POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Towards Secure Electronic Voting: a Literature Review on E-Voting Systems and Attacks

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Author: **Riccardo Barelli**

Student ID: 976484
Advisor: Prof. Stefano Longari
Co-advisors: Mario D'Onghia
Academic Year: 2022-23

Abstract

The trend of electronic voting has risen in recent years as an alternative to paper ballot elections bringing meaningful benefits in terms of efficiency and error proneness. Despite these premises, systems employed in real elections resulted to be subject to software errors and offered vulnerabilities exploitable by a malicious attacker. The challenge presented by electronic voting can be brought back to multiple properties required in the system design, with some of them apparently in conflict. In this thesis, the security and properties of the novel proposals for an electronic voting system are examined, differentiating in poll-site and remote systems. A viable solution is represented by verifiable systems, where a voter receives proof of his/her vote to check after the election. However, it can result especially difficult to design accurately in a remote setting due to the issue of coercion. In recent years, blockchain technology has significantly improved the possibilities, allowing the achievement of immutability and transparency of votes. Despite the interesting properties, the young age of the blockchain brings along scalability and anonymity issues. To complete the overview of the topic, the literature on attacks against e-voting systems is surveyed, allowing one to categorize the objective of attackers and the methodologies employed. Through analyzing current proposals and possible attacks, we conclude that the paradigm of electronic voting needs additional research to be considered secure to use in a general election setting. Further advances in technology can eventually make electronic voting a valid alternative, but special caution must be taken in protection against new attacks the same advances can lead to.

Keywords: electronic voting, verifiable systems, blockchain e-voting, attacks

Abstract in lingua italiana

L'interesse nel voto elettronico è aumentato negli ultimi anni come alternativa alle elezioni su scheda cartacea grazie ai vantaggi significativi apportati in termini di efficienza e propensione all'errore. Nonostante queste premesse, i sistemi utilizzati in elezioni effettive sono risultati soggetti a errori software e vulnerabili ad attacchi da parte di entità malevole. La sfida presentata dal voto elettronico può essere ricondotta alle molteplici proprietà richieste nella progettazione del sistema, con alcune di esse apparentemente in conflitto. In questa tesi, vengono esaminate la sicurezza e le proprietà delle nuove proposte per un sistema di voto elettronico, differenziando in voto al seggio e voto da remoto. Una soluzione praticabile è rappresentata dai sistemi verificabili, nei quali un elettore riceve la prova del proprio voto da verificare dopo il termine delle elezioni. Tuttavia, questa soluzione può risultare particolarmente difficile da progettare correttamente in un ambiente remoto a causa della possibilità di coercizione. Negli ultimi anni la tecnologia blockchain ha notevolmente migliorato le alternative, consentendo il raggiungimento dell'immutabilità e della trasparenza dei voti. Nonostante le proprietà interessanti, l'immaturità della blockchain porta con sé problemi di scalabilità e anonimato. Per completare la panoramica dell'argomento, viene esaminata la letteratura sugli attacchi ai sistemi di voto elettronico, consentendo di classificare l'obiettivo degli aggressori e le metodologie utilizzate. Analizzando le proposte attuali e i possibili attacchi, concludiamo che il paradigma del voto elettronico necessita di ulteriori ricerche per essere considerato sicuro da utilizzare in un contesto elettorale pubblico. Ulteriori progressi tecnologici potranno eventualmente rendere il voto elettronico una valida alternativa, ma è necessario prestare particolare attenzione alla difesa contro i nuovi attacchi a cui possono portare gli stessi progressi.

Parole chiave: voto elettronico, sistemi verificabili, voto elettronico basato su blockchain, attacchi

Contents

Abstract **i**

Abstract in lingua italiana **iii**

Contents **v**

Introduction **1**

1 Preliminaries **5**
1.1 Properties of e-voting 5
1.1.1 Fundamental properties 6
1.1.2 Secondary properties 8
1.2 Cryptographic background 9

2 Review of current e-voting systems and research **13**
2.1 End-to-end voter verifiable systems 14
2.2 End-to-end remote verifiable systems 19
2.3 Blockchain based e-voting systems 26
2.3.1 Why the blockchain? 26
2.3.2 Review of the systems 28
2.4 Considerations on current systems 40

3 Attacks against e-vote systems **43**
3.1 Introduction 43
3.1.1 Public elections 44
3.1.2 Private voting 44
3.2 Types of attacks 45
3.3 Review of the attacks 46
3.4 Considerations on the attacks 54

4 Conclusions and future developments **57**

Bibliography **61**

List of Figures **71**

List of Tables **73**

Acknowledgements **75**

Introduction

Voting is a fundamental process in a healthy democracy, where citizens can elect government and leaders. Even in smaller settings, it is essential to express ideas and take decisions. In the last few years computer systems have changed multiple aspects of modern society, making room for new opportunities and improving its processes. Voting is not an exception to this revolution but has been resilient to changing its established method and practices. This is mainly due to the delicate nature of voting and its roots laying in the core of society, which makes the act of expressing a preference secretly and without interference a fundamental right. The process of voting has historically relied on paper ballots, a system that can assure a high level of secrecy of the vote cast. In contrast, traditional voting requires the votes to be tallied and counted manually involving numerous staff members. This leads to high costs, a lot of time needed for tallying, and numerous errors in the process that can be traced back to the human component involved. As in other fields, the main advantage brought by involving machines in the voting process is the automation of the tasks which would otherwise be conducted by a human with the consequent disadvantages mentioned before. The first machines for voting were introduced during the sixties for accurately recording the vote without the doubts given by a sign manually placed on a paper ballot. Punch cards were provided to the voter which would punch the preferred candidate box and the machine would interpret the punched box. Through the years, following the need for accurate recording, optical scanners were introduced as an evolution to punch cards. In the last years of the twentieth century, further developments in computer technology led to the introduction of electronic systems also for casting the vote with the rise of Direct Recording Electronic machines (DRE), which would achieve large success and utilization in large democracies such as United States, Brazil, and India. These systems allow a voter to interact directly with the machine to cast the ballot, via a touch screen or a series of buttons, without the necessity of a physical ballot. The success of these systems was highly impacted by the issues which were found on multiple of these machines, involving errors, bad programming practices, or highly exploitable vulnerabilities [50, 84]. Despite the studies concerning the risks of these types of machines, a large number of countries are still employing them, including

the ones whose machines were the subject of the mentioned research. The diffusion of the Internet and personal device in the last twenty-five years allowed for the introduction of another way of voting, which would allow a person to vote wherever he/she was and with his device. This new paradigm named remote voting (or Internet voting since the Internet is the main network used) represents the most flexible, but also the most insecure, possibility for voting. Votes are recorded directly by the user system, which can be easily compromised without the owner being aware. Networks and servers where the votes are recorded are also vectors of attack for a malicious entity. This type of setting is the most challenging for designing a secure voting system since adequate countermeasures should be set in place against a large number and types of threats. Estonia is the only country that currently permits remote voting as a channel in local and national elections. Other countries like France [53] and the United Kingdom [32] proposed similar alternatives, but they were rapidly discarded as insecure or excessively complicated to implement. In recent years, with the rise of cyberespionage and states acting as malicious actors, the security of the voting process has come into the spotlight. Foreign countries can try to manipulate an election outcome to favor a particular candidate or try to sabotage the election to discourage citizens to vote. Even in smaller settings, such as a local council, a university, or a company board, a dishonest figure could try to change, or pay someone to do so, the votes cast in the system to obtain a deliberate result. To cope with these challenges researchers tried to design systems involving cryptographic primitives and, more recently, the use of blockchain technology.

Why the need for another survey?

The topic of e-voting has always got interested the research community and several surveys have been presented in the last few years exploring different aspects of the topic.
In [61] the authors explore the properties and modalities of e-voting by analyzing and comparing various proposed alternatives. However, due to the rise of blockchain-based systems in the years after the proposed survey, it can be considered outdated as regards almost ten years of research on the topic. The review presented in [82] also explores and confronts some of the proposals for e-voting systems, but the comparison looks incomplete since only a few systems are considered. It should be pointed out that the objective of the paper was only to introduce the properties and issues of an electronic voting system and not to represent a complete view of the proposals. However, it cannot be ignored that an important and developing trend as blockchain-based systems is not addressed. A similar survey is presented in [74], which focuses on remote voting by first exposing the problems

of this type of system and the cryptographic methods used to try to address these issues. As for the review cited earlier, only a few systems are considered and blockchain-based voting is not debated.

Regarding blockchain-based e-voting systems, recently few surveys have been published focusing on this category of schemes. In [1] the authors compare systems using blockchain by presenting the properties addressed by each system exposing advantages and disadvantages. The works presented in [79] and [41] explain in deep the functioning of the blockchain and the opportunities given by this technology while also comparing recent proposals on the topic. The authors of [39] also perform a complete exposition on blockchain technologies and a review of a number of recent papers related to blockchain-based voting systems. The works previously cited on blockchain-based systems are all complete and thorough, but only examine a possible type of solution. The purpose of this thesis is to give a full overview of current solutions while also addressing the literature on attacks against e-voting systems. The last topic, in particular, seems to have never received full attention in the various survey proposed. Only [5] partially focuses on attacks but results in any case limited and does not address recent vulnerabilities as the ones impacting systems employing the blockchain. An interesting paper is the review in [48] which describes and categorizes various systems from a cryptographic point of view, highlighting the advantages and disadvantages of each approach. Potential research directions are also explored in the review but, as with other surveys, attacks are partially addressed and not in a systematic way. Instead this thesis aims at categorizing and comparing these attacks, starting from current literature on the subject.

In the first chapter of this thesis the properties required for an electronic voting system are addressed, specifying their impact on the voting process, and a brief background on the cryptographic techniques commonly used in these systems is given. Then the state of the art in designing and implementing an e-voting system will be presented, exploring the categories of voter-verifiable systems and blockchain-based systems. Each system will be compared with the others highlighting properties, advantages, and disadvantages. In the third chapter attacks on electronic voting systems will be confronted and categorized by the methodology used and possible threats posed to the considered system. Last, in the conclusions, some considerations and observations on the future developments of electronic voting will be made.

1 | Preliminaries

1.1. Properties of e-voting

The voting process and its connected aspects are very complex and include several requirements to be defined as complete and usable during public elections. However, a list of general properties and requirements regarding elections using electronic devices and communication networks can be drawn up, with some of them being also common with paper ballot methodologies and others introduced using these technologies. Previous research as [33, 60] has tried to propose a series of requirements with [33] deeply analyzing them considering actors, type of requirement, preconditions and giving a complete description of the user and functional component of the requirement. The Council of Europe has also published [63], containing a series of recommendations on the properties of all the aspects of e-voting. This document is currently the main intergovernmental source in the field and has been voluntarily adopted by many states such as Norway, Estonia, Switzerland, and Belgium. In this paper, two sets of properties are identified, which can be divided into fundamental and secondary. The first set contains the core requirements for e-voting systems, which are mostly the ones in common with a classic paper voting system. The second set instead deals with issues related to the use of technology and contains requirements necessary to implement a large-scale e-voting system such as the one used for a national election.

1.1.1. Fundamental properties

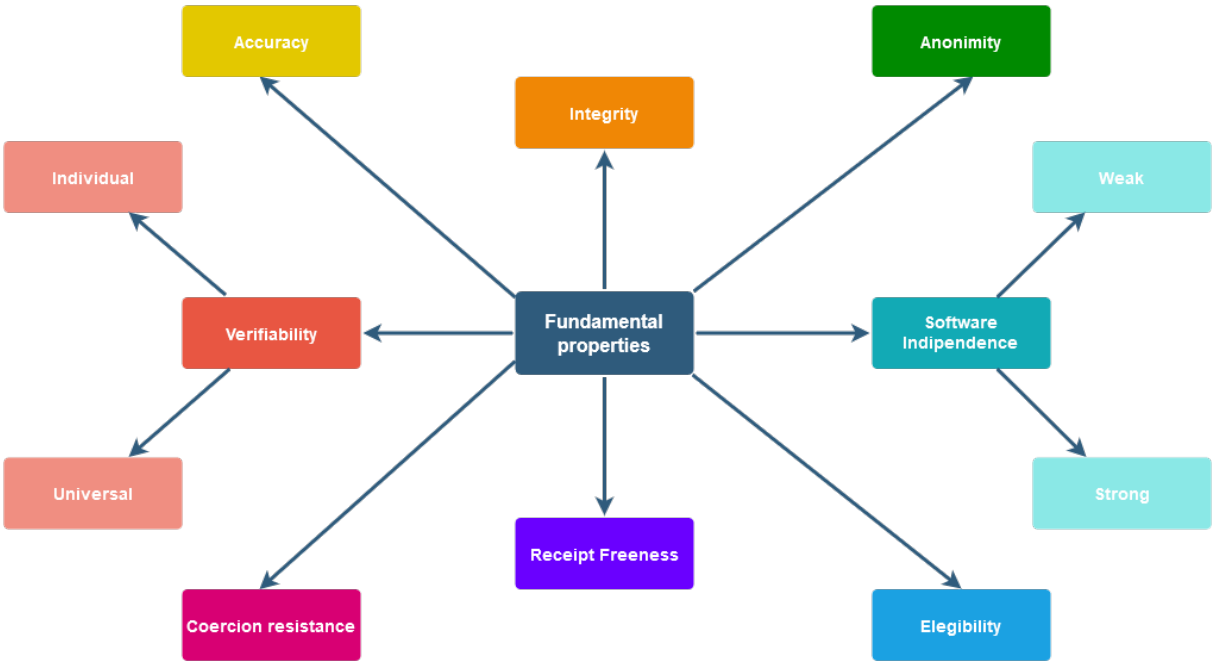

Figure 1.1: Scheme for e-voting fundamental properties.

In the democratic voting process, **integrity** and **anonymity** (or privacy) can be considered the two core properties for representing the will of the voters. Integrity defines that casted votes should not be able to be modified, created, or destroyed without detection and anonymity that no one should be able to tell how a participant in the election has voted. In a paper-based election generally, these properties are assured using electoral ballots and ballot boxes, but in an electronic setting, these properties are more difficult to assure and sometimes even in conflict. Indeed, it can be difficult to check if a vote has been modified or inserted maliciously without revealing the vote and consequently violating the privacy of the voter. Strictly related to these two properties is **eligibility**, which is defined as the property for which only valid voters who meet certain criteria and are authorized to vote can cast a vote. This process is in general performed by an authority that has a list of eligible voters and can deny access to vote to unauthorized entities.

To perform a correct voting process the property of **accuracy** (or completeness) is also required, defined as the requirement of correctly recording the votes. In this scope, the advantage of automation provided by electronic voting can be recalled, since in paper-based voting human errors during the tallying of the ballots often cause the overall result to be recorded wrongly. A complete e-voting system should resolve this problem by recording

all the votes without errors.
Another important requirement is **verifiability** which can be divided into individual verifiability and universal verifiability. The first one requires that a voter shall be able to verify that his/her vote is correctly recorded and counted in the final tally. This can mainly be done by using a public bulletin board and providing a receipt that can be compared to the one published on it. Universal verifiability instead requires that voters can check that all stored votes are included in the final tally.
In contrast with the need for individual verifiability, the property of **receipt freeness** is required, meaning that a voter should not be provided with a receipt that proves how he/she voted to any other entity. This requirement is strictly correlated with the one for **coercion resistance**. Coercion happens when a person or entity tries to manipulate a voter to vote for a certain candidate or not vote at all. A system should be resistant to coercion by not allowing the voter to prove how he/she has voted. By providing a receipt a voter can easily prove to a coercer that he/she has voted for the indicated candidate. The contrast between these two fundamental properties requires engineering a way to allow the voter to verify his vote by giving him/her something different from a receipt containing the actual vote. This alternative receipt can be then used by the voter to check the correctness of his vote on the bulletin board.
The final fundamental property is **Software Independence**, which was introduced in [69] and is limited to electronic voting systems. A system satisfies it if "an undetected change or error in its software cannot cause an undetectable change or error in an election outcome". In particular, strong and weak software independence can be defined. The first is achieved if a detected change or error in an election outcome can be corrected without re-running the election, and the second is if there is no recovery mechanism. The main two approaches to software independence are Voter-verifiable paper record (VVPR) approaches and Cryptographic voting systems. In the first category the paper ballot is voter-verifiable because the voter completes the ballot and can attest to its accuracy before it is fed into an optical scanner and the paper ballot can be used in post-election audits of the optical scanner's electronic results. The other category provides detection mechanisms for errors caused by software changes or errors by allowing voters to detect when their votes have been improperly recorded. Receipt-based systems are also a way to achieve software independence but violate the requirement for receipt freeness.

1.1.2. Secondary properties

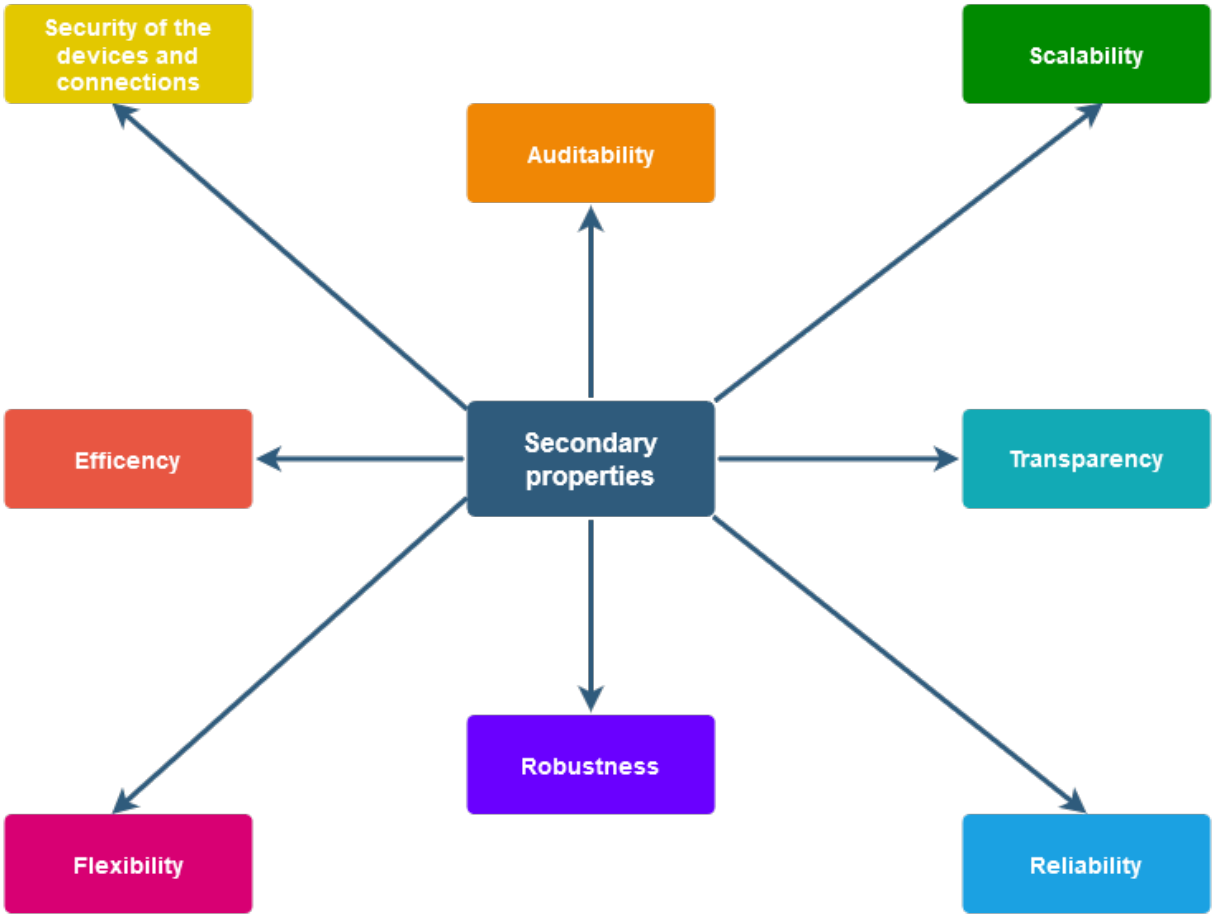

Figure 1.2: Scheme for e-voting secondary properties.

This set of properties as briefly mentioned earlier, mainly concerns organizational and technical aspects of the voting system.
The first property is **auditability**, which requires the system to maintain a record of the elections to be accessible in an audit process. Having the possibility to perform several audits is crucial to be able to discover possible errors or unwanted behaviors.
Transparency is also an important property for an e-voting system. The voter should have a general understanding of the system and its methodologies to effectively trust it. Low trust in the system would eventually result in a low turnout during the election period, which should be prevented.
Scalability and **efficiency** are technical requirements related to the adoption of a system for an election where the number of voters is high. Indeed, the system should do computations in a reasonable time and not be difficult to implement on a large scale concerning organizational and technological requirements. This includes also being reasonably ex-

pensive in relation to the size and compared to a traditional paper-based system.
A system adequate for general election requires also to function correctly. **Reliability**, that is the resistance to malfunction and errors, and **robustness**, the correct functioning regardless of a partial failure in the system represent these requirements. An electoral system failing during the election period could lead to voters not being able to cast their ballots and eventually not voting at all even if the system is restored.
Given the nature of elections, **flexibility** in the systems setting and ballot types is important for an electronic voting system. For an organization or the government would be highly inconvenient to have a system that supports only a type of election given also the variety of types of voting present even in a single country.
In the end, a system should address the **security of the devices and the connections** which includes that the equipment should not be infected with malicious software, and connections, if used, should be encrypted and not accessible by other entities. This property, if lacking, can lead to most other properties being compromised or completely invalidated. Without it, an opponent can compromise the system leading to threats to integrity, anonymity, and other properties depending on the correct functioning of the system. This property can result to be difficult to achieve especially in those e-voting models depending on the voter device and an Internet connection. A compromised device can modify the vote cast by the voter and show the voter the correct vote or an attacker monitoring an unprotected network can intercept the vote, modify it, or destroy it.

1.2. Cryptographic background

Cryptographic primitives and methodologies are extensively used in electronic voting systems to achieve the properties described in the previous paragraphs. While common cryptographic algorithms are still employed, additional concepts and schemes are introduced to cope with the challenges of the voting process. Especially anonymity requires particular attention and multiple methodologies try to ensure it while still respecting the integrity and privacy of the vote. Here the most significant concepts will be briefly presented to allow for a more in deep understating of the e-voting systems that will be examined in the next chapters.

Mix-nets The concept of mix networks was introduced in [21] and consists in routing protocols that allow almost un-tractable communications by employing a chain of proxy servers called mixes. The mixes take in messages from multiple senders, shuffle them, and send them out in random order to the next mix node or out of the network if the

node is the last one. This allows breaking the link between the sender and the receiver, making it harder for an entity that monitors communications to trace it and violate its anonymity. In addition to this, the possibility of the presence of a malicious node is mitigated since each node knows only the one from which it receives the messages and the next destination. The network also uses a "Russian Doll" layered encryption where each message is encrypted to each proxy using public key cryptography. In this way, every node which receives a message peels off a layer of encryption using its key to reveal where to send the message next. This scheme permits the achievement of un-traceability if all but one node are compromised by a malicious actor. Technologies that use onion routing or key-based routing like the Tor network are based on this concept.

Homomorphic encryption This type of encryption allows performing computations directly on cyphertexts without the need for decryption of encrypted data. The computations performed on the cyphertexts are still in an encrypted form which, when decrypted, will result in an output identical to the one that would be obtained by performing the same operations on the decrypted data. Homomorphic encryption is used in privacy-performing applications since there is no need to retrieve the original unencrypted data to perform computations on it. Homomorphic encryption incorporates several types of encryption schemes which differ from each other by the classes of computations over encrypted data. The most famous are the RSA cryptosystem [67] and ElGamal cryptosystem [27] for modular multiplications while the Benaloh cryptosystem [10] and Paillier cryptosystem [65] are the most common for modular additions. The last one, in particular, is widely used in the schemes proposed for electronic voting.

Group and Ring signatures The scheme using group signatures was first introduced in [18] and consists of a methodology for allowing a member of a group to anonymously sign a message on behalf of the group. In this way a signature will reveal only the group to which the signer belongs but not the exact signer itself, achieving a certain level of anonymity. A receiver can in any case verify that the signature is valid. Ring signatures were introduced in [70] as an improvement on group signatures. The scheme is similar to the previous one but, differently from it, the anonymity of an individual signature cannot be revoked in any way, and the possibility to use any set of users as a signing set without additional setup is introduced. Anonymity is also reinforced since one of the core properties of ring signatures is that it should be computationally impossible to determine from which of the set's members' keys the signature was produced.

Blind signatures They represent a type of digital signature introduced in [16]. In this form of signature, the content of the message is "blinded", or masked, before it is signed while still allowing the signature to be publicly verified against the original and unblinded message as for a regular digital signature. This methodology allows the achievement of a certain degree of privacy in settings where the signer and author of the messages are different entities. Blind signatures can also be used to obtain un-linkability, preventing in this way a signer from linking the blinded massage it is called to sign with an un-blinded message it has to verify later. In this case, the signer's response to a blinded message is unblinded before the verification so that the signature remains valid for the unblinded message. Blind signature schemes can be implemented employing different public signing schemes such as RSA [67] or DSA [45]. The main application of this class of signatures, other than voting systems, has been mainly digital cash schemes.

Zero-knowledge Proof This concept in cryptography refers to a method where a party, the prover, can prove to another entity, the verifier, that a certain statement is true without revealing additional information other than that the statement is indeed true. Essentially, it relies on the fact that is trivial to prove that a party possesses knowledge of certain information by merely exposing it, but it is challenging to prove the possession without revealing the information itself or some other additional information. A zero-knowledge proof of knowledge can be seen as a particular case happening when the statement consists only of the fact that the prover possesses the secret information which is needed by the verifier. Two categories of this method are present: Interactive zero-knowledge proof and Non-interactive zero-knowledge proof. In the first case, an interaction between the entity representing the prover and the one representing the verifier is needed for performing the validation process. Instead, in the second case, proofs are built upon any interactive scheme by relying on the instantiations of these proofs such as the Fiat-Shamir heuristic [29], which is the most widely used. However, it should be noted that computational assumptions are required for the validity of the proof.

2 | Review of current e-voting systems and research

The issue of designing a completely secure and reliable voting scheme based on electronic and computer systems led to multiple solutions for a system of this kind to be used in elections. Unfortunately, most of the systems designed and used by the governments during the last years of the twentieth century and early ones of the twenty-first were made by private companies using proprietary code and equipment. These systems, which included scanners, DREs, and remote voting, were usually found vulnerable to external attacks when analyzed by independent security experts. This led researchers to design systems that could assure higher levels of security without impacting the privacy of the votes. A new paradigm was created and researched to address this issue, introducing the property of verifiability for electronic voting previously described. Another advance in the research was made possible by the introduction of the blockchain in various fields of computer science. These two methodologies represent the core of the proposals made in recent years: end-to-end voting verifiable systems, distinguished in on-site or remote, and blockchain-based systems.

2.1. End-to-end voter verifiable systems

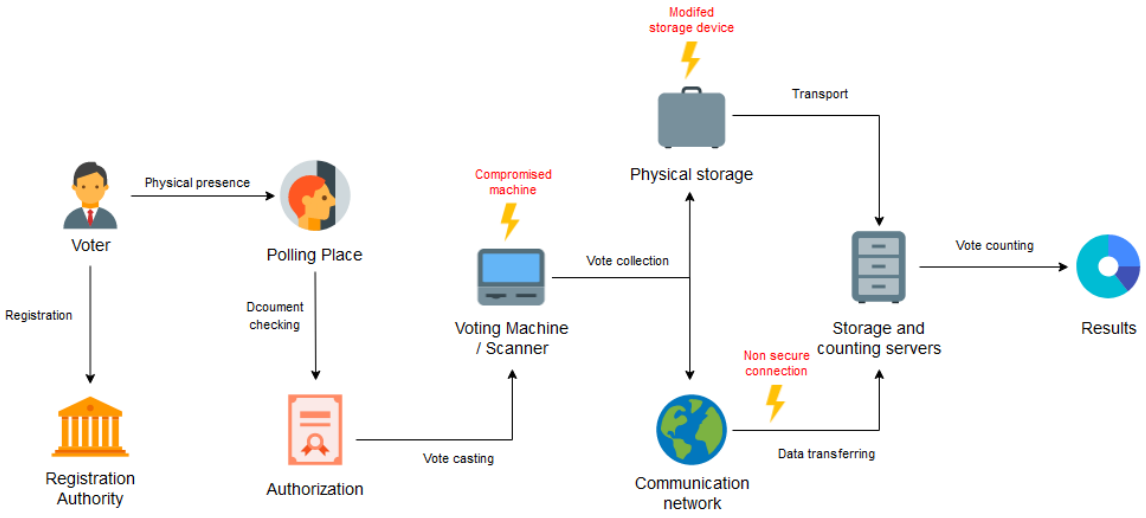

Figure 2.1: General scheme for a poll voting system with possible vulnerabilities

Starting from the twenty-first century, verifiability has always been one of the core properties that electronic voting systems tried to achieve. The concept of a voter-verifiable system has been introduced by [17] as the possibility how having voter-verifiable elections that let voters verify the election outcome, even if all election computers and records were compromised. This can be achieved by providing to the voter after he/she has cast a vote, a receipt that alone cannot prove to any third party the choices made but can be confronted to the one published on the election website. In this way, the voter can be sure that his vote has been correctly recorded. In particular, the author uses a layered receipt on which the vote is encoded to theorize this scheme implementation. Starting from this concept, end-to-end verifiable systems have been theorized and implemented, combining individual verifiability with universal verifiability. The first implementation of this type of system targeted election using optical scanners and DREs for recording the votes but, as presented in the next paragraph, this paradigm has been adapted also for remote voting.

An example of a system that tried to achieve verifiability on paper ballots is the Three-Ballot Voting System [68], which uses no cryptographic technique in the protocol. The methodology employed is pretty simple: each voter casts three paper ballots, with certain restrictions on how they may be filled out, and receives a copy of one of her ballots as a receipt. The voter is unable to use her receipt to prove how he/she has voted since the receipt does not reveal the votes cast. A voter can check on a website that his/her receipt

is present and the vote has been counted. To work correctly the three ballots must be filled following some rules that do not impact the free choice of the voter.
In the following years, more systems were proposed, applying the paradigm of having a receipt not directly linked with the vote cast.
The authors of [3] propose Scratch & Vote, a cryptographic voting system employing a random order for the candidates on each receipt provided. This receipt also contains a barcode for the receipt number and a scratchable surface. The voter receives a ballot with a detachable column containing the space for his/her vote, while the other part is a corresponding list of candidates. The voter keeps the column containing his choice, which after the casting is separated by the candidates' list, and can verify by using her name that the barcode and the choices correspond without revealing his/her real vote. In addition, a ballot can also be audited before casting by scratching the removable surface on the receipt part. This reveals the randomization values used for creating the ciphertexts used for the ballot encryption and lets a third party verify that the ballot is well-formed. The ballots are encrypted and tallied using Paillier homomorphic encryption, which assures the integrity and anonymity of the vote. The system is presented by the authors as very low cost and minimizing complexity. The fact the ballot is self-contained is also highlighted, since any helper organization or the voter can audit the ballot without interacting with election officials, before the actual vote casting.
The authors of [73] propose VoteBox, an e-voting system based on Direct Recording Electronic (DRE) machines that achieve end-to-end verifiability by having the machines print on a receipt the cyphertext of the voter's choices that will be afterward checked on a bulletin board. The system uses a broadcast network to log the encrypted votes and the logs of every machine. This solution represents a more robust alternative to having each machine maintain votes and logs. In addition, the security of the data is enhanced by providing hash chaining of the log entries, for which a tempering on a particular part of the data stored will result in a change also in the following entries. Homomorphic ElGamal encryption is used to encrypt the ballot on casting and the obtained ciphertext is printed on the voter receipt. As in the previous proposal, a voter can challenge a vote instead of committing the ballot. In this case, the system will publish the cryptographic value that it passed to the encryption function when it encrypted the ballot in question. Any party, including the voter, can then decrypt and verify the contents of the ballot without knowing the decryption key. The authors state that this proposal represents a complete system, combining features designed with human factors of both poll workers and voters in mind with the need for verifiability of the votes.

In [19] Scantegrity II is presented, an open-source system that uses letters in the choices box on the paper sheet to be marked by the voter to allow the verification. When casting his vote, he/she can write the letters on a small piece of the ballot with the ballot serial number, which is then removed and kept by the voter. Then if the letters next to his/her ballot serial code on the public board are different from the ones he/she recorded, he/she can contest the vote. The authors prove that anonymity is preserved since the letters and the disposition near the candidates are different for every ballot and assembled using a special board. In the back end, a structure called a switchboard, which uses only two permutations to anonymize the votes, permits less strict requirements on software and hardware components. An audit on the correct functioning of the switchboard can be performed by casting invalidated ballots and revealing the correspondent associations on the switchboard. Universal verification (every entity can check the correctness of the process and result) is also possible, after the election, via Randomized Partial Checking (RPC) which consists in randomly revealing a column of the switchboard and checking its correctness. The authors assert that this methodology allows a high degree of security, successfully preserving integrity and anonymity.

In [71] the authors propose Prêt à voter, a voting scheme taking inspiration from Chaum's concept for Voter Verifiable Systems. The overall design is shown to be similar to [19] with the utilization of a public board to verify votes but differs in the ballots design, ballot anonymization, and system back-end. The ballots consist of two parts (columns): the left one contains the candidates in random order and the right one the corresponding box to mark and a sequence of characters. After a voter cast his/her preference, the right side is detached and kept by the voter. Later, he/she can very his vote by checking that the box scheme associated with his sequence of characters is the same as the receipt and, in case it is different, contest it. The overall casting process is similar to [3]. For the back end, two alternatives are presented, each one having advantages and disadvantages, but both allow ballot audit and RPC for universal verifiability.

The first option is to use decryption mix-nets, but the authors point out that in this case ballots are generated by a single authority and if it is malicious, it can learn how a voter voted from the receipt.

The other possibility is to use re-encryption mix-nets where several authorities are involved in generating the ballot forms and the keys are used in a threshold fashion. The second approach, although more complex to implement, is presented as assuring more security in case of a malicious authority, the smuggling of a black ballot (the ballots are printed on the fly), and re-encryption networks can be run multiple times for auditing purposes. The authors of this system acknowledge that, besides the vulnerabilities of each of the two back-ends, both versions suffer from possible coercion mechanisms, especially

chain vote attacks, and randomization attacks.

In more recent years, the authors of [9] proposed StarVote, an end-to-end e-voting system that takes elements from [73] and uses them to build a sophisticated system. As in VoteBox, DREs are used, but only for letting the voters make the selections. The machine prints two paper elements: a paper ballot which includes a human-readable summary of the voter's selections and a random serial number, and a receipt that contains information about the machine used and time of voting, and a hash code that serves as a commitment to the vote but does not reveal its contents. The paper record can immediately be reviewed by the voter to make sure it contains the correct selections. The voter who wishes to cast the ballot then takes the paper ballot summary to the ballot box. The ballot box has a simple scanner that can read the serial number from the ballot and communicate it to a general controller. A voter can eventually spoil the ballot by returning it to a poll worker and having the ballot scanned as spoiled. The encrypted votes are posted on a public bulletin board where the voter receipt will correspond to a cryptographic hash of the encrypted vote. On the board, a voter can verify if his/her vote is present and consequently has been correctly recorded. If a voter spoiled a ballot, he/she should be able to see it on the board together with its decrypted selections. A risk-limiting audit process is then performed on the paper ballots to verify the correspondence between the paper records and the electronic records. In addition, in case of the need for a manual count, all the ballots casted and spoiled are kept. The authors state that, in this way, the system provides a triple assurance of the correctness of the results. Eventual inconsistencies revealed by one of these checks can be re-conducted to individual ballots that are mismatched, were counted wrongly, or were lost during a section of the voting process. The authors conclude that StarVote can be considered an evolution from the existing commercial voting systems used in elections and can lead to the further development of this technology in more directions as remote voting.

	Integrity	Anonymity	Individual verifiability	Universal verifiability	Eligibility verifiability	Receipt freeness	Coercion resistance	Auditability	**Methodology**	**Advantages**	**Disadvantages**
Scratch & vote [3]	•	•	•	•		•		•	Random order ballots Paillier homomorphic encryption	Autonomous auditing Limited official cooperation	Ballot and ballot secrecy attacks
VoteBox [73]	•	•	•	•		•	•	•	DRE machines Common local network	Common logs Hash chains	DRE vulnerabilities Network failure
Scantegrity II [19]	•	•	•	•		•	•	•	Random order ballots Permutation structure	Simplicity of the back-end	User filled receipts
Prêt à voter [71]	•	•	•	•		•		•	Random order ballots Decryption or reencryption networks	Good flexibility and usability	Coercion vulnerability
StarVote [9]	•	•	•	•		•		•	DRE machines Paper style ballots	Paper ballot audit Write-in votes	Coercion vulnerability

Table 2.1: Comparison between voter verifiable systems

2.2. End-to-end remote verifiable systems

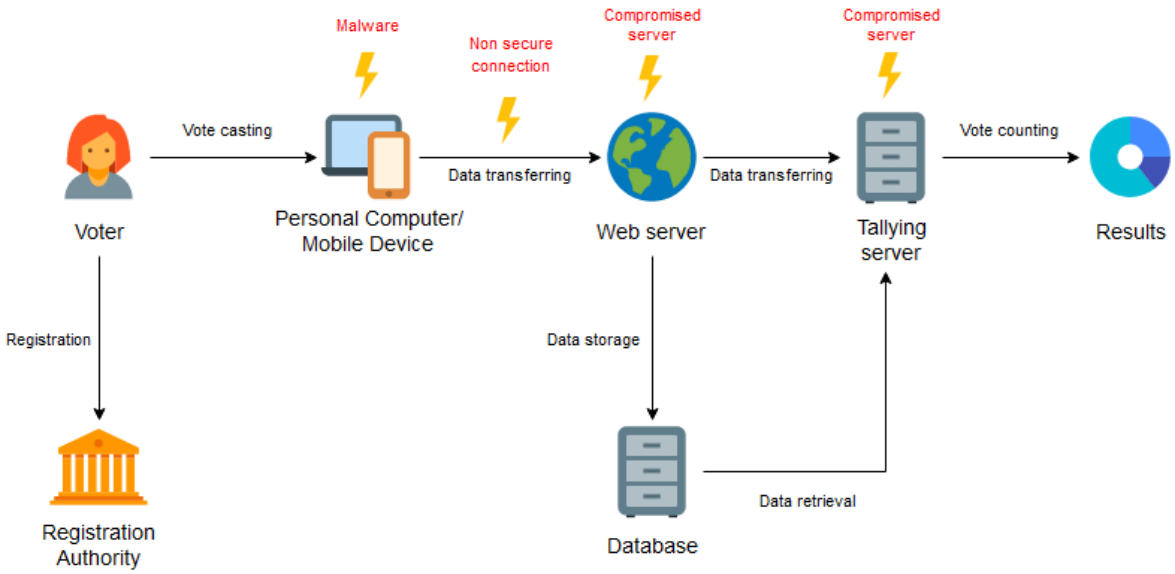

Figure 2.2: General scheme for a remote voting system with possible vulnerabilities

The need for verifiability in voting systems increased with the introduction of remote voting. Indeed, the presence of components external to the voting equipment as the user device introduces legit concerns on the correct recording and tallying of the casted vote. To cope with this challenge, researchers developed several solutions, focusing in particular on the need of preventing coercion. In previous verifiable voting systems, resistance to basic coercion was assured by the voting booth and a receipt that cannot provide to the coercer assurances on the choices made. In the case of remote voting a coercer can force a voter to cast his/her ballot in his presence and by doing this be sure that the voter has complied with his instructions. This issue has been confronted by designing systems that prevent it by letting the voter revote or being able to cast invalid votes. In the first case, the voter can cast one or more additional votes after the coerced one and only the last one will be counted, providing the voter the possibility to cast a legit vote. The second possibility is to design a system being able to detect and discard votes when a voter includes a particular element (a token or a password) in the casted ballot.
The first voter-verifiable remote system was Helios, an open-source web-based system presented in [2] to be utilized in small-scale online elections, where coercion can be considered low. In the Helios system, to allow everyone to audit a vote, vote encryption is done before authentication. After a voter has chosen the desired option, the system display to

him/her a hash of the cipher text of his/her choices. At this point, the voter can choose to cast the vote or audit it. In the case of an audit, also the randomness used to encrypt is shown to the voter, which can verify that the vote was encrypted correctly. After the vote, he/she can verify that the vote was correctly recorded by looking at the hash near his name or identification number. Since the hash can be given to a coercer to prove how a voter has chosen, Helios is not resistant to coercion attempts but represents the first implementation of a Voter Verifiable system adopting an online web-based approach to the issue.

In [24] the authors present Belenios, an online verifiable voting system built upon Helios. This system, as Helios, guarantees individual and universal verifiability and, in addition, provides eligibility verifiability, meaning anyone can check that ballots come from legitimate voters. Helios lacked this property, making it vulnerable to ballot stuffing, where a dishonest bulletin board could add ballots without anyone noticing.

This property is achieved by adding a signature system that still preserves the integrity and anonymity of the votes. Two other variants, Belenios RF and Belenios VS are proposed trying to address respectively the issue of coercion and malicious voting devices. The first deals with it by providing receipt freeness through having the server re-randomizing the ballots before publishing. This partially defeats coercion because voters can still sell their voting credentials. The second variant makes the voter scan with a device the ballot with his choices instead of making them on the device, preventing a malicious application to discover and modify the choices. Authors recommend, due to not being fully coercion resistant, the utilization of Belenios only in small-scale elections.

The authors present in [43] a formal definition of coercion resistance, which includes not only the case where a coercer instructs a voter to vote for a candidate but extends to randomization, force-abstention, and simulation attacks. Then, a theoretical scheme for achieving this property in a voting system is presented. This scheme is based on a unique per-voter anonymous credential, provided by the registration authority which is then included in the encrypted vote by the voter. A coerced voter can include a false credential, which is generated by an algorithm locally, in the coerced vote. The secret is then blindly compared with the register of voters' secret and only if there is correspondence the vote is counted in the tally. According to the authors, this scheme also eliminates the possibility of randomization or forced abstention attacks.

In [23] Civitas, an implementation of the scheme proposed by Juels, Catalano, and Jakobsonn, is presented. This system attempts to complete the previously described scheme and adapt it. These modifications are made:

- Civitas divides the registration authority into a registrar and a set of mutually

distrusting registration tellers. As part of this, it introduces the construction and distribution of credential shares.

- Civitas generalizes plurality voting by including approval and ranked voting methods.
- Civitas provides concrete instantiations of all the cryptographic components in the voting scheme, which were presented theoretically in the JCJ paper.
- Civitas introduces ballot boxes for vote storage instead of the bulletin board.

In addition to this, the researchers also studied the feasibility of this system and report their results. These experimental results show, according to the authors, that cost, tabulation time, and security can be practical for real-world elections. However, this system depends on a trusted voter client.

The authors of [83] try to solve the problem of not having untappable channels during the election by introducing masked ballots. Instead of receiving some credentials as in the scheme of JCJ, they receive masks for their votes. Before an election, a voter receives a mask from the election authority through an untappable channel and can use it for masking his/her vote. A coerced voter will use a fake random mask on his vote and his vote will not be counted. It is proved in the paper that votes masked with the legit mask are equal to the fake mask ones to an external viewer (they cannot be distinguished). Regarding verification, in the registration stage, the proof of correct encryption convinces the voter that the mask is correct while in the voting stage, the voter can use the private data to reconstruct the ballot and compare it to the posted ballot. The same can be done with a fake mask to convince a coercer that the voter has voted following his indications. The authors state that this approach particularly suits Internet voting from any lightweight device with network access. The main drawback is the need for the voter to securely obtain a single-use mask before each election.

In [22] the authors propose a coercion-resistant system named Selections, which archives this property by employing "panic passwords". In this system, the voter uses a panic password by a set of predetermined selected ones to cast a vote that will be discarded and a unique legit one for the intended vote. After anonymization, the votes committed to a panic password are eliminated and not considered in the tallying process. On registration, which can occur only one time for multiple elections, the voter chose a particular order in a set of words as his password and the same words in all other orders as panic passwords, making it easy for the voter to remember them during the elections. In this way, the authors state an over-the-shoulder resistance since a coercer cannot know if the password is the legit one. The main described benefits of Selections are: users can evade

coercion without computations, registration does not require a computer, tallying the votes is linear in the number of voters, and voters can have their registration efficiently revoked. However, the authors suggest further research on the trust of the device used.
The authors present in [72] a new voting protocol, Selene, which makes use of tracking numbers for vote verification, but voters do not learn their number until after voting has finished and the tracker/vote pairs have been posted to the bulletin board. The construction ensures a high level of assurance that the voter receives the correct tracker while ensuring that this is deniable to a third party. These tracking numbers are obtained by having a cryptographic term provided that generates the number. In case of coercion by employing a fake track number provided by the bulletin board the coerced voter obtains a fake term to decrypt the tracking number and to show to the coercer, hiding the real vote cast (for which the term will be received later). In this way, the coercer will verify a fake vote and the coerced voter the real one. The resulting scheme provides a good level of verifiability and coercion resistance while at the same time providing a simple-to-understand mechanism for voter verification. This scheme can also be retrofitted to existing voting systems without the need for major modifications or even be applied to a cryptographic election that has been contested.
In [57] VoteAgain is introduced, which is presented by the authors as the first revoting scheme that can handle elections with millions of voters. This system permits revoting a wanted number of times from different devices by authenticating with a polling authority, which provides the voter with a token that includes a voter identifier and ballot index, which is increased by one every time the voter revotes. The tally server identifies the last vote and takes it as valid. To avoid attacks where the coercer sees the number of votes in the filtering step, dummy ballots and voters are introduced. Any external auditor can use the Public Bulletin Board to verify that all steps in the tally and filtering phases were performed correctly. Compared to previous schemes using revoting or other cryptography techniques VoteAgain is demonstrated to be more rapid, reaching a quasilinear time in tallying, and in this way be suitable for elections with a large number of voters.
In addition to this, the authors provide a new coercion resistance definition and updated existing definitions for ballot privacy and verifiability to the revoting setting. It is then proved that VoteAgain satisfies all of them.
In [20] a new approach to building a coercion resistance system is presented with VoteXX, which introduces the nullification of votes by trusted agents (the "hedgehogs"). In this scheme, a voter receives two public-private key pairs. Without revealing their private keys, each voter registers their public keys with the election authority. Each voter may share their keys with one or more hedgehogs. During the election period, the voter, or one or more of their hedgehogs, can interact through an anonymous communication system to

nullify a vote by proving knowledge of one of the voter's private keys via a zero-knowledge proof without revealing the private key. In this way a coerced voter can signal to a hedgehog to nullify and even, as it is theorized in the paper, flip a vote. The nullification is done before the final tallying allowing all voters, even coercers, to verify that the selection is correct by having a Public Board which can be implemented using a blockchain. Multiple candidates voting is supported, and the system is supposedly resistant to malware on the voter device by having a two-phase voting process, where the voter verifies his vote on a different device. The authors assert that the presented solution makes fewer assumptions and protects against stronger adversaries than other techniques. It is also sustained that this scheme is inter-operable between different means of voting, achieving the theoretically best possible result for coercion resistance.

The authors of [52] present Ordinos, an end-to-end verifiable system providing fully tally-hiding, which is the property of not fully revealing the final tally while still providing the election results. The need for this property is justified by the authors as desired in situations where there are multiple rounds to not influence voter behavior by having intermediate election tallies. Other situations are when the set of voters is too small and revealing the full tally leads to a low level of privacy because a single voter's vote is "hidden" behind only a low number of other votes or when it might be unnecessarily embarrassing for the losing candidates to publish the number of votes they received. This process is achieved by employing homomorphic encryption and a multi-party computation (MPC) protocol, which permits the evaluation of a given function on a cyphertext vector and determines the result of the election. The authors formally prove that Ordinos provides privacy of the votes and suggest that, by achieving tally-hiding, provides more privacy by revealing less information. Full verifiability is also formally proved, even achieving accountability in the case of modifications by malicious entities. The authors implemented the systems resulting in an acceptable performance for several trustees and voters while having as the determining factor of the time needed the number of candidates. The system depends on a trusted client for vote casting.

In [40] Kryvos, a publicly tally-hiding verifiable e-voting is presented by the authors which include the researchers that previously designed Ordinos. Differently from Ordinos, which is a fully-tally hiding system, Kryvos is designed to allow each tallier to learn the homomorphically aggregated votes, while the public merely learns the final election result. This leads to a slight decrease in the provided privacy compared to the previous system but allows for no use of heavy cryptographic primitives as MPC protocols which tend to slow down the tallying process. The simplification of the cryptographic primitives also enables the utilization and implementation of complex ballots used in some countries as Instant Runoff-Voting, Majority Judgement, and many more. The difference in performance is

very obvious when compared with Ordinos, where the previous system needs much more time to tally the result for a determined number of choices than Kryvos. As for the other system designed by the authors, it is formally proven to be verifiable. The authors consider this system as a "practically relevant balance between privacy and efficiency compared to all previous approaches" and suggest the use in large-scale elections where the previously designed Ordinos would result in being heavily inefficient.

	Integrity	Anonymity	Individual verifiability	Universal verifiability	Eligibility verifiability	Receipt freeness	Coercion resistance	Auditability	**Methodology**	**Advantages**	**Disadvantages**
Helios [2]	•	•	•					•	Open audit system	Autonomous auditing	Low coercion environments
Belenios [24]	•	•	•	•	•		•		Helios Signature systems	Public eligibility check	Low coercion environments
JDJ05 [43]	•	•				•	•		Embedded anonymous voter credentials	Coercion attacks protection	Theoretical protocol High tallying overhead
Civitas [23]	•		•	•		•	•		Embedded anonymous voter credentials	Distributed registration authority Modalities of voting	Trusted voter client
WeBu09 [83]	•	•	•	•		•	•		Ballot masks	No untappable channel necessity	Mask non-reusability
Selections [22]	•	•	•	•		•	•		Panic passwords	Credential revocation process Linear tallying time	Trusted voter client
Selene [72]	•	•	•	•		•	•		Delayed tracking numbers	Voter verification simplicity Retrofittability	No real implementation
VoteAgain [57]	•	•	•	•			•		Insertion of dummy voters and ballots	Revoting Quasilinear tallying time	No pre-election coercion resistance
VoteXX [20]	•	•	•	•		•	•		Trusted agents	Retrofittability Compatibility with other techniques	No formal security proof
Ordinos [52]	•	•	•	•				•	Homomorphic encryption MPC protocol	Improved privacy Accountability	Trusted voter client
Kryvos [40]	•	•	•	•					Homomorphic encryption	Flexibility Improved performance	Less privacy compared to fully-tally hiding

Table 2.2: Comparison between remote voter verifiable systems

2.3. Blockchain based e-voting systems

In recent years a new paradigm for e-voting systems has gained popularity in the research community, which uses the blockchain to provide welcomed properties such as integrity and verifiability. Indeed most recent systems providing individual and universal verifiability utilize the blockchain as a building block. However, the presence of this new technology only limits the threats typical of remote voting, as the possible attacks on the voter device, before invoking the blockchain, are still possible. Before analyzing the proposals on these systems the reasons behind the interest in the blockchain should be discussed.

2.3.1. Why the blockchain?

Blockchain technologies first raised interest after they were utilized to create cryptocurrency systems, in particular Bitcoin proposed by Satoshi Nakamoto in [62]. It consists, essentially, of a decentralized digital ledger in a peer-to-peer network. A copy of the append-only ledger of the signed and encrypted transaction is kept by every participant (node) of the network. Thanks to the distributed nature of the system and the strong encryption applied, a blockchain enjoys various properties:

- **Decentralization**: The network has no central ruler, so each node has equal rights and obligations.
- **Immutability**: after a block of transactions is added to the chain, with the consensus of the majority of the nodes, it cannot be changed or deleted.
- **Non-repudiation**: by signing a transaction with his private key, the sender of the transaction cannot repudiate it since everyone can verify it with the sender's public keys.
- **Transparency**: in a blockchain, every transaction is publicly accessible (save private blockchains) except for identity information.
- **Pseudonymity**: every blockchain address used in the transactions is pseudo-randomly generated. However, some information about the real identity can be inferred by multiple transactions.
- **Traceability**: data in the blockchain can be tracked, mainly by following the chain of transactions.

These properties make blockchains suited for various uses other than cryptocurrencies and

there have been multiple proposals for e-vote systems based on them, especially internet-based systems. Another important element on which most of the proposed systems rely is smart contracts, automatic program contracts, which can be triggered automatically once the pre-set conditions are satisfied. These contracts can be executed in the blockchain, which provides a reliable environment for the execution. This allows the execution of trusted and immutable transactions by having pieces of software that are also immutable. Smart contracts can be applied to banks, insurance, and some processes of the voting action. Before approaching the proposals of recent years some explanations of the possible characteristics of a blockchain needs to be made.

Public blockchain vs private (or permissioned) blockchain In a public blockchain, which refers to the original proposal, there is no central authority, and every entity can send, receive, participate in the consensus process, and verify transactions. The most common consensus mechanisms include Proof of Work (PoW) and Proof of Stake (PoS). In a private blockchain, a centralized control authority is present, and only authorized nodes can participate. It is usually utilized by banks issuing private currency or companies for tracking purposes.

PoW vs PoS In the case of a Proof of Work consensus mechanism, nodes compete to be the first to commit the ledger by solving a complex mathematical problem. These systems generally cannot guarantee a high throughput and have high resource consumption, leading to scalability problems. Proof of Stake is based on the stake of each blockchain member, which determines the entities which will lead the validation. This allows high throughput and low resource consumption but weakens anonymity since the consensus is based on the stake and consequentially on the number of transactions of every member. Bitcoin is based on PoW, while Ethereum blockchains have recently moved to a PoS approach.

At present time, blockchain-based systems have been used for voting purposes as a pilot in Japan [42], Russia [54], and Switzerland [78] for non-binding elections or local electoral voting. Several specialized companies proposing electoral blockchain solutions have also been created with the intent to allow secure blockchain voting in small and medium-sized scenarios. However, most of the methodologies, documentation, and code used in the development of these systems have been kept secret by the owning companies and consequently have not been analyzed by independent security experts. In one case the analysis of a portion of the architecture used by the Voatz application has revealed serious security issues not depending on the use of the blockchain [76] that will be discussed in the chapter

on attacks against e-voting systems.

2.3.2. Review of the systems

Recent literature is rich in proposals for a blockchain voting system, which include different approaches regarding the type of blockchain, the consensus mechanism, secrecy, and anonymization of the votes.
In [55] a voting protocol using Bitcoin blockchain is presented, which according to the authors can solve fundamental issues of electronic voting thanks to the properties of the blockchain. The protocol is based on decentralization, even on authentication by introducing a Trusted Third Party (TTP). The TTP prevents the organization, which has control over eligibility, to forge votes or letting unauthenticated entities vote. This is achieved by exchanging hashes between the voters, the TTP, and the Organization permitting the validation of votes without exposing the voter's identity. The possibility of a consensus attack to disrupt the availability of the Bitcoin network is examined, leading to the conclusion that this attack needs massive resources and does not affect the security of the votes. The authors also address the possibility of not having enough miners in a private blockchain to provide reasonable throughput and suggest in that case to use of the public blockchain, which is prompted since it would provide enough miners without incentives.
The authors of [35] present a voting protocol for transmitting and recording votes for polling stations on a permissioned blockchain. Instead of using the Proof of Work to validate the blocks transmitted, a turn-based system is designed where nodes take turns to transmit and validate the votes. This allows achieving less computational effort and higher through compared to a protocol based on PoW. In the case of a node not being available when it gets a turn, the following node, using a timer, can start after an amount of time to craft its block without waiting for the token from the offline node, which will eventually retransmit later. The system also uses hashes and signatures to assure integrity and reliability in the blockchain. The authors then present a Python implementation and test it leading to proportional space and time results ranging from 1 to 500000 nodes.
In [8] a proposal for an e-voting protocol based on blockchain is made, which takes advantage of its core properties of transparency and immutability. After a brief review of the past attempt at general electronic voting systems and an introduction to the blockchain, the new protocol is presented, which differs from other proposals. Indeed, it employs different blockchains, one for each candidate, where the vote for that candidate will be appended. This approach prevents concusses in the blockchain and allows protest (or

blank) votes. The author highlights that the system assumes that the device is secure and that malicious software can eventually alter the vote on casting. Another drawback of this protocol is the inability to change a casted vote or directly revote.
The proposal in [12] utilizes the Bitcoin blockchain to implement a completely decentralized platform that allows to directly cast a vote in the blockchain without intermediate collecting levels. Authentication is performed using an Anonymous Kerberos protocol, which strongly guarantees anonymity. Another suggested alternative for authentication is the utilization of blind signatures. The protocol uses Digital Assets Coins (also simple tokens can be used) to represent the votes recorded on the blockchain. Employing the first choice has the benefit of having a clearly marked token to represent a vote that improves the post-voting phase while resulting in the simulations conducted only slightly more expensive than simple Bitcoin tokens. In both cases, the resulting cost is fairly low. The authors state that this implementation satisfies a large part of e-vote desired properties but does not support receipt freeness and consequently coercion-resistance. Moreover, is pointed out that seeing cast votes on the blockchain can influence the voters that have not voted yet. In [59] the authors present the first implementation of a decentralized and self-tallying internet voting protocol assuring maximum voter privacy using the Blockchain. The protocol does not rely on any trusted authority to compute the tally or to protect the voter's privacy and is written as a smart contract on Ethereum. Voters send encrypted votes to the smart contract using a Zero Knowledge Proof to hide their votes and to prove the validity of the vote, allowing them to obtain maximum privacy of the voters. The main drawbacks detected by the authors are the limited number of possible voters (the maximum recommended is fifty) and the lack of cryptography structures in the language used for writing the smart contract, which led to the inclusion of external libraries and the separation of the program into two smart contracts: one voting contract and one cryptography contract.
The authors propose in [56] a universal voting protocol, which does not depend on a particular blockchain platform and the presence of trusted third parties. The protocol requires each voter to own a pair of private and public keys utilized for signing the vote and verifying it. Blind signatures are also used for preserving voters' choices during the election. Three roles are present: the voter, which casts his vote, the organizer, which verifies the eligible voter and records their vote, and the inspector, which supervises the organizer's behavior. The authors declare that this protocol satisfies the majority of properties for an electronic voting system but is not coercion resistant since it uses a public blockchain for the votes. A permissioned blockchain is then suggested for avoiding this problem, underlining that this would impact transparency.
In [37] a proposal for a novel electronic voting system based on blockchain is made, which

addresses the matter on the district level. By using smart contracts on a blockchain platform a user can cast a vote in his district while the vote is recorded, tallied, and verified by a permissioned blockchain composed of the district nodes and other nodes belonging to the institutions participating in the election. A district-based approach to a permissioned blockchain has two main advantages according to the authors. The first is that, by being private, costs are minimized, and transaction output is maximized. The second concerns the modality of the vote, which does not use a web browser but collects the votes directly in the district making this system coercion resistant and not vulnerable to threats of online voting such as malicious actors exploiting the voter's device. In the final part, a discussion on the blockchain platform to use for implementing the smart contracts is performed, leading to the selection of Go-Ethereum as the most suitable due to its developer-friendliness and transaction rate dependency on the type of blockchain.
The authors of [38] propose an e-voting protocol employing Paillier's homomorphic encryption and oblivious transfer to construct a decentralized system. The authors state that the system proposed in this paper assures voter qualification, the anonymity of the voter's identity, non-repeatability, ballot eligibility, and ballot verifiability. According to the authors, a similar system can improve voters' confidence and reduce the resources needed for an election. In [46] the authors propose a blockchain-based e-voting protocol that conforms to the main requirements of the voting process and achieves end-to-end verifiability. The system adapts the Prêt à voter approach [71] and implements it using the Multichain platform to create a private blockchain. The system employs a strong authentication mechanism (fingerprints) assuring that the voter is who he/she claims to be and votes only a single time. The authors implemented and tested this system, obtaining the properties presented in the proposal. In the final part, a warning is made about the possibility of attacks on the malleability of the transactions, which can lead to the possibility of double spending on them.
The authors describe in [81] their proposal for a large-scale blockchain e-voting system based on smart contracts for Ethereum. In addition, a one-time ring signature is used to craft un-linkable signatures and achieve anonymity. ElGamal homomorphic encryption is then employed to effectively encrypt the casted votes. To achieve a higher throughput of transactions, the protocol does not use PoW to validate the blocks but instead opts for Delegated Proof of Stake (DPOS). This approach, based on delegates elected on consensus, makes the protocol especially fast and suitable for a large-scale election. The authors experimented with the system using an increasing number of candidates, leading to satisfactory results in particular for a small number of them. The protocol assures the majority of e-voting properties but does not support coercion resistance.
In [89] the authors present a scheme for blockchain e-voting based on smart contracts on

Hyperledger Fabric. This system is described as fully decentralized and without the need for a Trusted Third Party. Even without a TTP, the protocol can detect and correct cheating, be it solo or collaborative. It also supports re-voting and a complete distributed tally. The authors conducted a trial on an implementation consisting of twenty nodes which confirmed the desired properties for that number of nodes. This leads the system to be feasible for small to medium-scale elections but, according to the authors needs formal security proof. Other investigations are suggested on possible attacks and protocol parameters.

The proposal in [36] theorizes the blockchain as a transparent box and includes options such as casting an invalid vote or changing the vote cast. The system presented is fully decentralized, allowing each voter to collect the votes, validate them, and add them to the blockchain. A Central Authority is responsible for voter eligibility and, according to the authors, is the only possible point of failure since, if malicious, can cast votes for the ones that have not voted. The authors explain that a solution can be using a permissioned blockchain and involving multiple CA in the process.

In [15] the protocol "Verify-your-vote" is proposed, which the authors present as a "verifiable online electronic voting protocol using a blockchain". The protocol is built involving cryptographic primitives based on Elliptic-Curve Cryptography (ECC), pairings, and Identity Based Encryption (IBE). Differently from some previous schemes, in the scheme, the recipient of the voter can be used to check his vote while still providing privacy. The authors prove that the scheme ensures eligibility, privacy, receipt-freeness, fairness, individual and universal verifiability, but is not coercion resistant. Further directions of research are suggested regarding coercion resistance and the implementation of the protocol.

The authors describe [86] an Ethereum-based voting app using blockchain and smart contracts written in Solidity to perform the voting process. A user can cast his vote using an Android app or directly from an Ethereum wallet. It uses Ethereum coins which are given to the users to be able to cast the vote, which equates to a transaction. While supporting all major advantages given by a blockchain, the authors acknowledge that the problem of authentication is not addressed by using only the blockchain and the problem of scalability of the Ethereum blockchain remains a subject of further studies. A further problem highlighted is that Ethereum addresses do not provide full anonymization, which is necessary for a real election setting. In the paper, a full implementation is included with trial results that appear promising in a small-scale election.

In [75] the authors present a framework for a blockchain-based e-voting system introducing the concept of block creation and block sealing. Block creation refers to the moment when the block containing the transaction is created. In this methodology, neither Proof

of Work nor Proof of Stake is employed introducing a system based on hashes. Also, block sealing, which is employed to improve the security of the blocks, uses a hash on the blocks after the polling time is finished. The authors assert that these introductions do not impact the core structure of the blockchain, which is suggested to be of private or consortium type. In the paper, the voting process and the choice of the technologies employed are addressed. Then the process of creation, sealing, and collection of the result is explained focusing on the security addition given by the hashes. This framework appears to achieve multiple core properties of e-voting but unfortunately, the issue of coercion resistance is not discussed.

The author in [87] presents a scheme that offers methods for improving electronic voting security on peer-to-peer networks using blockchain technology. This scheme uses Elliptic Curves for the user credential model and allows a voter to replace a vote previously cast. The system was tested and designed on Linux systems in a P2P network, meeting the essential requirements for an e-voting process. The author advises that the cryptography schemes utilized can result in not being secure against a quantum computing attack since elliptic curves-based public keys are theoretically not secure against this type of threat.

The authors of [30] present a novel e-voting protocol based on blockchain which is stated as resistant to quantum attacks. This is achieved by employing a code-based public key algorithm requiring the resolution of the syndrome decoding problem, which is NP-complete and not subject to quantum attack. To resolve the problem of verifying public key certificates in a traditional public key cryptosystem, the authors introduce a certificateless traceable ring signature algorithm. The cryptographic elements employed and the transparent nature of the blockchain allow for an efficient and effective audit of the casted votes. The authors assert that this protocol is resistant, beyond quantum attacks, to replay, man-in-the-middle, counterfeiting, and modification attacks thanks to the signature procedure employed. The proposed scheme follows particular voting rules: voters are not allowed to abstain or vote for non-candidates, and only one vote is allowed. The first rule can be considered a limitation of the protocol. Through analyzing the scheme's performance, the authors concluded that the scheme is suitable for small-scale elections, where it has some advantages in security and efficiency while reducing efficiency when the number of voters becomes large.

In [80] the e-voting system based on Ethereum smart contracts Votereum is introduced. The system is empowered by the Ethereum platform, including one server that manages the entire system and another that handles all blockchain-related requests. An Election Authority is in charge of managing the election while a Registration Authority registers and verifies accounts on the system. The list of candidates, who owns no account on the system, is stored on the blockchain. A voter spends an amount of Ethereum currency to

cast the ballot as a transaction for which they will receive an id for verification purposes. The presence of this "receipt" makes the system not coercion resistant. After the election period has ended a smart contract tallies and publishes the results allowing public audit. During the voting process, the outcome can be kept secret on the blockchain until the voting stage has ended to ensure the fairness of the elections. The authors, while reviewing the security of the scheme against external attacks, found out that the system is vulnerable to DDoS attacks by having no way to identify the threat and defend against it. By analyzing the system performance and cost, the authors suggest the system for limited-scale election due to the fluctuation of Ethereum cost and the size of the network. To avoid these problems the use of a private blockchain is encouraged.

In [88] a system based on Ethereum smart contracts is proposed which, according to the authors, targets large-scale elections. They argue that actual blockchain-based systems don't achieve a series of desired properties that their proposed system, Chaintegrity, fulfills. This is performed by employing blind signatures for anonymous authentication and Paillier homomorphic encryption to perform operations directly on cyphertext (In particular additive operations). A hybrid component that combines the counting Bloom filter and the Merkle hash tree is also proposed to achieve fast authentication. To prevent adversaries from altering a voter's choice a code-voting technique is employed. By adopting this primitive, the voter and an auditing group (which can also review the voter client) share secret authentication codes, and the voter needs one more round of interaction with the auditing group in the ballot-casting phase. The authors state that the system is also resistant to Replay attacks, DDoS attacks, Man-in-the-Middle, and Sybil attacks. Some assumptions are instead considered in the paper: the election holders are semi-trusted (they cannot be compromised) and an anonymous network such as TOR is present to prevent IP trace attacks. The authors also warn that the system is not suitable for high coercion-risk elections since the blockchain is not considered suited for that type of election. The authors conclude that more research on blockchain e-voting is needed in the future.

In [14] the authors present a novel e-voting protocol, LOKI Vote, based on blockchain technology. This system is implemented on the Loki platform, which is based on the Monero blockchain proposing significant modifications to the Monero source code to ensure a high degree of privacy and provide a model for anonymous transactions and communications. This network is formed by Services Nodes which ensure the privacy and security of the network. These nodes operate the Low Latency Anonymous Routing Protocol (LLARP), which is an anonymous mix net combining the Onion Routing (TOR) and Invisible Internet Protocol (I2P). LLARP, according to the authors, fixes vulnerabilities of TOR and I2P protocols and provides a higher level of security and distribution than any existing

routing protocol. By using this network and other common cryptographic primitives as group signatures, the anonymity of the voters is ensured. As in other systems, fairness is guaranteed by encrypting the votes until the final tally. Loki Vote achieves coercion resistance by using fake credentials, which cannot be distinguished by a coercer, and casting another vote with the legit one since a revote is allowed. The authors suggest that thanks to the linear complexity of the protocol, it is suitable for large-scale elections. Further studies are also suggested on the actual implementation of the protocol and evaluation regarding the performance and scalability of the overall system.

The author of [26] presents an e-voting protocol based on the blockchain that is stated to achieve all the properties expected from secure elections without requiring effort from the voter. In particular, coercion resistance and receipt-freeness are ensured by means of a randomizer token, a tamper-resistance source of randomness which acts as a black box in constructing the ballot for the user. By having this randomness generated securely the voter can not prove to the coercer how he/she has voted while being able to verify his vote thank to the transparency of the blockchain. Even if the coercer asks for the randomizer token, functionality can be added which can "lock" a vote in the token randomizer, and, even if the coercer asks for a different vote to be cast, the locked vote will be posted on the blockchain remaining unlinkable to the coerced voter. Despite being described as a fairly simple item to use, the token randomizer can complicate the process of casting a vote for non-practiced users, resulting in problems for usability. It should also be accounted, however, that this method proves simpler than other anti-coercion mechanisms proposed. The proposed protocol also achieves fairness by having votes encrypted until the end of the casting period. Interesting results are reported on efficiency since the overhead for tallying the results emerges to be linear to the number of voters and the process for proof verification is rapid. These results, according to the author, make this protocol suitable for large-scale elections.

The authors of [66] present a DRE (direct-recording electronic voting machine) end-to-end verifiable system that can be deployed on a blockchain or using a cloud server. The authors state that the system preserves voters' privacy and integrity of ballots without any tallying authority or secure hardware storage even if the adversary gets temporary access to the DRE machine and prevents attacks such as ballot stuffing. Strong registration and authentication are assured by using biometric encryption, which is bound with a secret key using a dedicated algorithm. A blockchain, a cloud server, an Interplanetary File System, or a combination of the previous can be used to securely store the encrypted votes and security proof is provided for all implementations of the system. The authors warn that the Fuzzy Vault algorithm used for the verification of the fingerprint, which is their choice for biometric factors, has a non-negligible False Rejection Rate which needs

to be improved for real word use along with the implementation of more modality of the vote. The system achieves, according to the authors, good results in their analysis and experiments making it feasible for an actual election and becoming "the first end-to-end verifiable DRE-based e-voting system using blockchain" as stated by the authors.
The authors of [25] propose a voting network, SuffrageNet, built upon smart contracts on Hyperledger Fabric, which is chosen as a blockchain platform due to its avoidance of cryptocurrency and the absence of cryptographic mining operations. This network consists of two organizations: SUFFRAGE, and VALIDATOR. Each organization plays a specific role in the voting process. SUFFRAGE is the only one authorized to mint the ballots to manage identity and issue voter transactions, while VALIDATOR is the only one authorized the validation of the entire electoral process. The ballot is designed as a non-fungible token (NFT) that can only be minted by an authorized organization (SUFFRAGE). This approach guarantees, according to the authors, great flexibility on the ballots and type of elections. Also, the system is flexible, allowing multiple Certification Authorities to participate in the process to reduce the risk of the only CA being compromised. Similar to other proposed systems, this network is vulnerable to malicious software on the voter device, which is assumed as trusted.
In [7] a blockchain-based decentralized system, DVTChain, is proposed, which takes advantage of three smart contracts built on Ethereum to manage the electoral process: the voter contract, the candidate contract, and the voting contract. By using these three contracts, the voter registration process, voter authentication, and voting are done directly between the voter and the blockchain. During the registration process, the hash value of the voter's information is stored by the voter contract to provide integrity and anonymity. Each voter uses a provided coin to cast an encrypted vote for one of the candidates provided by the candidate contract. The voting contract adds this vote to the blockchain. To ensure fairness, the votes are decrypted only after the election has ended achieving verifiability by exploiting the transparent nature of the blockchain. The system is designed to accept votes cast from devices or polling places but is not immune to coercion since a coerced vote cannot be altered by the legitimate voter. The authors acknowledge also that the system does not provide a One Time Password and storing encrypted votes during the election increases the cost for the blockchain. Despite this last issue, during the tests conducted, the system provides acceptable gas costs for the properties achieved with respect to other similar systems.
The author of [4] proposes a voting system built upon Ethereum smart contracts. The blockchain, in combination with the use of servers for the voter list, the voting process, and voting counting, is employed to assure integrity and transparency of the voting process. The author state that his proposal introduces a new way to guarantee confidentiality

based on a database. The application generates a new account address using the member ID and password that a voter has initially input during the sign-up process. To ensure confidentiality, the accounts are encrypted and stored in a database along with the member ID values. Also, fairness is achieved by having voting results that can only be viewed after the time defined through systems parameters. The author state that the system achieves the fundamental properties of an e-voting system but suggests that maintainability of it could result difficult since contracts written in Solidity language cannot be modified after being deployed on the blockchain.

The authors of [90] present BCvoteMDE, a blockchain-based e-voting scheme that targets multi-district elections. This type of election generally includes voting at a district level and at the national level. The rules of elections can vary depending on the level. To support this type of election the authors introduce a two-layer blockchain where the first layer, the state layer, contains the ballots of the voters, and the second layer, the national, contains the results of the various states depending on the set of rules chosen. The system supports two types of voting: the classic multi-district election for two candidates and a "winner-takes-all" election where the winner of a district will receive all votes of that district at the national level. An Election administrator is present to set up the election and a National Election Authority is responsible for verifying that the users of the systems can vote in the election. The presence of these authorities can be considered a limitation to the full decentralization provided by the blockchain. The authors also assume a trusted and not compromised voter client. The full system is built upon a consortium blockchain permitting no use of any cryptocurrency and in this way boosting the affordability of the overall structure. In the test conducted, the authors found that the time needed to process the ballots increases following the number of voters and the number of nodes in each district.

In [31] a novel trust-less e-voting scheme based on blockchain technology that minimizes the intervention of authorities in the process is proposed. This system achieves free voting (the voter has not to pay the cost for the blockchain transaction) by using delegation servers and anonymity by using mix-nets. Delegation servers verify the signed encrypted vote of a user and add it to the blockchain paying the required fee. Dependency on authority is needed only during the preparation phase since anyone can fork an election from the list of eligible voters and everyone in the network can tally the final result. An anti-coercion measure is also put in place allowing a coerced voter to revote, revoke all his/her vote, and quit the election. This measure reduces coercion but is not considered fully suitable for a full-scale election, suiting better boardroom voting. Fairness is also achieved by encrypting the anonymized votes and decrypting them after the election period has ended. Desired features for usability are provided as the ability to retrieve a lost

or stolen key by using a seed phrase composed of words belonging to a set of predefined ones. The authors state that this system provides an efficient way to prevent election even in a large-scale setting, but further research is required regarding the delegation servers and the encryption to remove the exposure phase (the phase where the voter needs to expose their signed raw vote). The actual implementation of the proposed system is not provided.

In [6] a privacy-preserving e-voting system supporting score voting is proposed, which uses smart contracts on Ethereum. Score voting is a type of voting where the voter is allowed to assign a score from a defined range for each candidate instead of a yes/no, resulting in being more challenging to implement effectively. The system encrypts votes to assure privacy before the submission to the Blockchain. A malicious actor can modify the score value before encrypting it to manipulate the election result in favor of a certain candidate. To avoid this threat, the system allows voters to first prove that the submitted score lies in the predefined range before the vote is added to the Blockchain by using zero-knowledge set membership without putting to risk the anonymity of the vote. By setting up several tests to evaluate the system, the authors concluded that the scheme is suitable for small to medium-scale elections where low latency is desired. In addition to that it is pointed out that in all the tests performed the Ethereum network was able to achieve a 0% error rate no matter the size of the workload, proving high reliability.

	Integrity	Anonymity	Individual verifiability	Universal verifiability	Authentication	Correctness	Receipt freeness	Coercion resistance	Auditability	Methodology	Advantages	Disadvantages
Lee et al. [55]	•	•	•	•	•					Bitcoin Blockchain Trusted Third Party	No forged votes	Lack of miners
Hanifatunnisa et al. [35]	•		•	•						Permissioned blockchain Turn-based system	Computational efficiency	Few properties satisfied
Ayed [8]		•	•	•	•	•				Multiple blockchains	Protest vote Immunity to concusses	Trusted voter client No revote
Bistarelli et al. [12]	•	•	•	•	•	•			•	Anonymous Kerberos protocol Digital Asset Coins	Token as assets Low cost per voter	No coercion resistance Blockchain influence
McCorry et al. [59]	•	•	•	•						Ethereum smart contracts	No trusted authority	Limited number of voters Smart contract's limitations
Liu et al. [56]	•	•	•	•						Private/public keys Blind signatures	Independent protocol No trusted third parties	No coercion resistance
Hjálmarsson et al. [37]	•	•	•	•	•	•				District-based system Go-Ethereum smart contracts	Costs minimization Transaction output maximization	Limited to small sized countries
Hsiao et al. [38]	•	•	•	•	•	•			•	Paillier public-key cryptosystem Oblivious transfer	Ballot eligibility	No coercion resistance
Khan et al. [46]	•	•	•	•	•		•			Prêt à Voter scheme Multichain private blockchain	Biometrics	Blockchain attacks vulnerability
Wang et al.[81]	•	•	•	•	•	•	•		•	Ethereum smart contracts Homomorphic ElGamal encryption	Delegated Proof of Stake	No coercion resistance
Zhang et al.[89]	•	•	•	•	•	•	•		•	Fully decentralized protocol Smart contracts	No trusted third party Detectability of malicious votes	Small scale No formal security analysis
Hardwick et al.[36]		•	•	•	•					Transparent ballot box	Fully decentralized system Invalid votes	Central point of failure
Chaieb et al.[15]	•		•	•	•	•				Elliptic-Curve Cryptography Identity Based Encryption	Privacy-preserving receipt	No anonymity No coercion resistance
Yavuz et al.[86]	•		•	•		•				Ethereum smart contracts	Ethereum transaction Multiple ways of voting	No full anonymization Limited scalability
Shahzad et al.[75]	•	•	•	•	•	•				Private or Consortium blockchain Hash functions	Block sealing	No coercion resistance
Yi[87]	•	•	•	•						Elliptic Curves Cryptography Synchronized model of voting	Casted votes replacement	No quantum computing resistance
Gao et al.[30]	•	•	•	•					•	Certificateless ring signature Code-based public key cryptography	Resistance to quantum-attacks Audit function	Variation in efficiency No abstention
Vo-Cao-Thuy et al. [80]	•	•	•	•	•	•				Ethereum smart contracts	Secrecy of the partial outcome	No full coercion resistance Limited scalability

Table 2.3: Comparison between blockchain based systems

	Integrity	Anonymity	Individual verifiability	Universal verifiability	Authentication	Correctness	Receipt freeness	Coercion resistance	Auditability	Methodology	Advantages	Disadvantages
Zhang et al.[88]	•	•	•	•	•					Smart contracts with blind signatures Threshold Paillier encryption	Good performances Resistances to attacks	Semi-trusted election holders No coercion Resistance
Chaieb et al. [14]	•	•		•	•		•	•		Loki platform Non-Interactive Zero Knowledge Proof	Linear complexity Revoting	No implementation No evaluation
Dimitriou[26]	•	•		•	•		•	•		Bitcoin blockchain Randomizer token	Simplicity of use Large scale	Tamper resistant device Trust in authorities
Panja et al.[66]	•	•	•	•	•		•	•		DRE machines Biometric encryption	Strong authentication Flexibility of implementation	False Rejection Rate
Denis González et al.[25]	•	•	•	•						Smart contracts	Flexibility on elections and configurations	Trusted voter client
Alvi et al.[7]	•	•	•	•						Ethereum smart contracts	Multiple voting modalities Secrecy of partial results	No coercion resistance Increased cost
Ahn[4]	•	•	•	•	•		•			Ethereum smart contracts	Secrecy of partial results Strong encryption	Poor maintainability
Zhu et al.[90]	•	•	•	•	•	•				Two-layer blockchain	Flexibility on elections Affordability	Trusted electoral administrator Trusted voter client
Golnarian et al.[31]	•	•	•	•	•	•				Delegation servers Mix nets	Limited dependency on authority Vote update	Limited coercion No implementation
Alshehri et al.[6]	•	•	•	•						Ethereum smart contracts Zero-knowledge set membership proof	Score voting 0% transaction error rate	Limited scale

Table 2.4: Comparison between blockchain based systems

2.4. Considerations on current systems

Electronic voting systems and in general the paradigm of e-voting made huge steps towards completion in the last years. The systems presented earlier tried to address the inner issue of using electronic technologies in the voting process by employing different schemes, but all have as their main objective achieving full verifiability of systems. This can be seen, among all the necessary properties, as the one core objective of current e-vote research. A voter, to fully trust an electronic voting system, must be sure that his/her intentions were recorded correctly and counted in the tally. Consequently, having proof as a receipt gives to the voter, when compared with the counted votes, these assurances leaving few doubts of his/her vote being tampered with or the system behaving incorrectly. It has been examined that verifiability can be achieved with different degrees of difficulty depending on the setting of the voting action. In systems relying on on-site polls, a receipt can be physically obtained by a voter while in a remote scenario also the receipt is fully digital and can be more susceptible to tampering. The introduction of blockchain technologies had great importance in the development and the study of obtaining verifiability since the need for immutability and transparency required can now be delegated to the characteristics of the blockchain. It should not be surprising that, for this reason, most of the interest of the scientific community is now directed toward blockchain-based systems. However, while making it easier to achieve verifiability, the relatively young age of this technology carries with it implementation and performance problems that prevent it from being considered a full alternative for large-scale voting. Another important problem impacting blockchain-based systems for remote voting remains the lack of control on a part of the system: the voter device and the network connection. While not allowing tampering on recorded votes thanks to the immutable property of the blockchain, they can still suffer from tampering before the votes are recorded. Malware on the voter device can dynamically change to vote cast or sniff the voter credentials. Voters can also use an insecure network or be targeted by man-in-the-middle attacks. The nodes utilized for recording the votes can still be vulnerable to Denial of Service Attacks, not allowing voters to cast their vote and eventually discouraging them. Although these types of attacks, due to the small adoption of blockchain e-vote, have not been observed, the possibility is not remote, since a powerful attacker as an opposing nation can eventually compromise a large number of devices including network infrastructures. Once these issues will be resolved or mitigated this technology will certainly impact voting as it has impacted, in previous years, the financial world with the development of cryptocurrencies.

Remote voting systems and limitedly poll site ones can be susceptible to coercion and most of the proposals examined, while still providing verifiability, had to deal with the

issue of providing receipts that do not directly prove to an external how a person has voted. Also in the blockchain, if the basic case is considered, a coercer, thanks to the transparency of the blockchain, could trace back how a person has voted by looking at a certain transaction. In future research, these issues should be always taken into consideration as it represents a threat to democratic voting at the same level as a possible attack conducted by exploiting the voting system.

In addition to this, most current system proposals do not fulfill the property of software independence, which can be seen as a stronger assurance of the correct functioning of the system compared to verifiability alone. Indeed, verifiability helps in checking the correctness of the votes and the voting process but only systems that maintain paper ballots can currently assure software independence. Considering the previously presented systems, remote voting schemes cannot assure this property even when employing the blockchain, as software errors and manipulations can still occur as mentioned earlier. Given the strict requirement needed for achieving this property and the nature of remote voting, it appears that it will be nearly impossible to design such a system. However, high levels of verifiability can result in a system suited to a large number of situations where voting from remote is considered beneficial.

In my opinion, the best way to approach e-voting in future research is the one to try to adapt existing schemes for end-to-end verifiability to blockchain technologies and, from them, develop new solutions taking advantage of the benefits of each of the two paradigms. Indeed, the blockchain could serve the system by providing immutability and transparency while coercion be prevented by using techniques previously applied in end-to-end verifiable systems. Some of the previously examined proposals take inspiration from verifiable systems, but adapting them to use the blockchain turns out to be complicated.

In the near future, it is improbable to have a system that can respect all the properties of e-voting while still being affordable and adaptable to organize nationwide elections, but the trend of employing the blockchain while still referring to the previous successful end-to-end scheme can vastly improve the security, privacy, and usability of e-voting systems.

3 | Attacks against e-vote systems

3.1. Introduction

The electoral process has always been considered during history as one of the most delicate transitions in a democratic society and, consequently, the fear of interference in the legit outcome has always been present. Interference before and during the voting period by major actors occurred multiple times in history and even recently during the 2016 United Stated elections by the Russian government [58]. Before the introduction of electronic voting technology, this fear was mitigated by the assurances given by paper ballots which, in the general opinion, were difficult to be modified. Even coercion was limited thanks to the secrecy of the voting booth. Electronic voting, with the digitization of ballots, and the rise of remote voting open doors to a series of possible attacks given to the shortcoming and vulnerabilities of these technologies.

The possible threats menacing voting systems should be differentiated on the setting an attack can take place. Indeed, the menaces against polling place equipment such as DREs or optical scanners are different compared to the menaces against remote systems. Generally, the equipment which is used for recording votes in polling places should be interacted with physically for performing a modification or an attack, with the consequent need to access machines that should be strictly guarded. In the case of remote voting, multiple possible vectors of attacks are introduced. For example, a vote can be modified directly on the voter device by malware, by being intercepted on the network even on a compromised server. Most of the attacks in literature target these types of systems which, due to the open nature of the Internet and the presence of a device that is not controlled by the election organizer, results more difficult to protect against external threats. Before describing these attacks is necessary to identify the possible actors which could interfere in the election process. This will be done by differentiating public elections, where the population is called to vote for the government or important political decisions, and private voting, where a small institution such as a university, a committee, or a board is called to make decisions.

3.1.1. Public elections

Public elections are the setting where interference could potentially cause heavy effects on the election outcome. Candidates supported by malicious actors could win against the legit election winner, or the system could potentially be made unusable.

- **External threats**: Foreign nations could have an interest in supporting a determined candidate or party. This possibility is not remote as demonstrated by the attempts illustrated in [58]. It could be in the nation's interest also to not let people vote by mining the availability of the systems. The rise in recent years of cyber-espionage as a weapon against opposing states has made nations capable of performing complex attacks and representing, in the current world setting, the major threat against legit electronic voting elections.
- **Internal threats**: In a general election a candidate or a party could try to manipulate the election results by attacking the voting system. A single entity can be seen as less dangerous than a foreign nation as a threat, but it can not be excluded from the involvement of hackers paid by the entity or an opposing nation helping. A more serious possibility is that the election authority, which runs the elections, tries to change the outcome. In this case, the attempts could go completely undetected if the system is completely closed-source, the authority controls every aspect of the voting process and no external audits are performed. A solution is the use of verifiable voting systems but, as presented later, some attacks could be performed also on this type of system.

3.1.2. Private voting

Private voting is a setting where the possibility of interference caused by an attack on the voting system can be considered lower but not nonexistent, especially in the case of important institutions or companies. In this setting, coercion could result to be a major issue, in particular, when the election is ruined for deciding a much-desired position.

- **External threats**: As for public elections, this type of voting can see some interference from some entities outside the organization. In this case, the malicious actor is supposedly less powerful than a foreign power, but can still be a serious threat in the case of delicate matters like, for example, the election of the chief of a committee or a company. Rival companies can try to push for a compliant candidate and manipulate the election in his/her favor.
- **Internal threats**: In a small setting a candidate could try to gain the majority of

votes by performing an attack on the voting system used or trying or coerce voters. As in public voting, the organization could directly try to manipulate the election.

In this chapter, before analyzing the literature on attacks, a categorization of the attacks will be presented, ranging from the obvious ballot modification to more sophisticated attacks directly on the user's device.

3.2. Types of attacks

In literature, various attacks have been presented against various types of e-voting systems. In the next list, a categorization of types of attacks is presented, based on the methodology used by the malicious actor. Most of the proposals in the literature present multiple methodologies to accomplish different objectives during the same attack.

- **Ballot modification/substitution :** This type of attack consists in modifying the choices made on the ballot by a voter without the system noticing. This modification is usually conducted using malware that infects the voting equipment or a by intercepting a ballot on the network and substituting it. It is the most common type of attack since it fulfills the main objective of a malicious actor who wants to obtain a particular result. **[64]**,**[84]**,**[85]**, **[77]**,**[34]**, **[28]**,**[11]**, **[51]**,**[76]**
- **Ballot stuffing :** This type of attack fulfills a similar goal to the modification attack by adding fake ballots to the ones belonging to legit voters. An attacker adds, without the system noticing any discrepancy, a certain number of ballots which, in this way, can change the election outcome in favor of a certain candidate. **[64]**
- **Ballot destruction :** This type of attack can be considered the opposite of ballot stuffing, since an attacker, instated of adding ballots, destroys some legit one belonging to a voter whose votes are against a preferred candidate. **[64]**
- **Ballot secrecy violation :** This type of attack targets the privacy of a ballot cast by revealing the choice of a known voter. In this way an attacker, in particular a coercer, can know what a voter has chosen, undermining the principle of secrecy of the vote in democratic elections. **[64]**,**[84]**,**[85]**,**[34]**,**[76]**,**[13]**
- **Jamming :** This type of attack targets the availability of a system by making it unusable for voters to cast their ballots. This can lead to voters not being able to express their opinion and consequently a sense of distrust in the system. **[64]**

- **Ballot count modification :** This type of attack aims at modifying the election outcomes but, instead of acting directly on the ballot cast, it targets the counters of the voting system and modifies them. To not cause suspicion the votes received by the various candidates must remain consistent and not be drastically changed. Since modifications directly on the system are needed, a malicious component or software must be placed by the attacker on the machine counting the votes. **[49]**,**[84]**,**[77]**
- **Theft of user authentication data :** This type of attack consists in stealing the credential (or other authentication data) that a voter uses of accessing a voting system and cast a vote. In this way, an attacker can then vote in place of the person from which the credential has been stolen. Remote voting systems are mainly vulnerable to these attacks performed generally using malware on the voter's computer or smartphone. **[76]**
- **Interface modification :** This type of attack uses malware on the voter device to silently modify the code which displays the interface of the voting program/application. In this way, the malware can trick the voter into performing some operation that in reality consists of the ones wanted by the attacker. One example is making the voter believe he/she voted for a certain candidate when, in reality, the vote recorded and transmitted by the program is the one for the attacker's preferred candidate. **[76]**
- **Impairment of the system :** This type of attack aims at compromising the system equipment and facilities, leading to the eventual suspension of the election. The reasons for this type of attack can be similar to the ones for the jamming attack. **[85]**

3.3. Review of the attacks

In this paragraph, a literature review of attacks is presented, starting from the first attacks against physical machines to the ones about remote voting systems. Attacks against voter-verifiable systems are also presented, highlighting the vulnerabilities given by the receipts produced by these systems.
In [49] the authors present an attack against a popular optical scanner system used in various elections during the 2000s. Despite the language of the source code of the scanner having read-only access, due to the lack of cryptographic authentication, the researchers were able to modify some parts of the OS, in particular counters, states of the machine, and the ballot layouts. Then the crafting of a "time bomb" is described, a piece of code which, when triggered by certain conditions like the date or the number of votes cast,

swaps the results of two candidates. This malware bypasses the pre-election tests since it performs normally when the date is not one of the elections or the votes cast are only a few. It was pointed out that malware more complex could be created in this way, leading to the full compromise of the votes recorded by the machine. The malware could be loaded on the machine by compromising the memory card used by the system or accessing the serial port used normally for loading election data. In the final section, the authors highlight the importance of random audits to detect unwanted modifications of the software.
The authors in [64] present a series of possible attacks against the proposed Israeli e-voting scheme, which is based on contactless smartcards. In this system, the voter uses a machine to cast a vote that records it on an RFID card, which the voter can eventually verify on another machine, and then inserts the card in a classic ballot box. The authors exploit the RFID capabilities of these cards using relays and jammers to sniff, change, stuff, and destroy ballots. The first attack uses a relay placed between the verification terminal and the ballot box to enumerate all votes inside the ballot box and display the votes recorded on them violating the ballot privacy. The second named the Single Dissident Attack, lets the adversary suppress nearly all undesirable votes in a certain voting station while remaining virtually undetectable by placing a relay between the box, the voting terminal, and the verification terminal. The ballot stuffing attack instead is performed by placing a relay between the ballot box and the voting terminal. This allows the attacker to vote multiple times on the ballots that are already in the ballot box without a discrepancy in the voting count.
In addition to these attacks, a zapper can be used to invalidate the ballots in the box by emitting a powerful electromagnetic pulse. A selective denial of service is also possible by employing a simple jammer. All these attacks require physical access to voting equipment or the introduction of electronic devices in the polling site. This can partly be prevented by strict surveillance in addition to other security measures on the cards. Despite the presented mitigation, the authors consider the system not secure and unacceptable to replace a paper-ballot system.
In [84] an analysis of the Electronic Voting Machine used in the Indian election is performed, analyzing the design, exposing vulnerabilities, and theorizing two possible attacks. The design of the EVM is pretty simple, adopting the suggestion by experts that considered the complexity of the code one of the possible factors for vulnerabilities in a voting machine. Despite this shrewdness, the authors discovered serious issues in the machine that can be exploited to carry out attacks. This is mainly caused by the complete lack of authentication of the various components of the machine, leading to the possibility of substituting legit hardware with malicious one. Two attacks were performed by the authors: a Dishonest Display Attack and a Clip-on Memory Manipulator Attack. In the

first case, the authors replaced the legit display with a malicious one containing a hidden microcontroller, an antenna, and a Bluetooth module. By receiving a signal remotely, the microcontroller substitutes the casted votes with fraudulent ones. In addition, it displays always plausible but fraudulent vote counts for each candidate by using an online proportional boost algorithm. The second attack utilizes a device that is clipped on the EPROM memory of the machine. This simple device can steal votes by changing the records on the memory and is also able to download the complete vote record to a connected laptop, violating vote secrecy. All the main countermeasures placed by the election official are considered by the authors insufficient, in particular the ones regarding the physical security of the storage facilities and tamper-evident seals. Also, the use of secret source code is heavily criticized as being against fundamental security principles. In conclusion, the authors consider these EVMs vulnerable to serious attack and suggest taking a more transparent approach using voter-verifiable paper audit trail (VVPAT) or optical scanners.

In [28] an attack against a verifiable online voting system, Helios, is presented and discussed. The attack takes advantage of a buffer overflow in Adobe Acrobat Reader that allows the installation of a malicious extension in the target browser. The attack is performed by sending to the voter a pdf manifest for the election, which when opened by the reader uses the buffer overflow to silently install a malicious extension on the victim browser. This extension is capable to inject and modify JavaScript code in the election page, tricking the voter to think he/she is voting for a certain candidate while in reality the vote is cast for someone else. Having full control over what is displayed to the voter, the attacker can also implement functionalities to invalidate the verification of the vote and the audit feature. The authors suggest several methods to avoid this type of attack, like using updated software (which is useless if the malware uses a non-known vulnerability), disabling JavaScript in Adobe Reader, having third-party verifiers, or using hack-free dedicated hardware but also stated that the risk is not completely eliminated.

In [11] the authors present a powerful attack impacting all verifiable systems that provide the voters with receipts and suggest a simple solution to mitigate the threat. This attack, named “Trash attack”, uses as a potential vector of attacks the receipts that will not be verified by the voter after the election. In a normal setting where authorities do not know who will verify his/her receipt it does not impact these systems, as even checking a little fraction of the receipts would reveal any modifications of the votes. This changes if authorities know exactly who will not check the receipt by, for example, checking who refuses the receipt, examining trash cans near the polling sites, or social engineering. By accessing this information malicious authorities can modify the vote connected to a certain receipt and it would never be detected in any way.

The authors suggested a simple solution (similar to the ones employed by blockchains), which is to include in the receipt a cryptographic hash of the prior voter receipt. In this way altering a receipt, even if not verified, would lead to changes in the hashes of the following receipts making these receipts fail the verification process and detecting the tampering. Implementation for [6], [7], [8] are proposed revealing a fairly simple integration with existing systems.

The authors in [51] present and raise awareness of an attack against voter-verifiable systems, schemes where a voter can verify that his vote was correctly recorded and tallied. This class of attacks, named Clash attack, aims at providing different voters with the same receipt. As a result, the voting authorities can replace legit ballots with new ballots, and by this, manipulate the election without being detected. The authors present this attack against four different systems including Helios [2] and his variants, Wombat Voting, ThreeBallot [68], and VAV. All these systems similarly provide a receipt to the voter which is exploited by corrupt authorities to carry out this attack. Countermeasures for each of these systems are also presented. In the case of poll site voting systems (Wombat, ThreeBallot, and VAV) the simplest countermeasure is to print serial numbers on the receipt in advance and, by doing so, not allow the authority to provide the same receipt multiple times without being detected. In the case of Helios, which is an online voting system, cryptographic countermeasures are suggested. The authors propose then a formal analysis of accountability and verifiability of each variant of the Helios protocol (with and without countermeasures).

The authors show in [85] a series of attacks they were able to carry out during the open testing phase of the online voting system for overseas voters proposed by the government of Washington DC. Researchers were able to discover a vulnerability in the code for encrypting voted ballots uploaded by users and were able to inject a shell to compromise the web application server. This vulnerability alone permitted the injection of various payloads to perform several attacks. They were able to steal the server public key used to encrypt the ballots and the password of the database. This allows an attacker to change and reveal past and future votes, change the servers 'configuration and even retrieve the real voter credentials.

They were also able to carry out a separate attack on the network infrastructure by gaining access to the terminal server, router, switches of the internal network of the system, and even to some webcams of the testing facility. In their conclusion, the dangers and immaturity of internet voting systems are stressed, with the need for a significant advance in computer security to effectively secure them.

In this analysis presented in [77], authors were able to assist with the operations for setting up an election using the Estonian I-voting system and were able, via code analysis

and adversarial testing, to discover several possible attacks against the system. Two main attacks were theorized by the researchers exploiting the voter's personal device: a Ghost Click Attack and Bad Verify Attack. The first uses malware which infects the machine used by the voter. When he/she types the smartcard PIN to authenticate, the malware records the secret password. After the time allowed for vote verification via smartphone application ends, the malware checks if the smartcard is still inserted and available, opens the client in a hidden session, and, using the stolen PIN, casts another vote, invalidating the legit one. This attack is not possible on a major scale since a large number of revotes would be suspicious.
The second attack requires an impairment also of the smartphone application. In this attack the client malware modifies the vote cast by the citizen, displaying the QR code for the voted candidate. The smartphone malware app, on verification, reads the QR code and displays the user's intent to vote. This attack adds a layer of complexity and could rise suspicion when a code is read by another smartphone equipped with the legit app but is not unfeasible by large entities like foreign nations. Other attacks were also theorized by injecting an infected OS on the server during the configuration since DVDs are used for configuring the servers. This malicious software could then be used to activate itself on the counting machines and change the vote counters after the votes were decrypted. This attack could potentially go undetected since it acts after the description of the votes and is able to bypass various security measures, like checksum on the ISO of the operating system.
The authors concluded that "a state-level attacker, sophisticated criminal, or dishonest insider could defeat both the technological and procedural controls in order to manipulate election outcomes" and suggested the Estonian government discontinue the system.
The authors expose in [34] the vulnerabilities of the New South Wales iVote system, an online voting system used in Australia during the 2015 state elections. These discoveries are particularly critical because they were made during the election period and the attacks developed afterward could potentially change at least one seat in that particular election. The voter uses the credential provided to log in to the platform and cast his/her vote. He/she can then verify it using a telephonic service during the election period or after using a dedicated platform by providing credentials and a receipt. The authors found out that an external server used by the platform had a weak configuration and choice of encryption ciphers. This made it vulnerable to two attacks exploiting the weak cryptography choices of the connection used by the server: the FREAK and the Logjam attacks. The first is a TLS vulnerability discovered two weeks before the election allowing man-in-the-middle attacks while the second is a zero-day vulnerability allowing to attack any voter's iVote session. The authors exploited these vulnerabilities to replace

the JavaScript code downloaded from the vulnerable server with malicious code capable of changing the vote sent to the IVote main server. This code could also, without the voter noticing it, expose the vote that the voter intended to cast and sent it, along with the voter's authentication credentials, to a server operated by the attacker. The authors argue that, even if a verification service is present, it can be avoided in multiple ways such as delaying the vote send, making a fake verification service, or using a clash attack [41] on the receipts. Other problems regarding auditing, privacy, and usability are also highlighted. The authors conclude that the system is not secure and suggest the government avoid Internet voting until significant progress in the security field. It is also remarked the difficulty of secure online voting that, as in this case, can be subject to zero-day attacks, fragile standards for web development, and ineffective verification methods.

In [76] the authors analyze and present the vulnerabilities and possible attacks to Voatz, the first voting application used for a federal election in the United States. Voatz claims that the security properties of its system are guaranteed by a permissioned blockchain, biometrics, a mix net, and hardware-backed key storage modules on the user's device. The authors, having only limited access to the system documentation and the Android application, were able to perform attacks using the user device, network analysis, and theorize an attack on the voting server (of which documentation they were not allowed to access). It was discovered that, despite having an antivirus checking the user device for malware and rotting, these checks could be easily avoided by slightly modifying the client code without the Voatz app detecting it. This allowed to perform attacks to steal the user authentication data or modify the app UI to trick the user into voting for a particular candidate. A network analysis of the traffic between the application and the voting servers shows that the TLS encrypted packets' lengths vary depending on the length of the plain text of the choices made by the voter, making it possible to understand which candidate has been voted, in particular, if the ballot is blank. The authors highlight that these attacks can be performed assuming the full correctness of the server side and that the introduction of a blockchain, even is correctly used, doesn't mitigate them. In conclusion software independent systems using voter-verified paper ballots and Risk Limiting Audits are stated as remaining the most secure possibility.

The authors of [47] present the threat of transaction malleability in the context where the blockchain is used for e-voting purposes. Transaction malleability is a serious vulnerability in blockchain applications, that can be exploited by altering a transaction hash. In the paper, it is used to alter votes by mining a malleable transaction, which is semantically the same but syntactically to an honest one, before the honest transaction (in this case can be interpreted as a vote). This attack can lead to double spending, that is when the receiver of the transaction is made to believe that their transaction has been confirmed

whilst attempting to get another similar transaction (the malleable one) accepted by the blockchain network. In this way, the legit transaction (the vote) will be discarded by the network violating the integrity of the vote and the correctness of the electoral process. The authors simulated a transaction malleability attack introducing variable delays and tweaking the mining difficulty. The result led to the conclusion that a successful attack can be performed by keeping the hash rate and difficulty level constant across the two classes of transactions running instances and inducing some delays programmatically in the honest transaction release. The authors conclude that transaction malleability can produce an inconsistent blockchain allowing exploitation for malicious purposes.
In [13] a vulnerability and consequent vector of attack are presented regarding the confirmation pages sent by online voting systems after a voter has cast his choice. The authors discovered that, in the platform Simple Voting, while encrypted ballots choices are posted as fixed-length candidate IDs, the TLS encrypted confirmation page (which contains a summary of the choices) varied in length. This allowed to set up of a side-channel attack where, by sniffing the TLS traffic between the client and the voting server, a prediction on the candidate selected can be made in the case of a single-choice election. Also in more complex ballots, the model developed by the authors outperformed random guessing. However, for a large subset of ballots cast in an election, limited information in the form of certain combinations of candidates who were not voted for could be still obtained. After the authors' report, Simple Voting fixed the vulnerability by adding a random amount of padding bytes sampled uniformly in the interval [0,100). The authors comment that the fix, while substantially lowering the risk of the previously described attack, still poses risks in some limited cases and suggest a client-side generated confirmation page as the ultimate goal.

	Integrity	Privacy	Correctness	Availability	Type of attack(s)	Target	Attack vector(s)	Limitations
Kiayias et al. [49]	•	•			Ballot count modification	Diebold AccuVote Optical	Malware Memory card	Physical access Random audits
Oren et al. [64]	•	•	•	•	Ballot modification and jamming Ballot stuffing and destruction Ballot secrecy violation	Israel smartcard voting system	RFID relay RFID zapper Jammer	Physical access
Wolchok et al.[84]	•	•	•		Ballot count modification Ballot secrecy violation	India's Electronic Voting Machine	Substitute display Memory manipulator	Physical access
Estehghari et al. [28]	•				Ballot modification	Helios	Infected file Outdated software Malicious extension	Limited scope JavaScript disabled
Benaloh et al. [11]	•				Ballot modification	Verifiable voting systems	Unverified receipt	Chain of hashes
Kusters et al. [51]	•		•		Ballot replacement	Verifiable voting systems	Identical receipts	Countermeasures nullify the attack
Wolchok et al. [85]	•	•	•		Ballot modification Ballot secrecy violation System impairment	Washington D.C project	Shell injection vulnerability	Correct servers configuration
Springall et al. [77]	•		•		Ballot modification Ballot count modification	Estonian Internet Voting System	PC malware Smartphone malware Infected server	Small scale Non-infected smartphones
Halderman et al. [34]	•	•			Ballot modification	South Wales iVote system	Web server vulnerability	Verification service
Specter et al. [76]	•	•			Theft of user data Interface and ballot modification Ballot secrecy violation	Voatz application	Malware Compromised server Encrypted traffic	Partial system examination
Khan et al.[47]	•		•		Ballot substitution	Blockchain platforms	A malleable transaction	Required timescales
Brunet et al.[13]		•			Ballot secrecy violation	Simply Voting Online Platform	Vote confirmation page	Limited information Simple countermeasures

Table 3.1: Comparison between attacks against e-voting systems

3.4. Considerations on the attacks

In the previous paragraphs, it has been shown how a malicious entity can interfere with an e-voting system for different purposes. Despite the described attacks being only theorized and performed by researchers, it cannot be excluded that a real malicious attacker could perform or has already performed them since some of them allow not to leave traces and remain completely undetected. A possible attacker does not need time to fully understand the system, as demonstrated by [34] which crafted a successful man-in-the-middle attack during a live election period. One common factor between all the systems targeted by the attacks previously described, excluding the ones on verifiability, is the usage of closed-source code and the documentation of the implementation details kept private. This, according to vendors of these systems and some politicians, would increase the security of a system by not allowing an attacker to discover possible vulnerabilities thanks to the impossibility to access directly the source code. This belief, that is security through obscurity, has been widely refuted by the scientific community and important organizations such as the United States National Institute of Standards and Technology (NIST). Even considering cryptography, a basic notion is that a cryptographic system should be secure even if everything about the system is publicly known, except the key [44]. Hiding or even employing methodologies to actively obfuscate the code of the system can result in no additional security gained or rather losing the benefits given by the open-source paradigm. On the contrary, all the systems and protocols previously described in the literature review have their code and documentation publicly accessible, allowing further expand research starting from these works.

Another matter that has been widely discussed is the effective boost in security given by the utilization of the blockchain and the resistance against attacks. While the blockchain assures a higher degree of security thanks to its immutability, it is not free from possible vulnerabilities, as the one presented in [47]. However, the main issue remains the section of the system before the actual commitment to the blockchain and systems should design countermeasures to avoid impairment as discussed commenting on the review of the systems.

From the literature, it turns out that attacks are also possible against certain voter-verifiable systems by exploiting the receipt handed out. While these attacks pose a significant threat to a fair election, it should be noted that the attacker in this case is not an external entity but the election authority itself. In democratic countries this case can be considered very rare, finding a more suitable situation in regimes where a regent wants to trick the population into thinking democratic elections are held while still being sure to keep power. However, in regimes more simple and immediate methods like coercion,

social pressure, or ultimately violence are more likely to be used. A more realistic case would be a smaller setting like a council or a board meeting, where the entity in control of the elections could have an interest in rigging the voting in order to elect a preferred candidate. Remote voting is an alternative in this setting, but special attention must be paid to possible malware on the voter device which can in some cases bypass or counterfeit the verification as demonstrated in [28].

In the future, given the ever-greater funds some nations are employing in developing and strengthening their cyber-espionage capabilities, it is plausible that attack attempts against e-voting systems will be present or even increase. Along with the advances in computer science and the consequent new schemes that will be proposed, more vector and methodologies of attack will arise exploiting new techniques. An example can be made with quantum computing, where advances in this field could benefit electronic voting in the form of quantum cryptography resulting in more secure systems. On the other hand, these advances could possibly result in the development of more effective attacks employing quantum cryptography. Indeed, some of the cryptologic primitives commonly employed which are considered, at present time, enough secure could result vulnerable to quantum attacks. From this, protocols employing post-quantum cryptology should eventually be designed to increase the complexity of the voting systems. This reasoning can be made for all technologies employed by e-voting, leading to a kind of cat-and-mouse game between researchers and attackers. In my opinion, this can be considered the main difference between attacking a paper ballots system (or a system that partially relies on paper) and an electronic system. While paper ballots are not susceptible to innovation, electronic systems must be aware of and react to new threats given by the advances in the field of computer science.

Despite an actual attack by a malicious actor compromising an e-voting system was never been proven to have been performed, multiple attacks have been theorized and implemented questioning the actual security of these systems. It is short-sighted thinking that, in the near future, an attack could not be accomplished given new technologies and vulnerabilities to be discovered. Current and future proposals should consider the possible advances in the structure and complexity of the attacks, given the major boost yielded to espionage mentioned earlier.

4 | Conclusions and future developments

This study set itself the goal to survey and evaluate current proposals for an electronic voting system by assessing their security while also considering the attacks documented in the literature. The evolution in technology allowed researchers to design protocols able to challenge the needs of a complete voting system which can be a valid substitution for classic paper ballots. From examining the issue of secure voting, multiple properties resulted to be necessary for realizing a system suitable for public elections. They have been categorized as fundamental and secondary, with the second category representing properties that can be placed in the background for cases where the election is private and small-scale.

It has been observed that the trend in research is the search for verifiability in the systems starting from the proposals for poll site verifiable voting. By employing various cryptologic methodologies satisfying results are obtained for situations where the voter cast his ballots in fixed predetermined places. The difficulty of secure voting rises in the case of remote verifiable voting since the possibility of coercion and malicious software on the voter's device can impact the legitimacy of the electoral process. Proposals have been presented dealing with these issues by allowing revote and additional credentials, but these countermeasures have typically as an effect a decrease in the usability of the systems, with more measures and operations to be performed by the voter. Several blockchain systems proposals contributed to the search for verifiability, substituting a public bulletin board that needed precautions to preserve its integrity with an immutable structure as the blockchain. The proposals examined vary greatly in the details for the blockchain to be utilized showing the great number of possibilities that this technology offers, each one with advantages and disadvantages. However, these early visions show the immaturity of blockchain technology, facing important challenges in coercion resistance and scalability. Indeed, current proposals adopting a public blockchain typically have low transaction throughput while the ones adopting a private/permissioned blockchain tend to lose full decentralization by predetermining the validation nodes.

From the research conducted on the literature of the attacks against e-voting systems, a large number of possibilities and vectors of attacks have emerged, targeting all types of systems examined. It has been deduced that a single attack could have multiple objectives, which are not limited to the simple case of destroying a person's vote. Complex attacks can be set up to violate voters' privacy, undermine the availability of a system or even cause the full impairment of a part or all the electronic voting equipment. The objectives of an attack will depend on the final goal of the malicious entity which can be external or internal to the elections. An external actor as a foreign country could try to sabotage the elections by targeting the availability of the system while a candidate will tend to try to gain victory by changing votes or vote counters. In these cases, analyzed countermeasures were put in place, but most of the time were not able to stop or limit the attack performed. Despite no solid evidence of an attack targeting electronic voting systems has never been reported, the wide variety of attacks proposed in the literature, with most of them developed in a limited time, suggest that the possibility of such a risk is real and in the capabilities of powerful actors.

In conclusion, electronic voting has made huge steps in the right direction in the last few years. Paradigms and technologies have been developed to address the need for a system guaranteeing the same or even more assurance than classical paper ballots. By analyzing the literature, several systems have emerged as achieving most of the properties desired in a voting system but, however, one that fully guarantees the property needed to be researched. At present time, the most complete systems remain the ones that retain a paper component, as on-site voter verifiable schemes, achieving challenging properties such as software independence by merely being able to recount ballots. Remote voting instead, even with the introduction of blockchain technology, remains challenging and further research in methodologies to effectively protect the voting process from external interference on the voter's device is needed. Another aspect that must be kept in consideration is that, most likely, research behind attacks will go hand in hand with the one on securing electronic voting. Advances in cryptography, zero-day vulnerabilities and must always be kept in mind to design proper countermeasures or, in the unfortunate case of a new vulnerability discovered shortly before or during an election, consider having a fully functional backup plan.

Electronic voting will be a fundamental achievement in realizing a society where computer technologies are fully integrated into the services provided by a government but, in the current state, further research is needed to obtain a secure and private way to organize elections using electronic and computer mediums. Until a proven solution is designed, governments should avoid e-voting or a least make use of systems with a verifiable paper trail. Especially remote voting should be set aside for now, since the risks raised by

employing vulnerable solutions are significantly more impactful on the legitimacy of the elections when compared to the benefits given by being able to vote from any place. Considering smaller settings, where no specific threat is identified, remote e-voting solutions can make an impact in situations where it is difficult to organize a proper paper election and coercion can be considered low.

The concept of electronic voting, since its first steps in the second part of the twentieth century, has made huge advances forward but still requires additional time and research to be considered complete. At that point, an abandonment of paper ballots will be probable and governments will push for such a transition. At that moment in the future, it will be fundamental to have a proper solution to be able to fully assure the properties of voting and be accepted by the general public as a valid alternative to paper ballots.

Bibliography

[1] Y. Abuidris, R. Kumar, and W. Wenyong. A survey of blockchain based on e-voting systems. In *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*, pages 99–104, 2019.

[2] B. Adida. Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.

[3] B. Adida and R. L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 29–40, 2006.

[4] B. Ahn. Implementation and early adoption of an ethereum-based electronic voting system for the prevention of fraudulent voting. *Sustainability*, 14(5):2917, 2022.

[5] A. Al Sammak, A. Abd El Rahman, T. El Shishtawy, and A. Elewa. Challenges of electronic voting-a survey. *Advances in Computer Science: an International Journal*, 4(5), 2015.

[6] A. Alshehri, M. Baza, G. Srivastava, W. Rajeh, M. Alrowaily, and M. Almusali. Privacy-preserving e-voting system supporting score voting using blockchain. *Applied Sciences*, 13(2):1096, 2023.

[7] S. T. Alvi, M. N. Uddin, L. Islam, and S. Ahamed. Dvtchain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system. *Journal of King Saud University-Computer and Information Sciences*, 34(9):6855–6871, 2022.

[8] A. B. Ayed. A conceptual secure blockchain-based electronic voting system. *International Journal of Network Security & Its Applications*, 9(3):01–09, 2017.

[9] S. Bell, J. Benaloh, M. D. Byrne, D. DeBeauvoir, B. Eakin, P. Kortum, N. McBurnett, O. Pereira, P. B. Stark, D. S. Wallach, et al. Star-vote: A secure, transparent, auditable, and reliable voting system. In *2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13)*, 2013.

[10] J. Benaloh. Dense probabilistic encryption. In *Proceedings of the workshop on selected areas of cryptography*, pages 120–128, 1994.

[11] J. Benaloh and E. Lazarus. The trash attack: An attack on verifiable voting systems and a simple mitigation. Technical report, Technical Report MSR-TR-2011-115, Microsoft, 2011.

[12] S. Bistarelli, M. Mantilacci, P. Santancini, and F. Santini. An end-to-end voting-system based on bitcoin. In *Proceedings of the Symposium on Applied Computing*, pages 1836–1841, 2017.

[13] J. Brunet, A. D. Pananos, and A. Essex. Review your choices: When confirmation pages break ballot secrecy in online elections. In *Electronic Voting: 7th International Joint Conference, E-Vote-ID 2022, Bregenz, Austria, October 4–7, 2022, Proceedings*, pages 36–52. Springer, 2022.

[14] M. Chaieb and S. Yousfi. Loki vote: A blockchain-based coercion resistant e-voting protocol. In *Information Systems: 17th European, Mediterranean, and Middle Eastern Conference, EMCIS 2020, Dubai, United Arab Emirates, November 25–26, 2020, Proceedings 17*, pages 151–168. Springer, 2020.

[15] M. Chaieb, S. Yousfi, P. Lafourcade, and R. Robbana. Verify-your-vote: A verifiable blockchain-based online voting protocol. In *Information Systems: 15th European, Mediterranean, and Middle Eastern Conference, EMCIS 2018, Limassol, Cyprus, October 4-5, 2018, Proceedings 15*, pages 16–30. Springer, 2019.

[16] D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*, pages 199–203. Springer, 1983.

[17] D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE security & privacy*, 2(1):38–47, 2004.

[18] D. Chaum and E. Van Heyst. Group signatures. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings 10*, pages 257–265. Springer, 1991.

[19] D. Chaum, R. T. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. Ryan, E. Shen, A. T. Sherman, and P. L. Vora. Scantegrity ii: End-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE transactions on information forensics and security*, 4(4):611–627, 2009.

[20] D. Chaum, R. T. Carback III, J. Clark, C. Liu, M. Nejadgholi, B. Preneel, A. T.

Sherman, M. Yaksetig, and F. Zagorski. Votexx: A remote voting system that is coercion resistant. *UMBC Student Collection*, 2020.

[21] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[22] J. Clark and U. Hengartner. Selections: Internet voting with over-the-shoulder coercion-resistance. In *Financial Cryptography and Data Security: 15th International Conference, FC 2011, Gros Islet, St. Lucia, February 28-March 4, 2011, Revised Selected Papers 15*, pages 47–61. Springer, 2012.

[23] M. R. Clarkson, S. Chong, and A. C. Myers. Civitas: Toward a secure voting system. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 354–368. IEEE, 2008.

[24] V. Cortier, P. Gaudry, and S. Glondu. Belenios: a simple private and verifiable electronic voting system. *Foundations of Security, Protocols, and Equational Reasoning: Essays Dedicated to Catherine A. Meadows*, pages 214–238, 2019.

[25] C. Denis González, D. Frias Mena, A. Massó Muñoz, O. Rojas, and G. Sosa-Gómez. Electronic voting system using an enterprise blockchain. *Applied Sciences*, 12(2):531, 2022.

[26] T. Dimitriou. Efficient, coercion-free and universally verifiable blockchain-based voting. *Computer Networks*, 174:107234, 2020.

[27] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[28] S. Estehghari and Y. Desmedt. Exploiting the client vulnerabilities in internet e-voting systems: Hacking helios 2.0 as an example. *EVT/WOTE*, 10:1–9, 2010.

[29] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto*, volume 86, pages 186–194. Springer, 1986.

[30] S. Gao, D. Zheng, R. Guo, C. Jing, and C. Hu. An anti-quantum e-voting protocol in blockchain with audit function. *IEEE Access*, 7:115304–115316, 2019.

[31] D. Golnarian, K. Saedi, and B. Bahrak. A decentralized and trustless e-voting system based on blockchain technology. In *2022 27th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–7. IEEE, 2022.

[32] P. Griffiths. Watchdog says "high risk" e-voting should be

halted, Aug 2007. URL `https://www.reuters.com/article/uk-britain-election-internet-idUKL0216521920070802`.

[33] D. A. Gritzalis. Principles and requirements for a secure e-voting system. *Computers & Security*, 21(6):539–556, 2002.

[34] J. A. Halderman and V. Teague. The new south wales ivote system: Security failures and verification flaws in a live online election. In *E-Voting and Identity: 5th International Conference, VoteID 2015, Bern, Switzerland, September 2-4, 2015, Proceedings 5*, pages 35–53. Springer, 2015.

[35] R. Hanifatunnisa and B. Rahardjo. Blockchain based e-voting recording system design. In *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, pages 1–6. IEEE, 2017.

[36] F. S. Hardwick, A. Gioulis, R. N. Akram, and K. Markantonakis. E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1561–1567. IEEE, 2018.

[37] F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa, and G. Hjálmtỳsson. Blockchain-based e-voting system. In *2018 IEEE 11th international conference on cloud computing (CLOUD)*, pages 983–986. IEEE, 2018.

[38] J.-H. Hsiao, R. Tso, C.-M. Chen, and M.-E. Wu. Decentralized e-voting systems based on the blockchain technology. In *Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 17*, pages 305–309. Springer, 2018.

[39] J. Huang, D. He, M. S. Obaidat, P. Vijayakumar, M. Luo, and K.-K. R. Choo. The application of the blockchain technology in voting systems: A review. *ACM Computing Surveys (CSUR)*, 54(3):1–28, 2021.

[40] N. Huber, R. Küsters, T. Krips, J. Liedtke, J. Müller, D. Rausch, P. Reisert, and A. Vogt. Kryvos: Publicly tally-hiding verifiable e-voting. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1443–1457, 2022.

[41] U. Jafar, M. J. A. Aziz, and Z. Shukur. Blockchain for electronic voting system—review and open research challenges. *Sensors*, 21(17):5874, 2021.

[42] Japantimes.jp. Tsukuba first in japan to deploy online voting system,

Sep 2018. URL https://www.japantimes.co.jp/news/2018/09/02/national/politics-diplomacy/new-online-voting-system-introduced-city-tsukuba/.

[43] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, pages 61–70, 2005.

[44] A. Kerckhoffs. La cryptographie militaire. *J. Sci. Militaires*, 9(4):5–38, 1883.

[45] C. F. Kerry and P. D. Gallagher. Digital signature standard (dss). *FIPS PUB*, pages 186–4, 2013.

[46] K. M. Khan, J. Arshad, and M. M. Khan. Secure digital voting system based on blockchain technology. *International Journal of Electronic Government Research (IJEGR)*, 14(1):53–62, 2018.

[47] K. M. Khan, J. Arshad, and M. M. Khan. Simulation of transaction malleability attack for blockchain-based e-voting. *Computers & Electrical Engineering*, 83:106583, 2020.

[48] Y.-X. Kho, S.-H. Heng, and J.-J. Chin. A review of cryptographic electronic voting. *Symmetry*, 14(5):858, 2022.

[49] A. Kiayias, L. Michel, A. Russell, N. Shashidhar, A. See, A. Shvartsman, and S. Davtyan. Tampering with special purpose trusted computing devices: A case study in optical scan e-voting. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, pages 30–39. IEEE, 2007.

[50] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 27–40. IEEE, 2004.

[51] R. Kusters, T. Truderung, and A. Vogt. Clash attacks on the verifiability of e-voting systems. In *2012 IEEE Symposium on Security and Privacy*, pages 395–409. IEEE, 2012.

[52] R. Küsters, J. Liedtke, J. Müller, D. Rausch, and A. Vogt. Ordinos: a verifiable tally-hiding e-voting system. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 216–235. IEEE, 2020.

[53] M. Le Pennetier, L. Thomas, and J. Stonestreet. France drops electronic voting for citizens abroad over cybersecurity fears, Mar 2017. URL https://www.reuters.com/article/us-france-election-cyber-idUSKBN16D233.

[54] Ledgerinsights.com. Russia trials blockchain voting in controversial parliamentary elections, Sep 2021. URL `https://www.ledgerinsights.com/russia-trials-blockchain-voting-in-controversial-parliamentary-elections/`.

[55] K. Lee, J. I. James, T. G. Ejeta, and H. J. Kim. Electronic voting service using block-chain. *Journal of Digital Forensics, Security and Law*, 11(2):8, 2016.

[56] Y. Liu and Q. Wang. An e-voting protocol based on blockchain. *Cryptology ePrint Archive*, 2017.

[57] W. Lueks, I. Querejeta-Azurmendi, and C. Troncoso. Voteagain: a scalable coercion-resistant voting system. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1553–1570, 2020.

[58] S. McCombie, A. J. Uhlmann, and S. Morrison. The us 2016 presidential election & russia's troll farms. *Intelligence and National Security*, 35(1):95–114, 2020.

[59] P. McCorry, S. F. Shahandashti, and F. Hao. A smart contract for boardroom voting with maximum voter privacy. In *Financial Cryptography and Data Security: 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers 21*, pages 357–375. Springer, 2017.

[60] C. Mote Jr. Report of the national workshop on internet voting: issues and research agenda. In *ACM International Conference Proceeding Series*, volume 128, pages 1–59, 2000.

[61] M. F. Mursi, G. M. Assassa, A. Abdelhafez, and K. M. A. Samra. On the development of electronic voting: a survey. *International Journal of Computer Applications*, 61 (16), 2013.

[62] S. Nakamoto. Bitcoin whitepaper. *URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019)*, 2008.

[63] C. of Europe. Recommendation cm/rec(2017)51 of the committee of ministers to member states on standards for e-voting, 2017.

[64] Y. Oren and A. Wool. Attacks on {RFID}-based electronic voting systems. *Cryptology ePrint Archive*, 2009.

[65] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18*, pages 223–238. Springer, 1999.

[66] S. Panja and B. Roy. A secure end-to-end verifiable e-voting system using blockchain and cloud server. *Journal of Information Security and Applications*, 59:102815, 2021.

[67] R. L. Rivest. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[68] R. L. Rivest. The threeballot voting system. 2006.

[69] R. L. Rivest. On the notion of 'software independence'in voting systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3759–3767, 2008.

[70] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*, pages 552–565. Springer, 2001.

[71] P. Y. Ryan, D. Bismark, J. Heather, S. Schneider, and Z. Xia. Prêt à voter: a voter-verifiable voting system. *IEEE transactions on information forensics and security*, 4 (4):662–673, 2009.

[72] P. Y. Ryan, P. B. Rønne, and V. Iovino. Selene: Voting with transparent verifiability and coercion-mitigation. In *Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers 20*, pages 176–192. Springer, 2016.

[73] D. Sandler, K. Derr, and D. S. Wallach. Votebox: A tamper-evident, verifiable electronic voting system. In *USENIX Security Symposium*, volume 4, page 87, 2008.

[74] A. Schneider, C. Meter, and P. Hagemeister. Survey on remote electronic voting. *arXiv preprint arXiv:1702.02798*, 2017.

[75] B. Shahzad and J. Crowcroft. Trustworthy electronic voting using adjusted blockchain technology. *IEEE Access*, 7:24477–24488, 2019.

[76] M. A. Specter, J. Koppel, and D. Weitzner. The ballot is busted before the blockchain: A security analysis of voatz, the first internet voting application used in us federal elections. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1535–1552, 2020.

[77] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman. Security analysis of the estonian internet voting system. In *Pro-*

ceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pages 703–715, 2014.

[78] Swissinfo.ch. Switzerland's first municipal blockchain vote hailed a success, Jul 2018. URL `https://www.swissinfo.ch/eng/crypto-valley-_-switzerland-s-first-municipal-blockchain-vote-hailed-a-success/44230928`.

[79] R. Taş and Ö. Ö. Tanrıöver. A systematic review of challenges and opportunities of blockchain for e-voting. *Symmetry*, 12(8):1328, 2020.

[80] L. Vo-Cao-Thuy, K. Cao-Minh, C. Dang-Le-Bao, and T. A. Nguyen. Votereum: An ethereum-based e-voting system. In *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE, 2019.

[81] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu. Large-scale election based on blockchain. *Procedia Computer Science*, 129:234–237, 2018.

[82] K.-H. Wang, S. K. Mondal, K. Chan, and X. Xie. A review of contemporary e-voting: Requirements, technology, systems and usability. *Data Science and Pattern Recognition*, 1(1):31–47, 2017.

[83] R. Wen and R. Buckland. Masked ballot voting for receipt-free online elections. In *E-Voting and Identity: Second International Conference, VOTE-ID 2009, Luxembourg, September 7-8, 2009. Proceedings 2*, pages 18–36. Springer, 2009.

[84] S. Wolchok, E. Wustrow, J. A. Halderman, H. K. Prasad, A. Kankipati, S. K. Sakhamuri, V. Yagati, and R. Gonggrijp. Security analysis of india's electronic voting machines. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 1–14, 2010.

[85] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman. Attacking the washington, dc internet voting system. In *Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers 16*, pages 114–128. Springer, 2012.

[86] E. Yavuz, A. K. Koç, U. C. Çabuk, and G. Dalkılıç. Towards secure e-voting using ethereum blockchain. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–7. IEEE, 2018.

[87] H. Yi. Securing e-voting based on blockchain in p2p network. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–9, 2019.

[88] S. Zhang, L. Wang, and H. Xiong. Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. *International Journal of Information Security*, 19:323–341, 2020.

[89] W. Zhang, Y. Yuan, Y. Hu, S. Huang, S. Cao, A. Chopra, and S. Huang. A privacy-preserving voting protocol on blockchain. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 401–408. IEEE, 2018.

[90] H. Zhu, L. Feng, J. Luo, Y. Sun, B. Yu, and S. Yao. Bcvotemde: A blockchain-based e-voting scheme for multi-district elections. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 950–955. IEEE, 2022.

List of Figures

1.1 Scheme for e-voting fundamental properties. 6
1.2 Scheme for e-voting secondary properties. 8

2.1 General scheme for a poll voting system with possible vulnerabilities 14
2.2 General scheme for a remote voting system with possible vulnerabilities . . 19

List of Tables

2.1 Comparison between voter verifiable systems 18
2.2 Comparison between remote voter verifiable systems 25
2.3 Comparison between blockchain based systems 38
2.4 Comparison between blockchain based systems 39

3.1 Comparison between attacks against e-voting systems 53

Acknowledgements

First I would like to thank my advisor Stefano Longari and co-advisor Mario D'Onghia for guiding me in drafting this thesis and for the valuable advice they have given during this process.
Finally, a warm thank you to my family and friends who have always been there encouraging me to give my best.