Study on Poll-Site Voting and Verification Systems

Roger Jardí-Cedó, Jordi Pujol-Ahulló, Jordi Castellà-Roca, Alexandre Viejo

Universitat Rovira i Virgili, Departament d'Enginyeria en Informàtica i Matemàtiques,
UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Spain
E-mail: {roger.jardi, jordi.pujol, jordi.castella, alexandre.viejo@}@urv.cat

Abstract

Voting is an important part of the democratic process. The electorate makes a decision or expresses an opinion that is accepted by everyone. However, some individual or group may be interested in tampering with the elections process to force an outcome in their favor. Hence, controlling the whole voting process to ensure that it is performed correctly and according to current rules and law is, then, even more important. In this work, we present a review of existing verification systems for paper-based and electronic voting systems in supervised environments, from both academic and commercial worlds. To do so, we perform a fair comparison of a set of representative voting verification systems using an evaluation framework. We define this framework to be composed of several properties, covering important system areas, ranging from user interaction to security issues. Then, we model the natural evolution of verifiability issues on notable voting systems from academia and commerce which are influenced by restrictions on current laws and by the advance of technology.

Keywords: Verification, Voting Systems, Electronic Voting Systems.

1. Introduction

From the birth of democracy in Athens in 6th Century BC and the first form of electoral laws, electoral systems have been designed and developed according to variations in practice of democratic governments worldwide.

The elections process consists in choosing a person or party, a *candidate*, to represent all the members of a community (*e.g.,* a company, a state or a country). For a candidate, winning the elections carries a big responsibility in terms of representation, but it is also very attractive for some other reasons (*e.g.,* funds, ability of changing existing rules and laws). Therefore, there might be some individuals interested in diverting elections' results and easing the victory of a certain candidate.

However, it is a difficult task to check whether the elections' results correspond to the voters' preferences while ensuring voter secrecy and anonymity. In other words, *elections must be verifiable* and, the vote must be secret and not linked to the voter. For instance, if voter Alice votes for candidate Bob, any another person must not be able to deduce Alice's preferences from the elections process and results, but instead, anyone must be able to verify the correctness of the whole voting process. Therefore, *verifiability* becomes one of the most important elections' attribute to provide *trustworthiness* in the elections' results to both candidates and voters.

Verifying that elections' results correspond to voters' preferences depends on the voting system. From the location point of view, most of the existing systems are based on poll sites, where voters attend specific places in order to vote. Remote voting systems (such as mail or Internet voting systems) are also an alternative.

From the ballot perspective, traditional voting systems use ballots in paper format with standardized list of candidates. They were first introduced in the state of Victoria, Australia, in 1856 [8]. Paper ballots contain all the necessary information to vote for a specific candidate, in a human-readable format. Thus, in the vote counting or *tally*, any person can verify whether the ballot is correct and, if so, to which candidate it relates. However, the main drawbacks of traditional voting systems are that all operations are manual and their economic and logistic costs are elevated. Moreover, the tally process where votes are counted can turn in a long procedure susceptible to human errors, especially when the voting system is complex.

More modern voting solutions incorporate electronic devices to mainly accelerate the tally process and overcome the problems introduced by human errors [7], and also improve accessibility for disabled and illiterate voters. First initiatives appeared in 1964 in some states of the USA, which used punchcards and computer tally machines [8]. Broadly speaking, this kind of solutions can use different technologies, ranging from punchcards, optical scanners (to scan ballots), cryptographic techniques and direct-recording electronic (DRE) voting terminals.

Electronic voting (e-voting) systems effectively reduce the cost of traditional approaches, nevertheless, they also pose other kinds of challenges to elections verifiability. In this way, the work presented in [37] analyzes some relevant attacks which can be applied to e-voting infrastructures and also who could perform them. That information is summarized in Table 1.

These attacks may compromise the verifiability of a system. For example, let us assume that Alice scans her ballot in an optical-scanner-based (*opscan*) voting system. Let us also consider that a poll worker with enough access rights discards Alice's scanned ballot without informing her. After elections, if no proof of that scanning was provided to Alice, she or any other independent observer, could not be sure whether her electronic ballot has been eliminated or modified after her ballot casting.

In addition to verifiability issues, security holes in the technology used to implement an e-voting infrastructure may also jeopardize the *voter anonymity*. Note that, a system that allows a certain individual to link a vote with the voter opens the door to *coercibility attacks* (*i.e.,* a voter might be coerced into voting for a particular candidate). As a conclusion, ideal e-voting schemes should consider these issues in order to provide proper *verifiability*, ensure *voter anonymity*

Table 1: Summary of some relevant attacks on e-voting systems.

	Voter with forged identifier	Poll worker with access to storage media	Voting device developer
Vote multiple times	●		
Access administrative functions	●	●	
Modify system configuration		●	●
Modify ballot definition (*e.g.;* party affiliation)		●	●
Cause votes to be miscounted by tampering with configuration		●	●
Impersonate legitimate voting machine to tallying authority		●	●
Create, delete, and modify votes		●	●
Link voters with their votes		●	●
Tamper with audit logs		●	●
Insert backdoors into code			●

and reduce the *costs* in comparison with the traditional voting approaches.

Despite these additional challenges and problems, the trend is clear and firm towards using electronic voting means [28], in particular, not only electronic tally, but also electronic vote casting [7]. On the one hand, this fact means that there are more verification challenges as the voting system becomes more computationally complex. On the other hand, this kind of e-voting system may be significantly helpful for disabled and illiterate citizens. At the same time, the use of electronic voting technologies may reduce the economic and logistic costs of elections and consultations, while enabling geographically distributed citizens to vote.

Therefore, the *verifiability of the voting system* becomes essential for trustworthy elections. This *capability* is commonly considered under three different points of view, which lead to *individual*, *universal* and *end-to-end* types of verifications. Briefly speaking, *individual verification* allows voters to check that their individual ballots are correctly cast and counted. From the system point of view, *universal verification* allows voters, electoral and third parties to inspect that the elections' results correspond to cast ballots. The aim is to ensure that the whole voting process is performed correctly, which, in turn, leads to *trustworthy* elections' results. In traditional voting systems, both verifications can be achieved by a set of *procedures* (*i.e.,* manual operations addressed by elections officials, or also by independent entities and observers from candidates). On the other hand, in e-voting systems this is achieved by a mix of procedures and mainly, *technologies.* A later enhanced property is the *end-to-end* (E2E) *verifiability.* From the voter point of view, in an E2E verifiable voting system, a voter can check that her ballot is correctly cast and counted in the final tally. The goal is to increase the voters' *confidence* in the elections' results. Note that this property was hardly supportable in traditional voting systems, since the voter Alice concluded her interaction with the voting system when casting the ballot in the ballot box. However, new designs of voting systems and modern technologies facilitate an E2E voter verifiable voting process.

This survey presents *a fair comparison of the verifiability of 16 complete voting systems and 2 partial solutions which can be divided into two main categories: paper-based and electronic-based.* They are also named as *voting verification systems* (VVSs). The motivation behind this decision is that, nowadays, poll-site-based voting systems are the most common ones.

In this paper, the U.S. HAVA guidelines are used to perform a preliminary classification of all the analyzed VVSs. This step groups the different schemes according to similar fundamental features and allows their fair comparison. HAVA (*Help America Vote Act*) is a United States federal law [1] that pursues three main goals: (i) replace punchcard and lever-based voting systems; (ii) create the Election Assistance Commission to assist in the administration of Federal elections; and (iii) establish minimum elections administration standards. The HAVA classification requires VVSs to provide proofs that allow voters and other observers to verify that the voting process has not been tampered with. Therefore, e-voting schemes not providing this kind of proofs are discarded and not addressed in this paper.

Certain voting schemes used in some emerging countries [42] are examples of this last situation. These approaches are based on the use of DREs and the integrity and confidentiality of their voting processes uniquely depend on the security of the electronic voting terminals themselves and the trustworthiness of the elections officers. This also includes the trustworthiness of the certifications applied on the DREs.

These measures are insufficient to comply with the U.S. HAVA guidelines and, hence, they are not considered in this survey. Nevertheless, due to its scale and impact, it is worth to mention the e-voting scheme used in Brazil. In 2000, this country completed the first completely automated elections using DREs (electronic voting terminals) [62]. As explained above, the integrity and security of the whole voting process depend on the integrity of the DREs and the electoral officers who manage them. Even though the provision of printed receipts to the voters was initially considered to be used in elections scheduled after 2003, it was finally discarded in favor of digitally recording the votes and storing them in the DREs (being only accessible to the electoral officials). Other security measures provided by the electoral authorities focuses on showing that the DREs count the votes properly. Nevertheless, these measures are based on monitoring a subset of DREs leaving the rest of voting terminals unsupervised [15] and, hence, susceptible of being tampered with. In addition to that, external observers are not allowed to check the integrity of the software used in the DREs [61]. As a result, some experts have expressed their concerns about the security properties provided by this particular e-voting scheme [61, 16].

The contribution of this work is threefold:

1. Definition of a common evaluation framework (including 15 VVS characteristics) to fairly compare all systems.
2. Study and comparison of 18 notable voting systems.
3. Analysis of current and future trends in voting schemes and technology.

Document structure. The next section introduces the necessary background for the present work. Sec. 3 presents the evaluation framework. We then present a selection of notable paper-based voting verification systems (VVSs) (Sec. 4) and their analysis (Sec. 6.1). In the same way, highlighting electronic VVSs are introduced (Sec. 5) and analyzed (Sec. 6.2). The following main point presented in Sec. 6.3 is the analysis of observed trends. Finally, Sec. 7 presents the concluding remarks of this work.

2. Background

In this study, we consider the standard voting process composed of the following phases: (i) *voter registration* and identification, (ii) *vote casting* using ballots and (iii) *vote tally*, where all ballots are securely *tabulated* and unbiased results are made *publicly* available. The voting process also includes all *procedures* and *technologies* to reliably address the consultations or elections. Figure 1 shows a diagram of this standard process which includes some internal procedures.

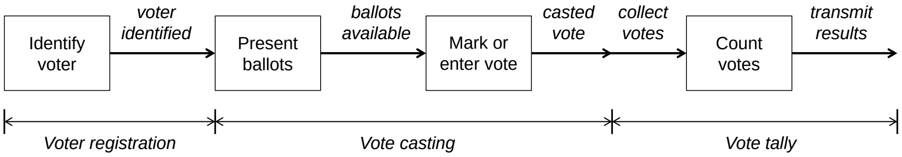

Figure 1: Standard voting process.

In the following section we present the system classification of the voting models and voting verification systems, according to the voting location and the U.S. HAVA classification. Note that, this classification is detailed in Sec. 2.1.2 and it will be used later in this work to organize the analyzed voting systems. Additionally, we briefly summarize the existing electronic voting paradigms in Sec. 2.2.

2.1. Voting models

We present two classifications of the voting models, according to the place where voters have to attend for voting (see Sec. 2.1.1), as well as according to the U.S. HAVA classification (see Sec. 2.1.2).

2.1.1. Location-based classification

According to the place where voters have to go to vote, voting systems are broadly classified into **poll-site-based** and **remote** voting systems. In the *former type*, voters go to a specific building, namely poll site. Nowadays it is the most widely used voting scheme. Alternatively, voters may remotely cast their vote in *remote voting systems*. These systems can be further classified as follows: **vote-by-mail**, **Internet**, **E-Mail voting**, **SMS voting** and **supervised remote**.

- **Vote-by-mail** was introduced in 1896 [26] and it is cheaper [60] than traditional voting systems. However, voting by mail suffers from vote loss or late delivery [7, 75]. To overcome these drawbacks, the next remote voting schemes appeared.
- **Internet voting** allows an electronic cast and tally, where the Estonian case [31] was the first worldwide Internet-based nationwide binding elections.
- **E-Mail voting** has been proposed as a voting model for citizens living abroad in some countries and under certain circumstances. For example, this system was used in the 2004 U.S. presidential and congressional elections by U.S. soldiers deployed in Iraq. This scheme has been criticized by the security problems related to electronic email environments (*e.g.,* vote manipulation while in transit and lack of privacy among others) [45].
- **SMS voting** (*i.e.,* mobile phone *short message service*) was used in Switzerland as a part of a series of trials in several regions of the country to introduce e-voting nationwide [33].
- **Supervised remote** voting allows *abroad* poll sites to cast votes, which are *electronically* gathered in the corresponding country (or county) for tallying. They are very helpful when voters are abroad (*e.g.,* the military), whilst reducing the tally time.

As mentioned earlier, our *focus* is to put on *verification systems of poll-site-based systems*, which also allows us to take supervised remote voting systems into consideration.

2.1.2. HAVA classification

This classification has been promulgated by the Election Assistance Commission (EAC), an independent agency of the United States government created by the Help America Vote Act of 2002 (HAVA). The 2005 Voluntary Voting System Guidelines [30, Volume 1, Appendix C] lists the VVSs in four types: (i) **Process separation-based** VVSs have a modular architecture split into two independent, totally isolated systems dealing the *generation and casting process*, respectively; (ii) **Evidence-based** VVSs are based on capturing all actions performed during the voting phase of voters; (iii) **Direct** VVSs generate a parallel registry of votes, which permits a direct verification of the vote to be cast; Lastly, (iv) **end-to-end cryptography-based** VVSs employ cryptographic methods to craft receipts that allow voters to verify that their votes were not modified, without revealing the voting preferences of voters. We classify the evaluated electronic VVSs according to this classification.

2.2. Electronic Voting Paradigms

Electronic voting (e-voting) systems are characterized by containing some procedure from within the voting process that is made by electronic, computerized means. Broadly speaking, according to the technology used by the e-voting systems, they are usually classified into the following *e-voting paradigms*:

- **Blind signatures.** Blind signatures were introduced by Chaum [17]. These ones belong to a class of digital signatures that allow to sign data without revealing its contents. In e-voting, “a ballot is blinded in order to achieve its confidentiality requirement. A voter is required to get the signature of a validator when he vote” [36].

- **Commitments.** Commitment schemes were formally defined by Brassard et al. [13]. By using commitments, a protocol player (*e.g.,* voter) chooses a value from a (finite) set and commits her choice (*e.g.,* an electoral candidate). This choice cannot be changed and must not be revealed. The player, though, may choose to reveal the value (anonymously) at some later time. In an e-voting system, Pedersen commitments are commonly used [55] because they provide information-theoretic privacy or perfectly hiding, also called *everlasting privacy* [5, 43].

- **Homomorphic cryptography.** E-voting schemes using homomorphic cryptosystems [25, 54] encrypt ballots so that when ciphered ballots are operated among them, their result is a cryptogram with the accumulated votes from all voters. This scheme is very efficient for the tally phase, since only few decrypting operations are necessary to obtain the elections’ results, while maintaining the voter anonymity and ballot privacy during the whole process.

- **Mix-nets.** A mix-net [23] in an e-voting system provides an anonymous channel shuffling the casted votes and preventing the correlation of their order. It is implemented with a set of mixing servers. Each server receives the votes, permutes their order, transforms the votes (typically re-encrypts or decrypts the votes) and finally, sends the votes to the next server. In the reencryption servers, the transformation is the encryption of each vote.

There is an encryption layer for each re-encryption server. In contrast, in the decryption servers, the vote is encrypted as many times as the number of mixing servers. These layers are removed when votes go through the servers. In both cases, it is hard (not possible nowadays) to correlate any output with its input. Once the votes have crossed the last mixing server, these have been disassociated from their voters. A fully robust and practical implementation can cause a less efficient tallying process than a tally based on homomorphisms [57, 56].

Some of these technologies comprise, as part of the protocol as a whole, the performance of some test to verify that the information managed remains unaltered (*e.g.,* the voter's vote was cast and counted as intended), without revealing the information in itself. In order to achieve this purpose zero-knowledge proofs (ZKPs) are used. They may differ in technology according to the cryptographic technique in use, even though they always provide [34]:

- **Completeness.** If the test is true, an honest verifier will be convinced by a honest prover.
- **Soundness.** If the test is false, a cheating verifier will convince the honest prover only at a small probability.
- **Zero-knowledge.** If the test is true, a cheating verifier only learns this mere fact, nothing about its content.

3. Common Evaluation Framework

In this section, we introduce the classification of the properties that we extract from the set of systems under consideration. All of them constitute the single, structured *evaluation framework* that we use to ease their comparison and analysis.

3.1. Classification of VVSs

We employ the following classification to *filter* the systems in order to obtain their natural organization (see Fig. 2). The year of publication of the academic publication or system is the last organizational property used.

- We distinguish between **paper-** and **electronic-based** VVSs. The former class requires voting by **paper** ballots, whilst the latter needs *electronic* votes to correctly proceed with the voting process.
- We use the aforementioned **HAVA classification** to distinguish from *process separation-*, *evidence-*, *end-to-end (E2E) cryptography-based* and *direct* VVSs.
- We organize them into **integral** or **independent** systems. Whereas *integral* ones perform the whole voting process, *independent* VVSs are designed solely to verify independently that another voting system is reliably working.

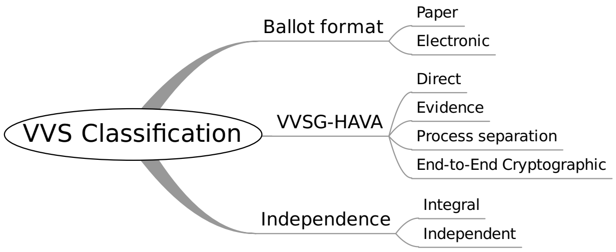

Figure 2: Classification of the voting verification systems.

According to the above categorization, we list the evaluated VVSs in Fig. 3.

3.2. Evaluated Properties from VVSs

In Fig. 4, we present the characteristics which are used to evaluate all the voting systems. We have classified them according to the following voting process concerns: *user interaction*, *security*, *integrability* (with an existing voting system), as well as *technical issues*. Note that any property definition is such that a *positive answer* corresponds to a *positive feature*.

User interaction. The *user interaction* greatly determines the voters' impression and usability of the voting system. We organize the user interaction analysis according to the following two properties:

1. **Accessibility.** Whether the system *does not* prevent a physically limited user from voting.
2. **Use impact.** Whether the system *does not* incorporate more complex or different phases compared to the standard voting process (see Sec. 2), which may alter the voter interaction. These changes might be more noticeable in the process to cast a vote.

Security. Security issues are mainly categorized into two broad sets, namely those related to the *voter* and to the *voting* process as a whole. In the first set, we consider in particular the voter verification (*i.e., individual verification*). In the second set, we consider the public verification (or *universal verification*). We also include a property describing whether a system can be audited. This is an important issue in e-voting systems (with no paper trail) in order to certify the final tally and the elections' results.

- **Voter-related:**

   3. **Ballot secrecy.** Whether the system *prevents* a third entity from seeing the content of the ballot.

      Everlasting secrecy or privacy. Security, more specifically the privacy, is provided in response to any future technological advance, given the current state of the art [5, 43].

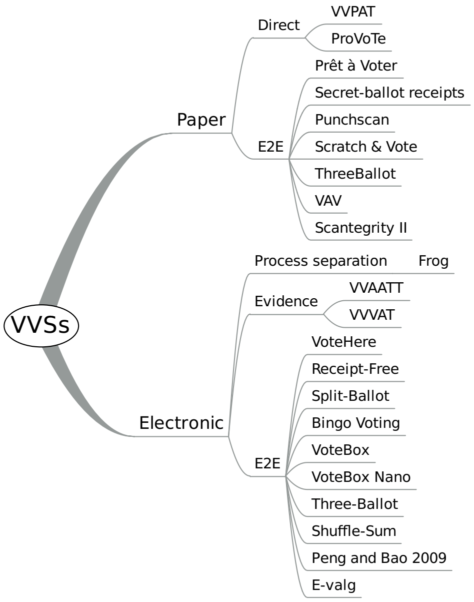

Figure 3: Summary of considered systems and their categorization. The finally evaluated systems appear in Figs. 5 and 6 for paper- and electronic-based VVS.

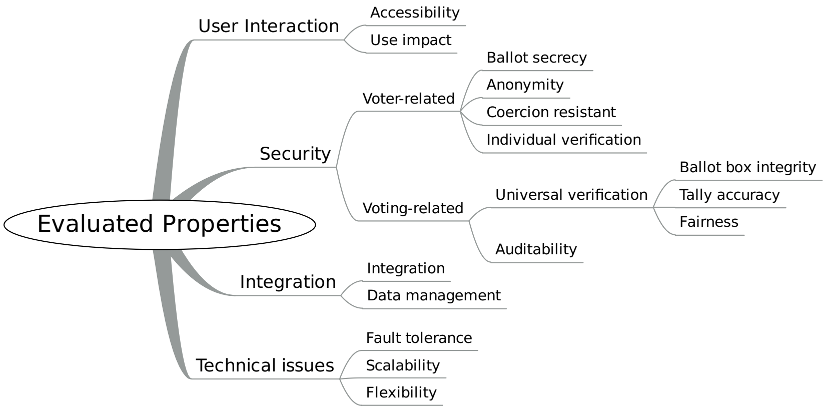

Figure 4: Summary of the evaluated VVS properties.

4. **Ballot anonymity.** Whether the system *prevents* ballots from being linked to voters.
5. **Coercion resistant.** This property refers to the fact that a coercer *cannot* verify how a voter has voted.

   In e-voting sytems, *coercibility* is the danger that outside of a public polling place, a voter could be coerced into voting for a particular candidate. Therefore, in supervised e-voting systems, wherein environment is controlled by the electoral authorities, coercion is limited to the possibility to prove the voter choice to the coercer. In schemes where vote receipts are given to voters, a coercer may use them to prove the voter choice. Besides, voting systems with a public bulletin board may open the door to certain "pattern voting" attacks which also allow coercers to obtain a proof of the voter choices.
6. **Individual verification.** A voter *can* verify that her vote was *cast-as-intended*.

- **Voting-related:**
  - **Universal verification:** A system is universally verifiable if anyone (whether a voter or not) can verify that votes were *counted-as-cast*. Universal verification is usually composed of two characteristics and we use them to assess systems' universal verifiability.
    7. **Ballot box integrity.** Only registered voters' votes *appear* at the end of the voting process (before the tally process). Votes must be unmodified. Besides, generally, only one vote from each registered voter must be allowed. Nevertheless, this last restriction depends on the purpose of the voting process.
    8. **Tally accuracy.** The tally process *counts* all the cast votes.
    9. **Fairness.** The voting system should ensure that no partial results become known prior to the end of the elections procedure. As stated in [65], fairness is an important concern as it may induce what is known as the "bangwagon effect", where a certain candidate gains momentum by winning a handful of districts and subsequently capitalizes on this win by either having more voters previously undecided turning to her side or having voters supportive of other candidates opting out from participating in the elections process.

10. **Auditability.** The e-voting system (with no paper trails) *allows* a third party to analyze what happened before, during, and after the vote was cast, without compromising other security properties, in order to certify the final tally and elections' results. Also a faulty implementation or incorrect procedures can still result in insecure elections. The auditing process will detect these issues. The party

responsible for verifying the correct development of the e-voting process is called *auditor* and it should be a team of individuals with knowledge about computer engineering and cryptography. This team is expected to act on behalf of the parties, the elections authority or, even, the voters.

Integration. Regardless of whether the VVSs are *integral* or *independent*, we consider the feasibility and effectiveness of adapting/interacting the evaluated system with other voting systems. In particular, we briefly consider the synchronization of operations, especially when votes are being cast, between a given voting system and the evaluated system *acting as an independent VVS* (as issued in Sherman et al. [80]).

11. **Integration.** *Ease* of implementing/adapting the evaluated system as an independent verifier system for other voting infrastructures.

12. **Data management.** Whether the vote cast subsystem of other voting systems and the evaluated system provides atomicity and/or data replication.

Technical issues. In this last category of properties, we analyze the performance of the considered VVS from a *technical viewpoint.* Broadly speaking, we will collect the structural, technological challenges imposed by the design of the given VVS.

13. **Fault tolerance.** A suitable voter *must be able* to cast her vote, whenever she can, along the established time. This implies that the voting system should be resilient to the faulty behaviour of up to a certain number of components or parts [65].

14. **Scalability.** The verifier system *scales* computationally.

15. **Flexibility.** It measures the level of freedom in the ballot size, format and type *allowed* by the verifier system (*e.g.,* number of candidates, write-in mode). For instance, a system providing the ability of introducing write-in choices, preferential and single choice voting presents a major flexibility degree. A system that does not offer write-in choices introduces a smaller flexibility. Therefore, a system presenting only single choice voting turns into an almost inflexible system.

Properties representation. For brevity, when summarizing these 15 properties for all the evaluated systems, we use the following notation:

4. Paper-based VVSs: Presentation and classification

In this section, we collect the most notable voting solutions based on paper in Fig. 5. The idea behind these VVSs is that they *require* paper ballots for voting and/or verification purposes. From the HAVA classification, we present two kinds of VVSs: *direct* and *E2E.*

Table 2: Value representation on the considered evaluation properties.

User interaction	↑ / ↓ / ∼: Good/Weak/Acceptable.
Security	**Y**/**N**/∼: **Y**es/**N**o/Partially.
Integration	**NT**: **N**o additional **T**echnical requirements (on voting consoles, etc). **T**: Additional **T**echnical requirements. **NSW**: **N**o additional **S**oft**W**are requirements (on voting consoles, etc). **SW**: Additional **S**oft**W**are requirements.
Data management	**NA**: There is **N**o operation **A**tomicity. **A**: There is operation **A**tomicity. **ROI**: Operations addressed by the verifier system are in **R**ead-**O**nly **I**nteraction mode. **RWI**: Operations addressed by the verifier system are in **R**ead-**W**rite **I**nteraction mode.
Technical issues	↑ / ↓: High/Low.
At any parameter	"N/A": When the property is not addressed.

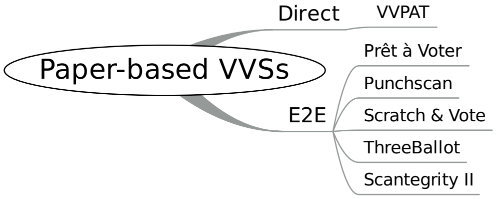

Figure 5: Summary of evaluated paper-based systems.

4.1. Direct VVSs

Direct VVSs allow individual and universal verification based on paper ballots. In this category, we analyze VVPAT [41].

4.1.1. Voter Verified Paper Audit Trail (VVPAT)

VVPAT is an independent verification system that relies on the use of DREs which print paper ballots with the voters' preferences. It is worth to mention that this scheme is not a complete voting system itself. Nevertheless, due to its relevance on the e-voting literature, it is covered in this survey.

Created by R. Mercuri [41], this technique is also known as *Mercuri Method*. Its goal is to construct an independent and verifiable paper ballot from the electronic one, *readable by anyone* without the use of any technology.

The phases are as follows: (i) voter makes her choices in the DRE; (ii) a physical paper ballot is printed from those choices and it is displayed under glass or clear plastic; (iii) voter checks whether the paper content matches her vote preferences (as in the DRE); (iv) voter finally accepts the paper ballot (concluding the voting process), and the system submits the paper ballot to the ballot box. Otherwise, the system discards the printed paper ballot and restarts the process. Note that, in VVPAT, at no point there is an opportunity for the voter to remove the paper record from the voting area. Therefore, all paper ballots serve as verification and audit trail of the elections.

Most recent initiatives (such as ProVotE [85]) are *formally designed and verified* to provide VVPAT-compatible solutions, resistant to security attacks and voluntary or involuntary errors in human procedures. The case of ProVotE, used in binding elections, is very interesting since it obeys the general recommendation of using *open software* (in particular, Linux O.S. and Java implementation for the voting and management routines) and generic hardware (touchscreen, general-purpose PC, uninterruptible power supply (UPS) and printer) to build the DREs and voting booths. In addition, it is sufficiently flexible to enable the use of different hardware and elections' rules [85].

4.2. End-to-End Verifiable VVSs

E2E VVSs presented in this section use the ballot (or a part of it) as a *receipt* which can be taken home to verify after the elections, that the voter's vote was correctly tallied. In this class of VVSs we study several striking systems: Punchscan [32], Prêt à Voter [67], Scratch & Vote [3], ThreeBallot [63] (together with VAV [64]) and Scantegrity II [19].

4.2.1. Prêt à Voter

Prêt à Voter, introduced by D. Chaum, P. Y. A. Ryan and S. Schneider in 2004 [22, 70, 67], is a paper-based integral voting system. Rather than building encrypted receipts during the voting phase, this system constructs pre-printed encrypted ballots. Prêt à Voter has two clearly different parts on the paper ballot in the form of a two-column table. The left column contains the list of candidates in an apparent random order. The right part is empty where the voter will indicate her preferences and where a *ballot identifier* also appears.

Actually, the most important part of the voting relies on this ballot identifier. This ballot identifier corresponds to the encryption of the randomized candidate order (with respect to a predefined candidate order). This encryption is formed as "onion skin layers" of *probabilistic public key encryptions*, based on a *threshold scheme*. The result of the concatenation of these "onions layers" is the final ballot identifier. This is used by the mixnet, which at each step decrypts partially each vote (removing a skin from the "onion"). However, Ryan et al. [70] also presented and compared a model using re-encrypting mixnets.

From the voter viewpoint, the voting process starts by taking a ballot, entering the voting booth and making the corresponding selections on the *right* part. Afterwards, the voter removes and destroys the left part (with the list of candidates), scans the remaining right part (which will be sent to the public bulletin board). The voter then may take home this part of the ballot as a receipt. Elections officials may insert a stamp on the receipt to prove its authenticity. This receipt, then, becomes a voting proof against elections authorities if this receipt does not appear in the public bulletin board.

From the system point of view, the public web of the bulletin board contains all the receipts. In the tally process, these receipts are decrypted by a set of elections officials (i.e., by a decrypting mixnet or re-encrypting mixnet with a final decryption phase [70]).

A similar approach was also used in [18], which applies a set of a re-encryption of votes as *Russian nesting dolls* (*i.e.,* the same idea than *onion skin layers*), to decrypt votes using a decrypting mixnet, all of this to provide highly probable ballot privacy and anonymity, respectively. Even though the encryption mechanism is very similar to Prêt à Voter, the system presents a two-layer translucent plastic ballot, which has a full meaning when overlapped, and is meaningless when separated. This two-layer structure is very similar to Punchscan though. Given the similarities among these systems, we omit the study of [18] in this work.

4.2.2. Punchscan

Punchscan was introduced by D.Chaum in 2006 [32], which consists of an integral solution for voting in a paper-based fashion. In this case, the ballot design and the related cryptography are the most important elements on the whole voting process. The ballot is constructed from two halves, one overlapping the other one. The upper half has some holes where we can see some symbols of the second half from below. In the upper half, there exists a list of candidates, each of which relates to a specific symbol. In the lower half, there exists an apparently *unordered* list of the candidates' symbols. The order of both candidates (from the upper half) and symbols (typed in the second half) is specified by permutations from a *canonical order*.

To vote for a candidate, the voter just uses a *dauber* to mark it through one of the holes. Several candidates can also be selected. One half is destroyed and the other one is scanned (by the voter) to collect the voter's preferences, and conclude with the voting phase. By taking away this receipt, the voter can later check its *receipt* in the public bulletin board, allowing for E2E verifiability. Otherwise, the voter can complain about it to the electoral authorities.

4.2.3. Scratch & Vote

B. Adida and R. Rivest presented Scratch & Vote in 2006 [3]. This system provides an integral paper-based voting system, which addresses most of the security issues found in previous works (such as Prêt à Voter or PunchScan). The innovation appears in the introduction of two new items in the paper-based ballot. A *2D-barcode* collects the order of the candidates list, conveniently encrypted using the Paillier cryptosystem [54]. This cryptosystem is also used to address the tally, by applying additive homomorphic encryptions. The other item is a scratch surface. It has two functions from a voter viewpoint. The former is to verify that the order of the list of candidate is correct and matches that shown under the scratch area. The second use is to leave it intact, so that the electoral official considers the ballot legal and valid to be cast.

This work presents two alternatives of a Scratch & Vote voting system, based on the Prêt à Voter and Punchscan systems. These alternatives present some slight differences from the original voting procedures. Firstly, as in the corresponding systems, half of the ballot must be destroyed before casting the other half part. Secondly, the remaining half with an *intact* scratch surface must be always presented to the electoral official. This official will then remove the scratch part from the ballot and will read the rest of the ballot with an authorized opscan device. These *ballots* are made public on a bulletin board and, once scanned, they are taken-away as *receipts*.

4.2.4. ThreeBallot

ThreeBallot [63, 35] is an integral solution introduced by R. Rivest in 2006 for a paper-based voting scheme. The goal of the approach is providing cryptographic properties (mainly focused on the E2E verifiability), but without using cryptographic technologies.

As its name depicts, the idea behind this method is to employ three identical single ballots (namely *multi-ballot*) when a voter votes, with certain restrictions in the way they are filled in. In particular, the lemma of ThreeBallot is "vote-by-rows but cast-by-columns". Each single ballot has a candidate per row, appearing in the same standard order in all single ballots. After indicating the voting preferences in the multi-ballot, the voter casts each of the three single ballots separately in the ballot box.

To elaborate on ThreeBallot, the voting process can be described in four different phases:

1. **Vote elaboration.** The voter selects three single ballots (they can also appear in an individual multi-ballot paper format), each of which has an individual, unique *ballot identifier*. To vote **for** (resp. **against**) a candidate, the voter must mark exactly 2 bubbles in 2 bullets (resp. 1 bubble in a bullet) in its *row*. The marks can appear randomly throughout the multi-ballot, forming a certain *vote pattern*, but always ensuring the aforementioned marking restriction.

2. **Vote checking.** Once the voter indicated her voting preferences in the multi-ballot, it is inserted into a *checker* machine, that *accepts* (resp. denies) the multi-ballot configuration, as long as the marks pass (resp. fail) the ThreeBallot restrictions. If the multi-ballot passes, the voter takes home a copy of *one of the three* single ballots as her vote *receipt*.

3. **Vote casting.** To conclude, the voter casts her vote by introducing *separately* the three single ballots in the ballot box.

4. **Vote tallying.** Once the voting period concludes, the tallying process starts. Accounting the real number of votes for each candidate is as easy as performing the subtraction $N - V$, where N is the number of actual votes, and V the number of participating voters.

Alternatively to ThreeBallot, but very similar in essence, Rivest and Smith [64] present VAV. VAV is a three-ballot voting scheme with two single **V**ote ballots, and an **A**nti-vote ballot. Having all three ballots with a valid vote setting, voters must have a ballot **V** and an **A** identical (so that they cancel each other), while the remaining **V** contains the voter's preferences. The utility of this scheme is that it is extensible to other voting formats unavailable for ThreeBallot (such as Condorcet or Borda). The authors also present the concept of "floating receipts". When a voting system employs *floating receipts*, voters take home the *receipt of another voter*. Therefore, this is a mechanism to increase the voter's security within the voting scheme. The idea behind that is to use a bin where to toss one's receipt and take another's receipt randomly. This is useful since it increases the anonymity of the receipts, which cannot be used (even voluntarily) to prove one's vote preferences. However, because of the brevity of their presentation [64], we only analyze ThreeBallot in this work and consider *floating receipts* in the analysis part.

4.2.5. Scantegrity II

Since Scantegrity II [19] is a VVS evolved by D. Chaum, P. Y. A. Ryan, R. Rivest, J. Clark and E. Shen of Scantegrity I [20], we elaborate on the former for brevity.

Scantegrity II stands for "Scantegrity Invisible Ink". This is an *independent* VVS for existing opscan-based voting systems. However, its application requires changes in the software and in the information managed by the voting procedure. This information consists of a set of four tables (some of them are made public), which provides permutation and randomization to unlink voting preferences of casted ballots (*i.e., mixing*). The aim is to provide *ballot privacy* and *ballot anonymity*.

The particularity of Scantegrity II is the use of a special, decoder pen and invisible ink. More precisely, the pen's ink reacts when marking the area with invisible ink, so that the verification code comes to light. The voter then types this code into the receipt area. After some minutes, the verification code disappears. Since each ballot is identified uniquely into the elections process, the receipt area (with both the ballot id and the manually written verification code) provides *individual E2E verification*. A part of this receipt is removed by electoral officials as a proof of the cast ballot.

5. Electronic-based VVSs: Presentation and classification

We present the evaluated *electronic based* VVSs in Fig. 6. The idea behind them is that they depend primarily on e-voting procedures, even though some of them may have paper *receipts* to provide E2E verifiability, in order to offer higher *confidence* to voters. From the HAVA classification, we present solutions on three out of the four types: *process separation-*, *evidence-* and *end-to-end cryptography-based*.

5.1. Process Separation-Based VVSs

As we have explained before, a process-separated VVS is divided into two independent and isolated subsystems: ballot generation and casting. In this type

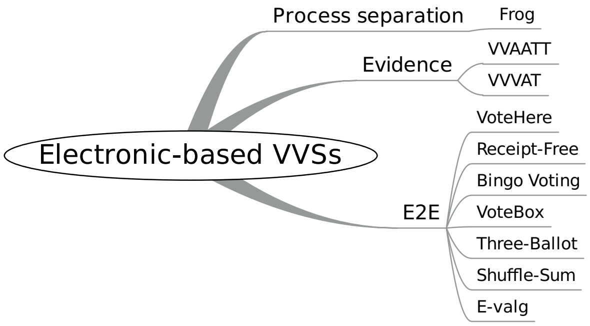

Figure 6: Summary of evaluated electronic systems.

of systems, the security constraints are mainly applied on the casting process. We present below Modular Voting Architecture, namely "Frog" [14], the most representative system in its category.

5.1.1. Modular Voting Architecture ("Frog")

S. Bruck, D. Jefferson and R. Rivest presented this system in 2001 [14]. It implements an integral e-voting solution that emphasizes and standardizes a separation between vote *generation* and vote *casting* components.

On the Elections Day, the voter identifies herself to the poll worker who takes a blank ballot (ballots are named frogs, which are hardware devices), initializes it and, then, returns the ballot to the voter. Afterwards, the voter inserts her ballot into the vote generation equipment, she selects her options through a direct-recording electronic (DRE) voting machine, and her choices are typed into her ballot. The second phase starts here. The voter introduces her ballot into the vote-casting equipment and checks the content of her ballot. When the voter agrees with the content, her ballot is digitally signed (using a single key for all votes), then frozen (blocked against writing) and finally deposited in the frog bin. At this moment, an electronic copy of her vote is randomly stored into a data memory unit and replicated in other memories for reliability. Once elections are over, elections officials publish the results for each precinct in a Web as two separated, unlinked lists: one with the voters' names and the second one with all cast ballots with a system-wide digital signature. Therefore, anyone can verify the digital signature and compute elections' results.

5.2. Evidence-Based VVSs

These systems capture the actions performed by voters when casting their votes, regardless of the voting system and invisible by the voter. In addition to that, to ensure information integrity, all recorded events are stored outside the vote terminal. Under this type of VVSs, we consider VVAATT [77] (together with VVVAT [27]).

5.2.1. Voter Verified Audio Audit Transcript Trail (VVAATT)

VVAATT is an audio verification system, introduced by T. Selker and S. Cohen in 2004 [77, 78]. This system records the audio in all events during the voting process into a physical medium (in a cassette tape or in a CD-W media), while this is complemented by the visual verification from the DRE. On the other hand but in the same line, Voter Verified Video Audit Trail (VVVAT) captures the sequence of screenshots on the DRE terminal (see Cross et al. [27] for an example).

5.3. End-to-End Verifiable VVSs

In this section, we present the E2E cryptographic-based VVSs, which among other capital properties have an end-to-end (E2E) verifiability. To do so, some of them generate paper receipts to allow voters to check that their votes were accounted for the tally process. The following solutions are the selected systems under analysis: VoteHere [46], Receipt-Free [43] (together with Split-Ballot [44]), Bingo Voting [12], VoteBox [71], Three-Ballot [73], Shuffle-Sum [10] (together with Peng and Bao [58]) and the most recent ErgoGroup/Scytl proposal [49].

5.3.1. VoteHere

VoteHere is an integral solution introduced by C. Andrew Neff and VoteHere, Inc. in 2001 [46, 84]. This system is based on the use of DRE terminals. It is built considering receipt- and cryptography-based verifications, in order to cover both *individual* and *universal* verifications.

For each voter, the voting system builds a *code* for each electable candidate before the elections start. Once the voter has chosen her preferences on the DRE, the DRE shows the codes related to each candidate. If they correspond to those pre-built codes, the voter confirms her vote and a *receipt* is printed with her *verification codes*. When the elections end, the *encrypted votes* are made publicly available (providing ballot secrecy) and, then, the voter can *check* if her vote was accounted for (or complain to elections officials otherwise).

5.3.2. Receipt-Free

This system was developed by T. Moran and M. Naor in 2006 [43]. It is based on statistically hiding commitments by using the secret of the commitment to cipher the vote in order to achieve everlasting privacy. An equivalence proof, a type of non-interactive ZKP, is defined combining these commitments with a set of challenges with the aim of allowing the voters to *challenge* or *question* the honesty of the DRE when it encrypts the vote. In addition, this system contributes with a formal definition of *receipt-freeness* and uses secure (integrity) proofs in the Universal Composability (UC) Model [1].

[1] The UC model is a means for defining the security of cryptographic protocols. The name stems from the fact that instances of protocols that are UC secure remain secure even if arbitrarily composed with other instances of the same or other protocols.

This DRE-based system clearly separates the voting process into two phases: cast and tally. In the cast phase, the system encrypts votes and constructs commitments to them with the aim (i) to ensure *statistically* that the content of the votes is as intended, and (ii) not to disclose the confidential information included in votes. Briefly speaking, the casting protocol is as follows:

1. The voter selects her choice. The DRE encrypts her vote (*i.e.,* computes the so-called *real* commitment) and prepares a proof that this commitment holds with her choice. This proof consists of computing k masked copies of the real commitment.

2. The voter introduces k dummy challenges (random words). The DRE generates k commitments in response to these dummy challenges for each candidate different to the voter's choice. Depending on the challenges, commitments are prepared to be either (i) *opened* (without linking the voter's choice) or (ii) *weighed* (*i.e.,* partially decrypted, obtaining a copy of the real commitment).

3. The system puts in a sole commitment x all the generated commitments, including the real commitment, and prints it on a receipt.

4. The voter inputs the real challenges.

5. The DRE computes the answers to the real challenges. It also prints on the receipt the name of the candidates with their respective challenges and the voter name.

6. The voter verifies that both the printed challenges on the receipt and those ones visible on the DRE screen are the same. Then, the DRE prints the response to the challenges and the line "RECEIPT CERTIFIED". Afterwards, it sends a copy of the receipt to the public bulletin board, along with the answers to the challenges and the information needed to open the commitment.

Note that all (real and dummy) commitments and challenges allow the voter to verify that her vote has been cast as intended (if the answers to the real challenge are correct). However, this proof can be difficult to address by the voter, since the voter should make hard calculations in situ (*e.g.,* respond to the challenges).

In the tally phase, each DRE announces the local final results. Then, the DRE carries out a set of proofs to check their integrity. This is performed by following the next protocol. For k times: (i) The DRE masks and permutes the ciphered votes; (ii) A random beacon generates a set of challenges; (iii) According to these challenges, the DRE *opens* the masked commitments (without identifying the voter thanks to the permutation), or *weighs* the masked commitments comparing them to the real commitments (ciphered votes). All this proves that the DRE has not altered the results. By following this protocol and computing these proofs successfully, this solution provides highly probable voter anonymity despite the voter's name being printed in her receipt.

A critical aspect is that private keys used to open the commitments are owned by a single party (the DRE). For this reason, Receipt-Free has been improved, resulting in the *Split-Ballot voting* scheme [44], so that the key is distributed among several parties by using a threshold scheme.

5.3.3. Bingo Voting

Bingo Voting is a integral solution introduced by Jens-Matthias Bohli, Jörn Müller-Quade and Stefan Röhrich in 2007 [12, 11]. This system is focused on the following three targets: (i) separate the relationship between ballot and voter, (ii) avoid voter coercion, and (iii) offer universal verification.

To do so, Bingo Voting uses some cryptographic techniques and tools: (i) commitments, (ii) ZKPs, (iii) receipts and bulletin board, and (iv) a secure random number generator. Broadly speaking, the commitments are essential to achieve everlasting privacy like Receipt-Free and to differentiate between the real choice from the "dummy votes". ZKPs are necessary to check the *tally correctness*. Receipts and the bulletin board help the voter to address her *individual* vote *verification*. Finally, the random number generator participates in the verification that votes are cast as intended.

All these techniques and tools are combined in a voting protocol to intelligently achieve the aforementioned targets. The originality of its approach encouraged us to include this system in our evaluation work. Briefly speaking, the protocol is as follows. There is a data structure with the following information: (i) a list of dummy votes per electoral candidate, and (ii) commitments of random numbers binded to each candidate (generated before the voting process starts). The number of commitments corresponds to the number of candidates and voters. In the cast process, when a voter makes a choice, a random number is linked to the choice. A dummy vote is assigned to the rest of the non-chosen options. The set formed by the dummy votes and the random number become the ballot, which is also delivered to the voter in the form of a *receipt*. The system also records her ballot to later count and verify the results. Once this operation is concluded, the voter can then check if the number that appears in the DRE screen is the same as that in the receipt. The tally process consists of automatically computing the unspent dummy votes and performing a suite of proofs to ensure the correctness of the results. Once this process has finished, all this information is published on the bulletin board to allow voters and third parties check that all votes were correctly accounted for.

5.3.4. VoteBox

VoteBox is an integral solution and was developed by D. Sandler, K. Derr and D. Wallach in 2008 [71]. VoteBox system uses a technique adapted from Benaloh's work on voter-initiated auditing [9] to gain end-to-end verifiability. In other words, the voting system is actually an audit system that records everything happened. Its properties are the following:

- **Pre-rendered user interface.** The user interface is built from *pre-rendered* graphics, a closed sequence of pages (screens) containing text and graphics that reduce runtime code size. The only interactive elements are buttons, rectangular regions of the screen (VoteBox supports touch screens), and other assistive technologies (computer mice, keyboards or audio feedback to state transitions).

- **Tamper-evidence and replication.** A *permanent*, *tamper-evident* audit system records the events along the voting process and provides *resistance* to data loss in case of failure or tampering. VoteBox consists of two parts: Supervisor console and VoteBox booths (*i.e.,* voting terminals). A broadcast network connects both parts, so that events from both parts (including ballot casts or supervisor commands) are replicated on all voting terminals and entangled with a hash chaining in order to provide *immutable* logs.

- **End-to-end verifiability.** To encrypt ballots, VoteBox uses an ElGamal variant that is *additively homomorphic*. Any cast ballot is encoded in a binary format and encrypted by a public key of the elections. Therefore, the tally is addressed by (i) the multiplication of all ballots and (ii) the multiplication result decryption in order to obtain the elections' results.

More recently, in 2009, E. Öksüzoğlu and D. Wallach have presented VoteBox Nano [53] that "follows the same basic design principles as VoteBox, a full-featured electronic voting system" while addressing some weaknesses with the VoteBox approach. Basically, this solution optimizes the full VoteBox protocol, including the hardware and software implementations, on security (integrating "a hardware true random number generator") and computational performance (providing a dedicated hardware). Specifically, from the security point of view, VoteBox Nano allows to verify whether the integrity of the system has been compromised with malicious software.

5.3.5. A Three-Ballot-Based Secure Electronic Voting System

This system, *Three-Ballot* for short [73], is based on the original, paper-based ThreeBallot system [63], but completely redesigned to provide a full electronic solution.

The idea behind the classic ThreeBallot approach is that a *ballot* consists of three single *parts*, with a list of candidates in the same order on all three parts. In order to vote for a candidate, the voters mark *any two parts* on the corresponding candidate (whilst marking only one part means not voting for). When casting the vote, the three parts are separated from each other and mixed with the rest of parts from other voters. The tally operation is done by a simple calculation of the number of marks for each candidate on all the parts. One out of the three parts is randomly chosen by the voter to *copy* and to take home as a *receipt*.

The electronic Three-Ballot scheme [73] follows the same idea behind the paper-based approach but it is implemented by means of a fully computerized architecture which is built using the following entities: a registration agent, a voting console, a voting manager, an electronic ballot box, and an electronic elections bulletin board.

To vote, voters go in person to the registration agent to get a credential that qualifies them to vote. The registration agent interacts with the voting manager to obtain the corresponding ballot IDs and uses them to build credentials that are returned to the voters. Later, after authentication, voters use the voting console to vote, and then the voting manager stores the votes in the electronic ballot box while the voting console gives a voting receipt back to each voter. When elections finishes, the electoral authority and election representatives start the counting phase, counting the votes and publishing the receipts in the electronic election bulletin board.

5.3.6. Shuffle-Sum

This is a verification system developed in 2009 by J.Benaloh, T.Moran, L.Naish, K.Ramchen and V.Teague [10], that implements a verifiable tally for preferential voting schemes (in particular for Single Transferrable Vote (STV) or Quota Preferential Voting). Similarly to VVPAT, this proposal is not a complete voting system but it is covered in this survey due to its relevance.

Widely known, preferential voting schemes suffer from coercion (commonly named "Italian coercion attack" [58]) when all (paper-based) ballots are disclosed for their verification. Clearly, this kind of voting scheme presents a trade-off between the verifiability and the coercion-resistance. This system's goal is, hence, to prevent voter coercion, while (i) votes remain publicly verifiable and (ii) ballots are not disclosed. The authors assume the existence of some mechanism for casting votes and convincing voters that the ballots are consistent with their intentions.

The protocol used to tally votes is verifiable, does not reveal the ballot's content and, to do so, combines homomorphisms and mixing schemes. Both techniques are blended differently in the other considered systems. The basis of this system is a complex data structure, called *table-sum*. It stores voter preferences and facilitates the tally process without the decryption of the votes (by applying homomorphic cryptographic properties). Tallying is an iterative process consisting of multiple rounds, each consisting of four steps: (i) Compute first-preference tallies, (ii) Elect a winner or eliminate candidates, (iii) Reweigh votes, and (iv) Eliminate candidates. Note that, in each iteration, ballot representation in the *table-sum* changes using mixing. In order to verify the correctness of the tallying process, the votes and his content have not changed, the protocol uses a set of ZKPs. Shuffle-Sum also uses threshold scheme in order to share the elections key in several parts, the responsibility is distributed in multiples authorities.

Very similar to Shuffle-Sum, Peng and Bao [58] have also designed an e-voting scheme relying on homomorphic cryptography and mixing techniques to prevent the "Italian coercion attack" in preferential voting systems. This alternative, however, uses matrices (rows indicate preference order and columns candidates) to format ballots, with only an encryption of a '1' into every row and column (the rest of the cells contain encrypted 0's). This scheme also conforms the same 4-phase process depicted before. Notice that, differently to Shuffle-Sum, (i) deleting a candidate means eliminating a column, and (ii) in every necessary round to calculate the winner, homomorphic re-encryption is used to hide any row or column update/deletion. As it happens in Shuffle-Sum, this scheme also uses threshold-decryption in the tally and ZKPs for universal verification. Given the high similarity among these schemes, we only analyze Shuffle-Sum in depth in the present work.

5.3.7. E-valg 2011

The Norwegian Ministry of Local Government and Regional Development initiated in 2008 a selection process of e-voting technological providers, which finished in December 2009. As a result of this process, ErgoGroup [2] and Scytl [3] were selected as the providers of the e-voting solution for the Norwegian municipal elections [49, 50].

The ErgoGroup/Scytl's solutions meet all the security requirements by using cryptographic techniques. Until now, ErgoGroup/Scytl consortium has designed various systems to support two types of voting: *poll-site-based* (compatible with DREs) [76] and *remote voting* [59]. Moreover, the latter allows a supervised remote voting model, which meets the system requirements specification of the E-valg 2011 project [51].

The electronic voting approach of the E-valg 2011 is specifically designed to work in a supervised remote environment but it also supports unsupervised voting models. Therefore, this scheme offers the voter two different options to cast her vote: (i) through a computer at the polling station; and (ii) from any remote computer connected to the Internet. In both cases, the workflow of the protocol is the same, however, in the latter situation, the presence of malicious programs in the voter's computer is specifically addressed.

The process of casting a vote in the Norwegian e-voting system, starts with the voter receiving a voting card by classic mail some time before the elections. The voting card contains pre-computed random generated receipt codes corresponding to possible voting options on the elections ballots. These codes are specifically generated for each voter but they are random and they do not reveal what the voter voted.

Then, the process at the polling site is the following:

1. The voter uses a computer provided by the elections officers and she provides her electronic ID credentials to the the voting application (Voting Client). This application authenticates the voter to the voting server.

2. The Voting Client displays the list of parties and candidates and the voter mark the options to issue her choices.

[2] http://www.ergogroup.no/default.aspx?path=\{2A1C0F50-F200-43C8-98C6-36CD82F7A587\}

[3] http://www.scytl.es

3. The voter submits her vote and, then, the ballot is encrypted using homomorphic encryption[57, 56], zero knowledge proofs are attached and everything is digitally signed with the voters electronic ID to achieve vote authenticity. The encrypted and signed ballot is sent to the vote collection server.

4. The system receives the vote and the digital signature and the cryptographic zero knowledge proofs are validated. If accepted, the system updates the electoral roll and issues a voting receipt of the vote for E2E verifiability. This receipt is sent to the voter by SMS and contains a code corresponding to the vote she has issued. Then, the voter can verify that this code is correct against the codes on her voting card.

After that, the process of counting votes begins and the electronic ballot box collected by the vote collection server is transferred securely to a new system in a secure location. The electronic ballot box, which is digitally signed to ensure its authentication and integrity, contains the digitally signed encrypted votes and cryptographic proofs generated by the voters. The digital certificates are validated and the digital receipts from all the e-votes are kept to allow voters to verify that their votes have reached the counting process. Then, the digital signatures of the ballot box and the e-votes are verified to check their authenticity and integrity. After these verifications, e-votes' digital signatures are removed to make them anonymous. Then, those anonymous votes are processed by a reencryption universal verifiable Mix-net, which breaks any correlation between the votes and their voting order. The output of this mixing process is all the votes re-encrypted and shuffled and a set of zero knowledge proofs for verifying the correct mixing process. The integrity of the mixing process is validated and the Electoral Board uses a *threshold scheme*[79] to decrypt the votes. Finally, the list of all decrypted votes are digitally signed by the Electoral Board and they are finally counted.

6. Study and comparison of VVSs

In this section, we introduce the analysis of the considered VVSs by type (Sect. 6.1 for paper-based and Sect. 6.2 for electronic-based ones) and the study of the synergies of voting systems and cryptographic technologies (Sect. 6.3).

6.1. Paper-based VVSs: Analysis and comparison

The properties considered in our common evaluation framework are next used to compare and analyze all evaluated paper-based VVSs. Note that VVPAT is considered to be a verification system (instead of a complete voting scheme like the rest of paper-based proposals being examined). Therefore, it is only evaluated according to the following set of properties: individual verification, universal verification, auditability and integration.

Table 3 shows and compares the results achieved by each scheme for each analyzed property.

6.1.1. User interaction

All systems present **accessibility** issues for disabled people (*e.g.,* blind voters) given that they all rely on paper representation of the voting preferences. However, the initiative for Voting-on-Paper Assistive Device (Vote-PAD) [52] can be applied to several of them (e.g. Punchscan and Prêt à Voter [21], and Scantegrity II [19]), so that these voting systems turn *accessible* for people with visual or dexterity impairments. The elaboration of the paper ballot may become difficult, and thus the voting act becomes more complex. For the purpose of exposition, in Punchscan, the paper ballot consists of two overlapped sheets of paper, with a randomized order of candidates in the top sheet, and a randomized order of symbols in the sheet beneath. The more candidates exist, the greater the ballot and the greater resource and computation consumption is. In consequence, the resulting voting procedure can take longer and turn more confusing than in traditional ones. For instance, voters in ThreeBallot must verify their ballot marks against the *checker machine* before casting it.

6.1.2. Security

Voter-related security. All the analyzed systems are opscan-based voting systems (note that, as explained previously, VVPAT is not included here). They all provide **ballot secrecy** implemented in different ways. Some of them provide human-readable information *only when all parts are together* (not when ballot parts are separated and mixed with others –ThreeBallot– or when discarded a half of the ballot –Prêt à Voter, Punchscan and Scratch & Vote–). Ballots in Scantegrity II, however, show all candidates and the voter's marks, but sensitive information appears encrypted (2D-barcode). In addition, all these systems (except ThreeBallot) use some tables for *mixing* and **anonymizing** ballots. On the other hand, Scantegrity II also addresses this property by scanning the ballot identifier (from a 2D-barcode) and the selected option. As a result, in these voting schemes, third entities (*e.g.,* observers or devices involved in the voting process) are prevented from seeing the content of the ballots.

As all these systems construct pre-printed ballots, voters could incorporate information into the ballot with the idea to uniquely identify it and, therefore, enable **coercion and vote selling**. This information could be inserted (i) somewhere in the paper ballot or (ii) as a specific selection of the candidates or "pattern voting" (*e.g.,* in the form of selected candidates or positions from the ballot). The first issue is clearly solved by (i) declaring the vote null by elections officials or (ii) by their omission at the opscan device, since opscan-based systems only scan pre-specified areas, and discard any other mark outside. However, addressing the second issue is a harder problem. Let us introduce the main well-known attacks that affect these voting systems.

The "Italian coercion attack" [58], specific for preferential elections, allows a voter to vote in a predefined ordering of selected candidates, so that it can be checked publicly at the end of the elections using the public bulletin board. The underlying idea is to build a "pattern voting" that can seldom be reproduced. This clearly enables voter coercion and vote selling.

ThreeBallot has been reported to be vulnerable to this attack [82]. However, in order to be successful, it is required a sufficient number of candidates to choose from, so that unique candidate selections can be made. In this way, some solutions addressed this attack in ThreeBallot by ensuring the "short ballot assumption" (SBA) [64]. As stated in Clark et al. [24], and further developed in Henry et al. [35], the number of candidates should be small enough (*i.e.*, guaranteeing the SBA) to prevent this lack of *resulting* anonymity. Similarly to the SBA, another alternative to prevent this issue is partitioning single ballots into smaller parts [64, 35], so that the number of candidates in each part remains sufficiently small. Nevertheless, the authors in [38] measured the degradation of coercion-resistance of ThreeBallot in case of applying the so-called short ballot assumption and they showed that the level of coercion-resistance provided by this voting scheme was significantly lower than that of an ideal system, even in case of short ballots.

Regarding Scantegrity II, this voting system does not suffer from this attack, as the public bulletin board only allows for codes interaction instead of graphical representations of the receipts. In this way, the authors in [39] prove that Scantegrity II enjoys an optimal level of coercion-resistance.

Another kind of "pattern voting" attack, called coerced randomization, suitable for *pre-printed* paper ballots and opscan-based voting systems with a public bulletin board (*i.e.*, Prêt à Voter, Punchscan and Scratch & Vote), allows for *randomized vote coercion* [3]. In this type of attack a coercer forces voters to mark a specific position in the list of candidates. Therefore, with a sufficient number of candidates, the coercer can reduce statistically the number of votes for any party. A workaround for this attack is to print ballots in-place.

Finally, since they all provide receipts to take away, voters can **individually verify** their vote selections with augmented E2E verifiability. In ThreeBallot, even though only 1/3 of the multi-ballot is verified, the probability of ballot box tampering is decreased exponentially by the set of all voter verifications. The **individual verifiability** of VVPAT is *limited* to the operations at the poll site and relies on: (i) the verification from independent and party observers; and (ii) the trust on the custody chain of ballots.

Voting-related security. All these systems consist of a mix of processes and technologies. Therefore, depending on the way they are combined, this situation may lead to certain security issues. To ensure **ballot box integrity**, voters are registered when voting and they all (except VVPAT and Scantegrity II) show the receipts on a public bulletin board with verification codes. Besides, all except VVPAT and ThreeBallot allow verification of part of the mixing tables by elections officials. When counting votes, these systems present different **tally approaches**. VVPAT relies on the DRE-based counting, while printed ballots are counted only when required by law, to check partial (district) results against electronic ones. Conversely, Punchscan, Prêt à Voter and Scantegrity II use specific tables to decode votes and then, to count the results. Scratch & Vote uses additive homomorphic encryption (encoded in the 2D-barcode), whilst ThreeBallot presents the simplest tally addressed by an arithmetic operation ($N - V$). In Prêt à Voter, though, elections officials may collude to alter tally

Table 3: Detailed properties of the paper-based VVSs.
For details on notation see Table 2.

VVS	User Interact.		Security								Integration		Technical Issues		
			Voter-related				Voting-related								
							Universal Verification								
	Accessibility	Use Impact	Ballot Secrecy	Anonymity	Coercion Resistant	Individual Verification	Ballot Box Integrity	Tally Accuracy	Fairness	Auditability	Integration	Data Management	Fault tolerance	Scalability	Flexibility
Prêt à Voter	↓	↑	Y	Y	N	Y	Y	Y	Y	Y	T/SW	N/A	↑	↓	~
Punchscan	↓	↑	Y	Y	N	Y	Y	Y	Y	Y	N	N/A	↑	↓	~
Scratch & Vote	↓	↑	Y	Y	N	Y	Y	Y	Y	Y	T/SW	A/NDL	↑	↓	~
Three Ballot	↓	↓	Y	~	N	Y	Y	Y	Y	Y	N/A	N/A	↑	↓	↓
Scantegrity II	↓	↑	Y	Y	Y	Y	Y	Y	Y	Y	T/SW	A/NDL	↑	↓	~
VVPAT	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	Y	Y	T/SW	N/A	N/A	N/A	N/A

results [69].

However, all these systems provide **universal verifiability** that, in turn, enables to check if there are errors at any point of the system and, hence, offers highly probable box integrity, tally accuracy and fairness. In addition, since these systems mix procedures and cryptographic technology, their **auditability** is quite different from those strictly electronic ones (see Section 6.2). The procedures are auditable following the notes in paper documents (*e.g.,* voter lists) taken by electoral officials. On the contrary, cryptographic parts (mainly mixing tables and verification codes) are auditable before, during and after elections in all of them except VVPAT. In addition, as Prêt à Voter, Punchscan, Scratch & Vote and ThreeBallot provide the content of the receipts in the public bulletin board (in ThreeBallot all 3 parts), they provide public auditability of the cast ballots by voters, independent and party observers.

It is necessary to mention that Punchscan and Prêt à Voter have served as the basis for new voting protocols (like [83]). In the case of Van De Graaf [83], the proposed voting protocol turns *unconditionally private* and the auditing tables *computationally binding*, when some premises are assured (*e.g.,* using bit commitments -like in Punchscan- that are *unconditionally hiding and computationally binding*; or using several copies of the auditing tables to reduce exponentially the probability of electoral officials to cheating and not being caught). As a result of the use of the permutations and bit commitments in the mixing table, whenever the auditing procedures succeed in both pre- and post-elections phases, the elections' results are (w.h.p.) integral and correct even if electoral officials misbehave or there is some error in the voting software.

6.1.3. Integration

VVPAT is designed to be an independent verification system and it provides a nice **integration** with DRE-based voting systems. Note that, VVPAT was originally designed to work with DREs but recent research has shown that it can also be integrated with other mechanisms [68]. Scantegrity II is a complete voting scheme but its verification part can also be integrated with other opscan-based voting schemes. In the same way, Scratch & Vote can be easily adapted to other paper-based voting systems (as depicted by authors in Adida and Rivest [3]). The rest are hardly integrable with other systems due to their paper ballot characteristics. Some authors (like in Prêt à Voter) state that their systems could be integrated with DREs in order to build in-place paper ballots (instead of pre-printed ones). Additionally, Scratch & Vote and Scantegrity II, which uses encrypted 2D-barcodes, also provides good **data management** properties, such as voting *atomicity* and *no data loss*.

6.1.4. Technical issues

All these schemes are based on pre-printed ballots and, hence, they can be considered to provide a high level of **fault tolerance**. VVPAT only requires a printer device attached to the DRE. For the same reason, given that they all (except ThreeBallot) require at least *twice* (*thrice* on ThreeBallot) the number of total voters in pre-printed ballots for verification purposes, these systems do not scale in terms of number of ballots. The opscan-based voting systems do not allow for write-in modes, but single/multiple mark or ordering. Note that this limitation prevents quite flexible schemes like Prêt à Voter from achieving the high level of flexibility. Specifically, this voting scheme is able to handle multiple elections methods with the same interface [86] but it does not allow write-in modes.

6.2. Electronic-based VVSs: Analysis and comparison

The properties considered in our common evaluation framework are next used to compare and analyze all evaluated electronic-based VVSs. Note that Shuffle-Sum is not a full system. It only implements a verifiable tally and assumes the existence of some mechanism for casting votes and convincing voters that the published ballots are consistent with their choice. For this reason, this system is only evaluated according to the following set of properties: universal verification, auditability and integration.

Table 5 shows and compares the results achieved by each scheme for each analyzed property.

6.2.1. User interaction

Given that all VVSs use DREs to emit votes, all of them provide a certain degree of **accessibility**. Also, some systems improve it by using audio guides (VVAATT), or indeed with other assistive technologies (such as mice or keyboards) (VoteBox and E-valg). For the E-valg case, this is proved by the studies of Sherman et al. [80], Norwegian Ministry of Local Government and Regional

Development [48]. Nevertheless, [66] states that DRE machines and other assistive mechanisms often do not work as promised. As a result, inadequate or malfunctioning equipment forces voters with disabilities to ask for assistance or compromise the privacy of their vote. On the other hand, some current systems are not designed for blind people yet (*e.g.,* Bingo Voting). As for the **use impact**, systems like Frog, VoteBox and Three-Ballot present a more complex and likely longer voting process. For instance, in Frog, there exists a strict separation of the generation and cast processes (even though a voter can bring a filled ballot from home); VoteBox allows voters to perform an "immediate ballot challenge" [9]; or Three-Ballot uses a multi-ballot composed of three parts. Other systems, like VoteHere, Receipt-Free, Bingo Voting and E-valg, make easier the user interaction. In these cases, the user only must easily assess the equality of a code between a receipt and the DRE's screen, or between a receipt and a bulletin board. In the systems based on the Neff's "MarkPledge" scheme [47, 2], like VoteHere, Receipt-Free and Bingo Voting, the user must also type in the DRE various random strings as "challenges", and then, verify the responses of the DRE. However, the verification of these challenges could be a difficult work for the voter.

6.2.2. Security

VVAATT/VVVAT do not ensure vote confidentiality, given that all (audio or video) recordings show the *sequential* voting order. In addition, VVAATT/ VVVAT suffer from *weak* recording equipment protection (they must be accessed often) and *untrustworthy* information extraction techniques. As a consequence this system does not cover the anonymity of the voter and is not reliable in its entirety. Note that VVAATT/VVVAT has been included in the study not only as the referent for evidence-based systems, but also as one of the first systems that provides an incipient audit mechanism. Next, we will only focus on the rest of systems.

Voter-related security. Except Frog, all systems ensure **ballot secrecy** using a public key infrastructure (PKI), most of them based on ElGamal [29] algorithm. Some other systems hide the vote in a Pedersen commitment [55]. These electronic-based VVSs use very different techniques to provide highly probable **ballot anonymity**, though. While Frog uses a simple randomization algorithm, Three-Ballot separates each of the three parts of a ballot and stores them using their hash values. More complex techniques also appear: mixing (VoteHere), additive homomorphic (VoteBox) or a hybrid scheme (multiplicative homomorphic and mixing in E-valg). In addition, some of the systems based on commitments (Receipt-Free and Bingo Voting) are a different kettle of fish because they mix not only the encrypted votes (*i.e.*, the commitments) but also other "masked" commitments used in the verification procedure. This mixing procedure consists of masking and permuting the commitments, by using the homomorphic properties of the commitment scheme. It is worth noting that one of the most important problems of these two previous systems is that a single entity has the responsibility of the private keys (*e.g.,* to open the commitments). However, Split-Ballot's main proposal is just to solve this drawback, by

Table 4: Security techniques used by VVSs

		Security techniques								
		Encryption Techniques	Anonymity Techniques	Threshold Scheme	Receipts	Return Codes	Bulletin Board	ZKPs	Digital Signatures	Audit System
VVS	Frog	N	Randomization	N	Y	N	Y	N	Y	N
	VoteHere	ElGamal	Verified Shuffling	Y	Y	Y	Y	Y	Y	N
	Receipt-Free	Commitments	Remasking + Permutation	N	Y	Y	Y	Y	N	N
	Bingo Voting	Commitments	Remasking + Permutation	N	Y	Y	Y	Y	N	N
	VoteBox	ElGamal	Additive Homomorphic	Y	Opt.	Opt.	Y	Y	Y	Y
	Three-Ballot	RSA	Mixing (Hash)	N	Y	N	Y	N	Y	N
	Shuffle-Sum	N/A	Additive Homomorphic + Mixing	Y	N/A	N/A	Y	Y	N/A	Y
	E-valg	ElGamal	Multiplicative Homomorphic + Mixing	Y	Y	Y	Y	Y	Y	Y

Y/**N**/**"N/A"**/**Opt**: **Y**es/**N**o/**N**ot **A**ddressed/**Opt**ional

using a threshold scheme. In consequence, Split-Ballot shares this responsibility through the electoral board, and prevents against well-known security attacks to the electoral system. Receipt-Free, Bingo Voting, VoteBox, Three-Ballot, and E-valg are **resistant to coercion and vote selling** because the information necessary to verify the vote, kept by the voters, is not enough to demonstrate their vote intention to a third person. The same is not true for VoteHere, since it may have a flaw given that it shows both encrypted ballots and receipts with return codes [6]. To conclude with the voter-related security, all these systems (except Frog) render *augmented* **individual verification** with *E2E* voter *verifiability* through receipts. Receipt-Free and Bingo Voting enhance this property and allow the voter to distrust the DRE. Their casting protocol combines commitments with challenges in order to prove statistically that the content of the vote generated by DRE is the same as the intended vote. Per contra, the other systems depend on the goodwill of the DRE.

Voting-related security. Except for Frog, all systems ensure **ballot box integrity** through different technologies (like ZKPs, digital signatures or threshold schemes). Some of them also keep the integrity by means of verification protocols that detect and recover from any changes in the ballot box. For example, Bingo Voting can find discordances between the number of votes (which are not commitments, but random numbers) and the number of unused commitments. See Table 4 for more details.

VoteHere, Receipt-Free, Bingo Voting, VoteBox, Three-Ballot, Shuffle-Sum and E-valg provides highly probable **tally accuracy** and **fairness**. Receipt-Free must be particularly highlighted, since it combines homomorphic techniques with a special protocol that enables the complex tallying for preferential elections. In consequence, most of them provide highly probable **universal verifiability**. As for **auditability**, Three-Ballot creates logs for any voter-related operation, even though nothing about the tally process. The evaluated strongest audit systems appear in VoteBox and E-valg that use immutable logs. VoteBox, however, builds a distributed total audit system, while E-valg only centrally audits the critical system elements.

6.2.3. Integration

In order to be **integrated**, the evaluated VVSs have some software or technological dependences (see Table 5 for more details). On the other hand, Shuffle-Sum is uncoupled from the cast mechanism: it does not imposes a particular method and, hence, it has a good level of integrability. At the end, only VoteBox and E-valg [80] ensure vote atomicity, loss resistant and tamper evident solutions.

6.2.4. Technical issues

VoteBox is the only system that structurally provides distributed *replication* of sensitive information, which leads to a high degree of **fault tolerance**.

Homomorphic systems that aggregate all the votes in a single ciphertext provide poor **scalability** due to the fact that they restrict the maximum number of voters who can participate in the voting process. More specifically, this value is fixed according to the cryptographic key length in use and the number of parties involved in the elections. In contrast, homomorphic schemes that use a *vector ballot* (*i.e.,* different ciphertexts are used to aggregate the votes for each party) do not suffer scalability problems in the tally process (but they increase the complexity of the vote casting process due to the generation of zero knowledge proofs). Regarding the systems which are based on mixing techniques, the cost of shuffling “n” votes is linear (*i.e.,* O(n) [56, 57]) and, hence, they provide low scalability as well. Finally, those proposals that combine mixing and homomorphic techniques offer significant computational scalability in the tally process because combination/aggregation of votes (*e.g.,* by means of homomorphic properties) reduces the input dataset to the mixing and, hence, the computational cost also decreases. VoteBox uses homomorphisms and the vector ballot approach [71]. Besides, it can also be used in an hybrid fashion by combining homomorphic and mixing techniques [72]. E-valg is also an hybrid scheme and, hence, it is able to use either mixing, homomorphic properties or a combination of both depending on the situation. As a result, these two proposals achieve a better level of scalability than others. Nevertheless, since E-valg and VoteBox address the supervised remote voting, they could suffer from an overload of the voting system if many voters cast their vote at the same time. Therefore, these schemes must provide the necessary infrastructure in order to prevent this breakdown.

Broadly speaking, VoteBox and E-valg are less **flexible** than the rest because the use of homomorphic techniques force encoding the vote in a special fashion. This restricts the type of the ballot in two features: (i) preventing the realization of other opinion polls at the same electoral process; and (ii) limiting the number of electoral candidates. Systems based on commitments and mixing techniques (like Receipt-Free and Bingo Voting) are more flexible, given that they do not allow making several opinion polls at the same time, but, contrariwise, they do not restrict the number of candidates. The reason behind that is because commitments must be computed before the elections, and there must be one commitment for each candidate and voter. Lastly, Three-Ballot is only suitable

Table 5: Detailed properties of the electronic-based VVSs.
For details on notation see Table 2.

VVS	User Interact.		Security								Integration		Technical Issues		
	Accessibility	Use Impact	Voter-related				Voting-related				Integration	Data Management	Fault tolerance	Scalability	Flexibility
			Ballot Secrecy	Anonymity	Coercion Resistant	Individual Verification	Universal Verification			Auditability					
							Ballot Box Integrity	Tally Accuracy	Fairness						
Frog	↓	~	N	Y	N	~	~	N	N	N	T/SW	N/A	N/A	↓	↑
VVAATT VVVAT	~	↑	N	N	N	N	N	N	N	~	T	NA/DL	N/A	↓	↓
VoteHere	↓	↑	Y	Y	N	Y	Y	Y	Y	N	SW	N/A	N/A	~	~
Receipt-Free	↓	↓	Y	Y	Y	Y	Y	Y	Y	N	T/SW	N/A	N/A	~	~
Bingo Voting	↓	↓	Y	Y	Y	Y	Y	Y	Y	N	T/SW	N/A	N/A	~	~
Vote Box	↑	↓	Y	Y	Y	Y	Y	Y	Y	Y	T/SW	A/NDL	↑	↑	↓
Three-Ballot	N/A	↓	Y	Y	Y	Y	Y	Y	Y	Y	N/A	N/A	N/A	~	↓
E-valg	↑	↑	Y	Y	Y	Y	Y	Y	Y	Y	N/A	A/NDL	N/A	↑	↓
Shuffle-Sum	N/A	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	N	SW	N/A	N/A	N/A	N/A

for multi-ballot formats composed of three single parts, even though the ballot content is flexible.

6.3. Study of trends in VVSs

From the present analysis we can extract *four clear trends* related to the following issues: (i) voting location, (ii) voting technology, (iii) degree of verifiability and (iv) auditability. Apart from those systems being included in this work, we also present a new *trend in voting processes.*

Voting location study. We have evaluated *poll-site-based* VVSs. All of them use paper ballots or electronic voting terminals (DREs). Clearly, DREs are very helpful in order to manage votes electronically. It is worth noting the demonstrated trend from *poll-site-based* to supervised remote voting systems. All systems are of the first type, while VoteBox and E-valg are supervised remote voting systems. This trend is a consequence of not only the technology, but also the temporal appearance of the systems. However, while VoteBox was *adapted* to support supervised remote voting schemes, E-valg was *structurally* designed to do so.

Voting technology study. Here, we consider the voting technology used from the ballot cast to the tally and, therefore, VVAATT/VVVAT-based systems are not considered. The idea behind this technology is to address security issues such as ballot anonymity, ballot box integrity, tally accuracy and fairness among others. These systems present a clear evolution on this issue. We detail them from simpler to more complex and reliable solutions, separating paper-based from electronic solutions. As for paper-based systems, ThreeBallot employs arithmetic computation, Punchscan uses generation of permutations, Prêt à Voter makes use of encrypted permutations in a human readable code, Scantegrity II utilizes encrypted 2D-barcodes and invisible ink and Scratch & Vote deploys Paillier PKI with additive homomorphic properties, and encrypted 2D-barcodes. In the electronic systems, while Frog uses only a simple *randomization* algorithm to anonymize votes, others, like VoteHere, Receipt-Free or Bingo Voting, utilize more reliable mixing techniques to address *ballot anonymity*, with *everlasting privacy* in Receipt-Free and Bingo Voting. VoteBox and Three-Ballot use *additive homomorphic* cryptography, to provide highly probable *ballot anonymity* and perform the *tally*. The most complex but reliable technology, with a certain flexibility in the ballot format and type, is used by E-valg: the *hybrid scheme*, which is composed by multiplicative homomorphic cryptography (computationally less hard than the additive schemes [57, 56]) and mixing mechanisms. Clearly, the paper- and electronic-based technology presents a *trade-off* between ensuring (i) more secure, trustworthy and reliable voting technologies, while at the same time providing (ii) fast and resource-efficient ones. This trend from simple randomization techniques to hybrid schemes is a direct consequence of the continuous permeability of voting systems with regards to the latest cryptographic advances.

Verifiability study. We can organize the analyzed systems as follows. In paper-based systems, the verifiability in VVPAT, ThreeBallot and Scantegrity II *eventually* relies on human-readable paper ballots, so that they can be independently and universally verified without the interaction of computerized technologies. This is very attractive for detractors of exclusively e-voting systems. The rest of paper-based approaches require mixing tables to decode ballots and perform the tally, so that both parts (tables and ballots) should be carefully verified. In summary, all these systems provide individual, universal and E2E verifiability (except E2E in VVPAT). Their verifiability becomes a bit harder due to mix of procedures and technologies in the voting process. Instead, in electronic VVSs, (i) VVAATT/VVVAT-based and Frog systems provide *deficient* or *basic* verifiability in voting processes, respectively. They mainly provide at some degree individual verifiability, yet the same is not true for universal or E2E verifiability. (ii) VoteHere and Three-Ballot VVSs offer an *acceptable* degree of verifiability (individual, universal and E2E), while E-valg and VoteBox ensure a *good* level of verifiability, while, at the same time, they define a tough audit system. Finally, (iii) systems like Receipt-Free and Bingo Voting *go beyond* the individual and universal verification, because they provide highly probable everlasting privacy in cast ballots. Otherwise, voters can detect cheating DREs when encrypting their ballot.

Auditability. Complimentary to verification of the elections process, auditing mechanisms addressed by independent parties lead to elections integrity [4]. The idea behind that is to prevent from, deter and foil any attempt of misbehaving or attack on elections. Since a thorough audit of all procedures, technologies and votes is unfeasible or impossible [4], a random selection is taken for such purpose (like in mixing tables, or recounts of paper ballots in some districts). The auditing procedure must cover pre-, in- and post-elections phases. In paper-based systems, most of the (hand-made) audits cover the manual procedures of voters, and more importantly, of elections officials, to detect any voluntary or involuntary human error. Instead, in opscan and electronic-based systems, the hardware and software (including firmware) are carefully audited by *manual* and *electronic* means, respectively. In particular, white-box and black-box[4] software audits should be addressed. Additionally, it has been proven that such kind of pre-, in- and post-elections audits greatly reduces the capability of adversaries to perform attacks to the system [4]. As for demonstration, the elections audited in [4] demonstrated that (i) procedural verifications were correctly performed, (ii) all parts should be audited (even hardware and software vendors), and (iii) opscan hardware worked as expected. It is worth noting that, differently from expected, discrepancies between manual audits and opscan results appeared only due to human errors in the auditing phase [7]. These kind of experiences are necessary to make the electorate aware of the good properties of the electronic voting mechanisms.

Once all VVSs have been analyzed and compared with each other (see Tables 3 and 5) and considering a higher weight for the security properties, Scantegrity II, VoteBox and E-valg result into the best alternatives for paper-based and electronic voting systems, respectively. However, VoteHere, Receipt-Free and Bingo Voting contribute with innovative cryptographic mechanisms such as everlasting privacy not reached in other electronic VVSs.

7. Conclusions

In this paper, we have presented an evaluation framework, common for all systems, in order to conduct a fair study among paper- and electronic-based voting verification systems (VVSs). To do so, we have proceeded as follows: (i) we have defined a classification of VVSs, (ii) we have specified an evaluation framework (combining 15 characteristics from VVSs), (iii) we have selected and analyzed a notable set of 18 VSSs (from both commercial and academic worlds), and (iv) we have extracted important trends in the field.

The *VVS classification* obeys a criteria suitable for the present work, gathering together the VVSs in a natural fashion. In particular, we have used the HAVA classification (USA). Additionally, we have also differentiated from integral and independent solutions, regarding to whether the VVS is tightly tied to a voting system or not.

The *evaluation framework* is composed of a collection of 15 important properties to evaluate on VVSs. These characteristics concern to the system's security, voter confidence on the voting system, and to the vote flexibility (*i.e.*, allowed ballot type and format). Concretely, ballot secrecy, voter anonymity, tally accuracy and fairness are not only highly important, but also essential for

[4] White-box auditing is a method of testing the internal structure and working of the system software. Oppositely, black-box auditing is a method of testing the functionality of the system software.

trustworthy elections. Voter confidence is also a necessary characteristic for a voting system. All this makes citizens trust these kind of tools and mechanisms that support modern democracy.

This work makes *a survey* of 18 notable paper- and electronic-based VVSs. As we have seen, most recent advances in technology and cryptography are making voting systems move from paper-based approaches to electronic-based ones. In particular, some of the strengths of the electronic VVSs are the enhanced security, voter confidence and increased voter accessibility (also for blind and illiterate citizens).

Cryptographic techniques include all the following points: (i) *vote ciphering*, (ii) *digital signature* and *threshold scheme*, (iii) *mixing techniques* and *homomorphisms*, (iv) *ZKPs*, (v) *return codes* and *receipts*, and (vi) *audit systems*. The benefit of these *cryptographic techniques* are as follows. Voting systems use asymmetric cryptosystems (*e.g.,* RSA or ElGamal) to provide ballot secrecy. When digital signatures and threshold schemes are used, systems provide highly probable vote authenticity and an enhanced ballot secrecy. Mixing techniques and homomorphisms ensures voter anonymity. In particular, homomorphic encryption improves the tally efficiency, but it reduces the flexibility of the ballot type and format. Conversely, mixing techniques do not impose any restrictions in the ballot, but usually result in slower tallies. Both mechanisms require ZKPs to ensure universal verification. ZKPs also ensure voter anonymity in the case of mixing techniques. Most of the evaluated electronic VVSs use return codes and receipts to provide the E2E individual verifiability, whilst they prevent voter coercion and increase the voter confidence in the given system. In the end, the auditing capability is progressively a more common feature in the most modern e-voting systems.

As for the *tally process*, it is easy to see an evolution from the use of mixing techniques to the use of homomorphic schemes. In particular, additive homomorphic encryption was used firstly, followed by multiplicative, and then hybrid schemes. Hybrid schemes join mixing techniques and homomorphic encryption and bring together the benefits from both, while at the same time they improve on tally efficiency and ballot flexibility (compared to solutions using solely either mixing or homomorphic schemes).

We have seen an evolution from paper-based to electronic VVSs, thanks to the inclusion of technological advances. Within the set of electronic VVSs, later ones support supervised remote voting schemes. In addition, there exist an increase on the use of the remote (postal) voting [81]. As this evolution demonstrates, we foresee that the *future trend* on the use of electronic voting means is *remote e-voting*. Along these lines, there are already evidences of some initial *supervised remote* and *Internet voting* experiences [31, 40, 74]. The global acceptance of these remote e-voting schemes will empower citizens with new democratic participation tools. This will probably lead to direct and binding citizen consultations and elections in the near future [7].

However, for a full adoption of these remote e-voting schemes, some issues must be resolved beforehand: (i) a higher level of the system's *security* and (ii) an *easier use of and access* to the system (from a voter viewpoint). They both are related to each other. Clearly, the technology employed to ensure a secure e-voting process would likely make it difficult to set up correctly the necessary software and hardware on voters' computers. In particular, future systems in this line should closely take into account the voter's non-supervised environment, which facilitates the voter coercion and vote selling.

Acknowledgments

Authors thanks to Ronald L. Rivest for his comments and suggestions. Authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization. This work was partly supported by the Spanish Ministry of Science and Innovation (through projects eAEGIS TSI2007-65406-C03-01, CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004 and Audit Transparency Voting Process IPT-430000-2010-31), by the Spanish Ministry of Industry, Commerce and Tourism (through projects eVerification2 TSI-020100-2011-39 and SeCloud TSI-020302-2010-153) and by the Government of Catalonia (under grant 2009 SGR 1135).

[1] 107th U.S. Congress, 2002. Help america vote act of 2002 (pub.l. 107-252). U.S. Government Printing Office.
URL `http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ252.107`

[2] Adida, B., Neff, C. A., 2006. Ballot casting assurance. In: EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop. USENIX Association, Berkeley, CA, USA, pp. 7–7.

[3] Adida, B., Rivest, R. L., 2006. Scratch & vote: self-contained paper-based cryptographic voting. In: Proceedings of the 5th ACM workshop on Privacy in electronic society (WPES '06). ACM, New York, NY, USA, pp. 29–40.

[4] Antonyan, T., Davtyan, S., Kentros, S., Kiayias, A., Michel, L., Nicolaou, N., Russell, A., Shvartsman, A. A., 2009. State-wide elections, optical scan voting systems, and the pursuit of integrity. Trans. Info. For. Sec. 4 (4), 597–610.

[5] Aumann, Y., Ding, Y. Z., Rabin, M., jun 2002. Everlasting security in the bounded storage model. Information Theory, IEEE Transactions on 48 (6), 1668 –1680.

[6] Barnes, R., 2004. VoteHere VHTi: A verifiable e-voting protocol. Online June 2010.
URL `http://www.cs.virginia.edu/crab/VoteHere.pdf`

[7] Barrat Esteve, J., 2006. A preliminary question: Is e-voting actually useful for our democratic institutions? what do we need it for? In: Proceedings

of 2nd International Conference on Electronic Voting (E-VOTE '06). GI, pp. 51–60.

[8] Bellis, M., 3rd November 2009. The history of voting machines. Online Feb. 2010.
URL http://inventors.about.com/library/weekly/aa111300b.htm

[9] Benaloh, J., 2007. Ballot casting assurance via voter-initiated poll station auditing. In: Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology (EVT'07). USENIX Association, Berkeley, CA, USA, pp. 14–14.

[10] Benaloh, J., Moran, T., Naish, L., Ramchen, K., Teague, V., 2009. Shuffle-sum: coercion-resistant verifiable tallying for stv voting. Trans. Info. For. Sec. 4 (4), 685–698.

[11] Bohli, J.-M., Henrich, C., Kempka, C., Müller-Quade, J., Röhrich, S., 2009. Enhancing electronic voting machines on the example of bingo voting. Trans. Info. For. Sec. 4 (4), 745–750.

[12] Bohli, J.-M., Müller-Quade, J., Röhrich, S., October 4-5 2007. Bingo voting: Secure and coercion-free voting using a trusted random number generator. In: Proceedings of the First International Conference on E-Voting and Identity (VOTE-ID '07). Vol. 4896 of Lecture Notes in Computer Science. Springer, pp. 111–124.

[13] Brassard, G., Chaum, D., Crépeau, C., 1988. Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. 37 (2), 156–189.
URL http://dx.doi.org/10.1016/0022-0000(88)90005-0

[14] Bruck, S., Jefferson, D., Rivest, R., August 26-29 2001. A modular voting architecture ("FROGS"). In: Proceedings of the Workshop on Trustworthy Elections (WOTE '01). pp. 97–106.
URL http://www.vote.caltech.edu/drupal/files/working_paper/vtp_wp3.pdf

[15] Brunazzo, A., Rezende, P., 2010. Security measures for brazil's e-vote – act two: Parallel testing.
URL http://www.cic.unb.br/~pedro/trabs/Brazilvote2.htm

[16] Camargo, V., 2005. Security in e-voting (Ph.D. Thesis). Department of Computer and Systems Sciences Royal Institute of Technology.

[17] Chaum, D., 1982. Blind signatures for untraceable payments. In: Proceedings of Advances in Cryptology (CRYPTO '82). pp. 199–203.

[18] Chaum, D., January 2004. Secret-ballot receipts: True voter-verifiable elections. IEEE Security and Privacy 2, 38–47.
URL http://portal.acm.org/citation.cfm?id=1435610.1436084

[19] Chaum, D., Carback, R. T., Clark, J., Essex, A., Popoveniuc, S., Rivest, R. L., Ryan, P. Y. A., Shen, E., Sherman, A. T., Vora, P. L., Dec. 2009. 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.

[20] Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., Vora, P., 2008. Scantegrity: End-to-end voter-verifiable optical-scan voting. IEEE Security and Privacy 6, 40–46.

[21] Chaum, D., Hosp, B., Popoveniuc, S., Vora, P. L., 2009. Accessible voter-verifiability. Cryptologia 33 (3), 283–291.

[22] Chaum, D., Ryan, P. Y. A., Schneider, S. A., 2005. A practical voter-verifiable election scheme. In: Proceedings of ESORICS'2005. pp. 118–139.

[23] Chaum, D. L., February 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM 24 (2), 84–90.
URL http://doi.acm.org/10.1145/358549.358563

[24] Clark, J., Essex, A., Adams, C., 2007. On the security of ballot receipts in e2e voting systems. In: Proceedings of the IAVoSS Workshop on Trustworthy Elections (WOTE '07). pp. –.
URL http://www.cs.uwaterloo.ca/~j5clark/papers/BallotReceipts.pdf

[25] Cohen, J. D., Fischer, M. J., 1985. A robust and verifiable cryptographically secure election scheme. In: Proceedings of Annual IEEE Symposium on Foundations of Computer Science (FOCS '85). IEEE Computer Society, Los Alamitos, CA, USA, pp. 372–382.

[26] County, S., 2010. Vote by mail: Voting historical background. Online June 2010.
URL http://stanvote.com/ballots.shtm#vote-by-mail

[27] Cross, E., Rogers, G., McClendon, J., Mitchell, W., Rouse, K., Gupta, P., Williams, P., Mkpong-Ruffin, I., McMillian, Y., Neely, E., Lane, J., Blunt, H., Gilbert, J., July 2007. Prime III: One machine, one vote for everyone. In: Proceedings of 2007 Voting Competition Conference. pp. –.
URL http://vocomp.org/papers/primeIII.pdf

[28] E-Voting.CC, Competence Center for Electronic Voting and Participation, 2009. Map of electronic democracy. Modern Democracy 2 (1), 8–9.
URL http://e-voting.cc/files/e-voting-map-2010

[29] El Gamal, T., 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. In: Proceedings of CRYPTO 84 on Advances in cryptology. Springer-Verlag New York, Inc., New York, NY, USA, pp. 10–18.

[30] Election Assistance Commission (USA), 2005. Voluntary voting system guidelines. Online February 2011.
URL http://www.eac.gov/assets/1/workflow_staging/Page/124.PDF

[31] Estonian National Electoral Committee, 2005. iVote. Online March 2010.
URL http://www.epractice.eu/en/cases/ivote

[32] Fisher, K., Carback, R., Sherman, A. T., May 2006. Punchscan: Introduction and system definition of a high-integrity election system. In: Proceedings of the Workshop on Trustworthy Elections 2006. pp. 1–8.
URL http://www.punchscan.org/papers/fisher_punchscan_wote2006.pd

[33] Gerlach, J., Gasser, U., 2009. Three case studies from switzerland: E-voting. Berkman Center Research Publication No. 2009-03.1.
URL http://cyber.law.harvard.edu/sites/cyber.law.harvard.edu/files/Gerlach-Gasser_SwissCases_Evoting.pdf

[34] Goldreich, O., Micali, S., Wigderson, A., 1987. How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design. In: Proceedings on Advances in cryptology—CRYPTO '86. Springer-Verlag, London, UK, pp. 171–185.
URL http://portal.acm.org/citation.cfm?id=36664.36675

[35] Henry, K., Stinson, D. R., Sui, J., 2009. The effectiveness of receipt-based attacks on ThreeBallot. Trans. Info. For. Sec. 4 (4), 699–707.

[36] Ibrahim, S., Kamat, M., Salleh, M., Aziz, S. R. A., January 2003. Secure e-voting with blind signature. In: 4th National Conference on Telecommunication Technology. pp. 193–197.
URL http://eprints.utm.my/3262/

[37] Kohno, T., Stubblefield, A., 2004. Analysis of an electronic voting system. In: Proceedings of the IEEE Symposium on Security and Privacy. pp. 27–40.

[38] Küsters, R., Truderung, T., Vogt, A., 2010. A game-based definition of coercion-resistance and its applications. In: Proceedings of the 23rd IEEE Computer Security Foundations Symposium. pp. 122–136.

[39] Küsters, R., Truderung, T., Vogt, A., 2010. Proving coercion-resistance of scantegrity II. In: Proceedings of the 12th international conference on Information and communications security - ICICS 2010. pp. 281–295.

[40] Madise, Ü., Vinkel, P., Maaten, E., 2006. Internet voting at the election of local government councils on october 2005.
URL http://www.vvk.ee/public/dok/report2006.pdf

[41] Mercuri, R., October 2000. Electronic vote tabulation checks and balances. Ph.D. thesis, University of Pennsylvania,School of Engineering and Applied Science, Department of Computer and Information Systems, Philadelphia, PA, USA, supervisor-Norman I. Badler.

[42] Monteiro, A., S. N., Oliveira, R., Antunes, P., 2001. Sistemas electrónicos de votação. Tech. rep., Departamento de Informatica Faculdade de Ciencias da Universidade de Lisboa.

[43] Moran, T., Naor, M., September 2006. Receipt-free universally-verifiable voting with everlasting privacy. In: Dwork, C. (Ed.), Proceedings of 26th International Cryptology Conference (CRYPTO '06). Vol. 4117 of Lecture Notes in Computer Science. Springer, pp. 373–392.
URL http://www.seas.harvard.edu/~talm/papers/MN06-voting.pdf

[44] Moran, T., Naor, M., 2007. Split-ballot voting: Everlasting privacy with distributed trust. ACM Trans. Inf. Syst. Secur. 13 (2), 1–43.

[45] Nakashima, E., 2006. E-mail voting comes with risks. Washington Post.
URL http://www.washingtonpost.com/wp-dyn/content/article/2006/10/30/AR2006103001062.html

[46] Neff, C. A., 2001. A verifiable secret shuffle and its application to e-voting. In: Proceedings of the 8th ACM conference on Computer and Communications Security (CCS '01). ACM, New York, NY, USA, pp. 116–125.

[47] Neff, C. A., 2004. Practical high certainty intent verification for encrypted votes.

[48] Norwegian Ministry of Local Government and Regional Development, November 2009. E-vote 2011: Accessibility and usability evaluation of e-vote prototypes. Online Feb. 2010.
URL http://www.regjeringen.no/upload/KRD/Kampanjer/valgportal/e-valg/e_valg_systemlosning/report_evoting_usability_accessibility_eval_nr_iter2_final.pdf

[49] Norwegian Ministry of Local Government and Regional Development, December 2009. E-vote 2011: Contractor solution specification. Online Feb. 2010.
URL http://www.regjeringen.no/upload/KRD/Kampanjer/valgportal/e-valg/e_valg_systemlosning/Tilbud_ergogroup/SSA-U_Appendix_2A_Contractor_Solution_Specification.pdf

[50] Norwegian Ministry of Local Government and Regional Development, February 2009. E-vote 2011: Project directive for e-valg 2011. Online Feb. 2010.
URL http://www.regjeringen.no/upload/KRD/Vedlegg/KOMM/Evalg/Project_directive_evalg2011_v101_english.pdf

[51] Norwegian Ministry of Local Government and Regional Development, October 2009. E-vote 2011: System requirements specification. Online Feb. 2010.
URL http://www.regjeringen.no/upload/KRD/Kampanjer/valgportal/e-valg/Anskaffelse/System_Requirements_Specification1.pdf

[52] of State Office of Voting Systems Technology Assessment, S., 2005. Voting-on-paper assistive device (Vote-PAD). Online June 2010.
URL http://www.vote-pad.us

[53] Öksüzoğlu, E., Wallach, D. S., 2009. Votebox nano: a smaller, stronger fpga-based voting machine. In: Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections. EVT/WOTE'09. USENIX Association, Berkeley, CA, USA, pp. 8–8.
URL http://portal.acm.org/citation.cfm?id=1855491.1855499

[54] Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT '99). Springer Verlag, pp. 223–238.
URL http://www.gemplus.com/smart/rd/publications/pdf/Pai99pai.pdf

[55] Pedersen, T. P., 1992. Non-interactive and information-theoretic secure verifiable secret sharing. In: CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology. Springer-Verlag, London, UK, pp. 129–140.

[56] Peng, K., 2009. A hybrid e-voting scheme. In: Proceedings of the 5th International Conference on Information Security Practice and Experience (ISPEC '09). Springer-Verlag, Berlin, Heidelberg, pp. 195–206.

[57] Peng, K., Aditya, R., Boyd, C., Dawson, E., Lee, B., 2004. Multiplicative homomorphic e-voting. In: Proceedings of 5th International Conference on Cryptology in India (INDOCRYPT '04). Springer, pp. 61–72.
URL http://www.springerlink.com/content/5a6d6d0jaxury5aq

[58] Peng, K., Bao, F., 2009. A design of secure preferential e-voting. In: Proceedings of the 2nd International Conference on E-Voting and Identity (VOTE-ID '09). Springer-Verlag, Berlin, Heidelberg, pp. 141–156.

[59] Puiggali, J., Morales-Rocha, V., 2007. Independent voter verifiability for remote electronic voting. In: Proceedings of International Conference on Security and Cryptorgraphy (SECRYPT '07). Springer Verlag, pp. 333–336.

[60] Qvortrup, M., 2005. First past the postman: Voting by mail in comparative perspective. The Political Quarterly 76 (3), 414–419.

[61] Rezende, P., 2010. Electronic elections: A balancing act. In: Towards Trustworthy Elections. Vol. 6000 of Lecture Notes in Computer Science. Springer, pp. 124–140.

[62] Riebeek, H., 2002. Brazil holds all-electronic national election. IEEE Spectrum 39 (11), 25–26.

[63] Rivest, R. L., January 2006. The ThreeBallot voting system, unpublished draft.
URL http://people.csail.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf

[64] Rivest, R. L., Smith, W. D., 2007. Three voting protocols: ThreeBallot, VAV, and Twin. In: Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology (EVT '07). USENIX Association, Berkeley, CA, USA.

[65] Rosenberg, B., 2011. Handbook of Financial Cryptography and Security. Chapman & Hall/CRC.

[66] Runyan, N., 2007. Improving Access to Voting: A Report on the Technology for Accessible Voting Systems. Voter action & Demos.

[67] Ryan, P. Y. A., 2005. A variant of the chaum voter-verifiable scheme. In: Proceedings of the 2005 workshop on Issues in the theory of security – WITS'05. ACM, pp. 81–88.

[68] Ryan, P. Y. A., June 2006. Verified encrypted paper audit trails. Technical report CS-TR-966, School of Computing Science, Newcastle University.

[69] Ryan, P. Y. A., Peacock, T., September 2005. Prêt à voter: a systems perspective. Technical report cs-tr-929, School of Computing Science, Newcastle University.
URL http://www.cs.newcastle.ac.uk/publications/trs/papers/929.pdf

[70] Ryan, P. Y. A., Schneider, S. A., Ryan, P. Y. A., Schneider, S. A., 2006. prêt à voter with re-encryption mixes. In: 11th European Symposium on Research in Computer Security – ESORICS'06.

[71] Sandler, D., Derr, K., Wallach, D. S., 2008. Votebox: a tamper-evident, verifiable electronic voting system. In: Proceedings of the 17th conference on Security symposium (SS '08). USENIX Association, Berkeley, CA, USA, pp. 349–364.
URL http://www.cs.rice.edu/~dsandler/pub/sandler08votebox.pdf

[72] Sandler, D. R., April 2009. Votebox: A tamper-evident, verifiable voting machine. Ph.D. thesis, Rice University.

[73] Santin, A. O., Costa, R. G., Maziero, C. A., 2008. A Three-Ballot-based secure electronic voting system. IEEE Security and Privacy 6 (3), 14–21.

[74] Schryen, G., Rich, E., 2009. Security in large-scale internet elections: a retrospective analysis of elections in estonia, the netherlands, and switzerland. Trans. Info. For. Sec. 4 (4), 729–744.

[75] Schumer, C. E., July 15 2009. Schumer, Chambliss and Nelson announce key senate committee unanimously approves bipartisan bill to make voting easier for military voters. Online June 2010.
URL http://schumer.senate.gov/new_website/record.cfm?id=315837

[76] Scytl Online World Security S. A., December 2004. Auditability and voter-verifiability for electronic voting terminals. Online April 2010.
URL http://www.scytl.com/_a_home/PNYX.VM_White_Paper.pdf

[77] Selker, T., September 2004. The voter verified audio audit transcript trail. Online Feb. 2010.
URL http://www.dos.state.pa.us/election_reform/lib/election_reform/VVAATT_CalTech.pdf

[78] Selker, T., Cohen, S., May 2005. An active approach to voting verification. Online Feb. 2010.
URL http://www.vote.caltech.edu/drupal/files/working_paper/vtp_wp28.pdf

[79] Shamir, A., 1979. How to share a secret. Commun. ACM 22 (11), 612–613.

[80] Sherman, A. T., Gangopadhyay, A., Holden, S. H., Karabatis, G., Koru, A. G., Law, C. M., Norris, D. F., Pinkston, J., Sears, A., Zhang, D., 2006. An examination of vote verification technologies: findings and experiences from the maryland study. In: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop (EVT '06). USENIX Association, Berkeley, CA, USA, pp. 10–10.

[81] Storer, T., Little, L., October 2005. Electronic voting in the uk: Current trends in deployment, requirements and technologies. In: Ghorbani, A., Marsh, S. (Eds.), Proceedings of the Third Annual Conference on Privacy, Security and Trust. pp. 249–252.

[82] Tjostheim, T., Peacock, T., Ryan, P., 2007. A case study in system-based analysis: The three ballot voting system and prêt à voter. In: Proceedings of VoComp.

[83] Van De Graaf, J., 2009. Voting with unconditional privacy by merging Prêt à Voter and PunchScan. Trans. Info. For. Sec. 4 (4), 674–684.

[84] Varner, P. E., March 2001. Vote early, vote often, and VoteHere: A security analysis of VoteHere. Ph.D. thesis, University of Virginia.
URL `http://www.cs.virginia.edu/~evans/theses/varner.pdf`

[85] Villafiorita, A., Weldemariam, K., Tiella, R., 2009. Development, formal verification, and evaluation of an e-voting system with vvpat. Trans. Info. For. Sec. 4 (4), 651–661.

[86] Xia, Z., Culnane, C., Heather, J., Jonker, H., Ryan, P., Schneider, S., Srinivasan, S., 2010. Versatile prêt à voter: Handling multiple election methods with a unified interface. In: Proceedings of the 11th International Conference on Cryptology in India - Progress in Cryptology - INDOCRYPT 2010. pp. 98–114.