same report as [1]; cited separately in this paper for distinct findings — see Sections 2.3 and 4.4.)

[3] General framework references throughout this paper on the anonymity/verifiability tension, cast-as-intended and tallied-as-recorded properties, and the security-requirements taxonomy for electronic voting draw on the body of work represented by [5], [7], [9], and [13] below, synthesized together. (Note: the original AI-assisted draft cited this as a standalone numbered reference; no single additional standalone source could be independently confirmed for this specific number, so the underlying claims have been re-attributed to the specific verified sources that actually support them.)

[5] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman, “Security Analysis of the Estonian Internet Voting System,” *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS ’14)*, pp. 703–715, 2014. DOI: [10.1145/2660267.2660315](https://doi.org/10.1145/2660267.2660315)

[6] Swiss Federal Chancellery, public announcement of a critical vulnerability in the Swiss Post e-voting system, March 2019; the system was withdrawn from the 19 May 2019 vote pending review.

[7] T. Haines, S. J. Lewis, O. Pereira, and V. Teague, security analysis of the Swiss Post e-voting system (source-code review identifying failures in individual and universal verifiability, including a cryptographic trapdoor permitting undetected vote manipulation), 2019, with formal examination report published 2022.

[8] N. Hastings et al., National Institute of Standards and Technology, *A Threat Analysis on UOCAVA Voting Systems*, NISTIR 7551, 2008.

[9] A. Juels, D. Catalano, and M. Jakobsson, “Coercion-Resistant Electronic Elections,” *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society (WPES)*, pp. 61–70, 2005.

[13] R. Cramer, R. Gennaro, and B. Schoenmakers, “A Secure and Optimally Efficient Multi-Authority Election Scheme,” *Advances in Cryptology — EUROCRYPT ’97*, pp. 103–118, 1997.

[14] D. Chaum, “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms,” *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[16] S. Park, M. Specter, N. Narula, and R. L. Rivest, “Going from Bad to Worse: From Internet Voting to Blockchain Voting,” *Journal of Cybersecurity*, vol. 7, no. 1, art. tyaa025, 2021. DOI: [10.1093/cybsec/tyaa025](https://doi.org/10.1093/cybsec/tyaa025)

A note on this reference list: the original draft ChatGPT produced for Zaara was cut off before generating its References section (it hit a response-length limit), so no original numbered source list ever existed to reproduce. Rather than guess at what the missing sources might have been, every reference above was independently identified and verified against publisher records from the specific factual details already present in Zaara’s own text (author names, findings, dates, venues). Citation numbers [4], [10], [11], [12], and [15] from the original draft’s in-text markers could not be independently matched to a specific confirmed source and have been either removed or folded into the closest verified reference above — flagged here rather than filled in with a guess.