Anonymity and Verifiability in Voting: Understanding (Un)Linkability

Lucie Langer[1], Hugo Jonker[2], and Wolter Pieters[3]

[1] Technische Universität Darmstadt, Cryptography and Computer Algebra Group
[2] University of Luxembourg, Faculty of Science, Technology and Communication
[3] Centre for Telematics and Information Technology, University of Twente
langer@cdc.informatik.tu-darmstadt.de, hugo.jonker@uni.lu,
w.pieters@utwente.nl

Abstract. Anonymity and verifiability are crucial security requirements for voting. Still, they seem to be contradictory, and confusion exists about their precise meanings and compatibility. In this paper, we resolve the confusion by showing that both can be expressed in terms of (un)linkability: while anonymity requires unlinkability of voter and vote, verifiability requires linkability of voters and election result. We first provide a conceptual model which captures anonymity as well as verifiability. Second, we express the semantics of (un)linkability in terms of (in)distinguishability. Third, we provide an adversary model that describes which capabilities the attacker has for establishing links. These components form a comprehensive model for describing and analyzing voting system security. In a case study we use our model to analyze the security of the voting scheme Prêt à Voter. Our work contributes to a deeper understanding of anonymity and verifiability and their correlation in voting.

Keywords: anonymity, verifiability, unlinkability, e-voting, adversary model.

1 Introduction

Correctness of the election procedure and freedom to vote are fundamental requirements for elections. If the voters cannot be assured that the election procedure has been followed correctly, they have no reason to trust the result, and hence no incentive to vote. Similarly, if the voters cannot be guaranteed freedom to vote, that is, if they cannot be assured that their votes cannot be interfered with, and that their votes shall not give rise to repercussions, they have no reason to express their true intentions. Absence of either assurance thus results in an empty, meaningless exercise instead of a democratic process. Therefore, provisions have been developed to provide these assurances. Correctness is assured by means of verifiability: a voter can verify that her vote affects the election result as she intended, and that the result is comprised only of votes cast by eligible voters, all handled correctly. Freedom is assured by means of anonymity:

M. Soriano, S. Qing, and J. López (Eds.): ICICS 2010, LNCS 6476, pp. 296–310, 2010.

the system ensures that the voter's preference is not revealed to anyone. Thus, verifiability ensures a voter can link her vote to the result, while anonymity ensures no one can link a voter to her preference.

Yet, there is an apparent contradiction between the verifiability and anonymity, since verifiability seems to require traceability of the votes through the system, whereas anonymity is at odds with precisely this concept. Confusion is added because, in literature, there are opposing results. For example, [7] proves incompatibility of these notions, while other works such as [6,13], do combine notions of verifiability and anonymity.

To understand how such seemingly contradictory results can hold, a thorough understanding of both anonymity and verifiability in voting is necessary. We alleviate the confusion by showing that both these fundamental security properties can be expressed as properties of (the set of) links between objects and entities in voting systems, although these links are different for verifiability and anonymity. To this end, this paper introduces a comprehensive security model for voting. It consists of:

1. an (un)linkability model (introduced in Sect. 3),
2. semantics in terms of (in)distinguishability (Sect. 4), and
3. components for constructing adversary models (Sect. 5).

The first two components provide a deeper understanding of anonymity and verifiability and their correlation in electronic voting, and thereby improve possibilities for reasoning on the security of voting systems. The third component allows to determine reasonable adversary models for individual election scenarios. All three components together can be used to evaluate the security of voting schemes by considering the (un)linkability provided in light of the adversary capabilities assumed, which we have done for the voting scheme Prêt à Voter. This case study is provided in Sect. 6.

2 Background and Motivation

This section discusses the background on verifiability and anonymity in voting, in order to define precisely the concepts that we wish to study.

In voting, verifiability has traditionally been split into two notions:

> **Individual Verifiability.** The voter can verify that her vote affects the result correctly.
> **Universal Verifiability.** Anyone can verify that the announced result is the correct accumulation of the individual votes.

Together, they assure that the result includes all votes correctly. The literature is divided as to whether universal verifiability comprises verifying eligibility, i.e. the fact that only eligible voters cast a vote (cf. [1,7]). In [20] eligibility verifiability was addressed first as a separate voting requirement.

Distinguishing between individual and universal verifiability has shortcomings, as pointed out by [5]. This work gave rise to the domain of end-to-end verifiability, which distinguishes the following types of verifiability (cf. [1,18]):

Cast-as-intended. The voter can verify that her ballot correctly represents her preference.
Counted-as-cast. The voter can verify that her vote counts "as cast", that is, in favor of the candidate that she voted for. This is sometimes split into the following notions:

Recorded-as-cast. The voter can verify that her vote is stored by the system as she cast it.
Counted-as-recorded. Anyone can verify that the announced result is a correct amalgamation of the set of recorded votes.

The combination of cast-as-intended and recorded-as-cast is named *ballot casting assurance* in [1]. These notions enable a voter to follow her vote from input to the final result. However, they have been interpreted in different ways to date: counted-as-cast has, for example, also been said to allow *anyone* to verify that the final tally is an accurate count of the ballots cast [16], or to allow any voter to verify that *all* votes are counted as cast [3].

As remarked, anonymity has always been seen as essential to voting. Using techniques such as anonymous channels, blind signatures, and homomorphic tallying, classical voting systems such as [10] ensure that no one can learn how a voter voted. However, over the years the notion of anonymity has been further refined into the following forms:

Privacy. Privacy ensures that no observer learns how a voter voted.
Receipt-freeness. Introduced by [2], receipt-freeness ensures that the voter is forced to keep her vote private, even if she would like to share it. It makes sure that the voter cannot prove how she voted *after* the elections, thus preventing vote buying.
Coercion-resistance. Introduced by [15], coercion-resistance ensures receipt-freeness and resistance to the following attacks:

- Randomization: the voter is forced to vote for a random candidate.
- Simulation: the voter is forced to give her voting credentials to the adversary, who then votes in her stead.
- Forced abstention: the voter is forced not to vote.

Note that this definition of receipt-freeness does not necessarily imply the above definition of privacy, unlike others in literature (e.g. [8]). Moreover, according to [8,17], coercion-resistance implies receipt-freeness, while [4] claims that it is possible to have a voting scheme which is coercion-resistant and yet not receipt-free. These contradictory assertions show that the concept of anonymity requires clarification as well. In the following, we use "anonymity" as a general concept, while "privacy", "receipt-freeness" and "coercion-resistance" are specific instantiations of this concept in voting.

Anonymity and verifiability seem at odds with each other. In [7], Chevallier et al. prove that unconditional privacy and universal verifiability cannot be simultaneously achieved without additional assumptions (such as private channels). Most schemes in literature make such assumptions, and under these may

be able to achieve both verifiability and anonymity. However, the precise relation between the two concepts within a single system is not yet thoroughly understood.

To determine if a claim of anonymity or verifiability is valid, the voting scheme is considered in the presence of an adversary. Various abilities may be attributed to the adversary, depending on his anticipated strength. Examples of such abilities are breaking cryptography and full control over communications. The adversary models used to evaluate electronic voting schemes are usually tailored to the specific scenario given by the respective voting protocol (see for example [15]). A general adversary model which would allow to evaluate several voting schemes (and, thus, to compare them with each other) has not yet been provided. One approach would be to take the adversary model proposed by Dolev and Yao [9]. However, as we will see in Sect. 5, this model has certain drawbacks and, depending on the specific use case, can either be too strong or even too weak.

3 (Un)Linkability Model

Section 2 has shown that anonymity and verifiability and their correlation in voting are not fully understood yet. In this section, we present the first part of our solution in terms of an (un)linkability model for voting which captures both properties.

Terminology and notation. Any election can be described as follows: each **voter** prefers a certain **candidate**[1] and expresses this preference via her **vote**. The vote is input to the voting system via the **ballot**, which represents the vote and conceals it at the same time. Note that this description applies to both paper ballot systems and e-voting systems. While the paper ballot usually is an envelope containing the vote, the electronic ballot conceals the vote for example by means of cryptography. Entities involved in the process include real-life *persons* (voters $v \in \mathcal{V}$, candidates $c \in \mathcal{C}$) and *objects* (votes/options $o \in \mathcal{O}$, ballots $b \in \mathcal{B}$) belonging to the voting system.

From this general description, we observe that any election can be modelled by a set of *links*. These include the links voter–vote, vote–ballot, and ballot–candidate. This enables us to express properties in terms of linkability. The links between persons and objects are captured as follows:

- $\beta\colon \mathcal{V} \to \mathcal{B}$ links a voter v to her ballot b.
- $\tau\colon \mathcal{B} \to \mathcal{O}$ links a ballot b to the vote o contained therein.
- $\pi\colon \mathcal{O} \to \mathcal{C}$ links a vote o to the selected candidate c.
- $\omega\colon \mathcal{V} \to \mathcal{O}$ links a voter v to her vote o (note that $\omega = \tau \circ \beta$).
- $\gamma\colon \mathcal{V} \to \mathcal{C}$ links a voter v to her preferred candidate[2] c (note that $\gamma = \pi \circ \omega = \pi \circ \tau \circ \beta$).

[1] Spoiling one's vote can be modelled by voting for an empty candidate.

[2] The modelling can be extended to encompass Single Transferable Votes (STVs) by having each of the possible orderings of candidates constitute one option that the voter can vote for (i.e., a "candidate" in the system).

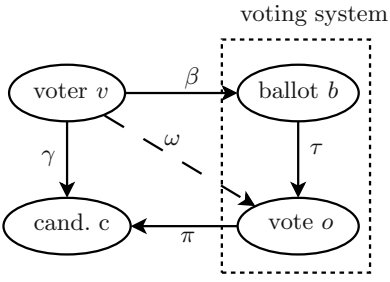

Fig. 1. Individual-related model

3.1 Anonymity as Unlinkability

Since anonymity concerns individuals, we consider an individual-related model (see Fig. 1). This model considers individual entities and mappings between them as introduced above: voter v casts a ballot $b = \beta(v)$ containing the vote $o = \tau(b)$ which refers to candidate $c = \pi(o)$.

Anonymity requires unlinkability of voter v and candidate $c = \gamma(v)$, i.e. this link must remain secret. We assume that there is no direct link in the system between the voter and her preferred candidate since such a link would not be under the control of the voting system (imagine, for example, each voter standing next to the candidate she prefers). Thus, in practice we always have the decomposition $\gamma = \pi \circ \tau \circ \beta$. The function $\pi\colon \mathcal{O} \to \mathcal{C}$ is assumed to be public. Thus, anonymity can be broken down to unlinkability of voter v and vote $\omega(v)$ (depicted by a dashed line in Fig. 1), which due to $\omega = \tau \circ \beta$ can be established in two ways: unlinkability of voter v and ballot $\beta(v)$, or unlinkability of ballot b and vote $\tau(b)$. For example, blind signatures and mix-nets can be used to conceal the link between voter and ballot, while unlinkability of ballot and vote is provided in homomorphic schemes where an individual ballot is never decrypted. In conclusion, anonymity can be expressed as unlinkability of individual voters, ballots and votes.

3.2 Verifiability as Linkability

As (universal) verifiability concerns *groups* of individuals and *sets* of votes/ballots, we extend our individual-related model to a *set-related* model, see Fig. 2. The set of all received ballots $B(\mathcal{V})$ is given by

$$B(\mathcal{V}) = \{b \in \mathcal{B} \mid \exists v \in \mathcal{V}\colon \beta(v) = b\}\,,$$

where we assume that all ballots are unique (otherwise, ballot duplication (e.g. by replay attacks) would be easy). Similarly, $\Omega(\mathcal{V})$ denotes the set of all votes that have been cast. In order to express this for homomorphic tallying as well, where individual votes are never decrypted, we define $\Omega(\mathcal{V})$ as a multiset of all cast votes (where each cast vote originates from a ballot), i.e.

$$\Omega(\mathcal{V}) = \{(o, n) \in \mathcal{O} \times \mathbb{N} \mid \exists b \in \mathcal{B}\colon \tau(b) = o \wedge n = (\#b \in \mathcal{B}\colon \tau(b) = o)\}\,.$$

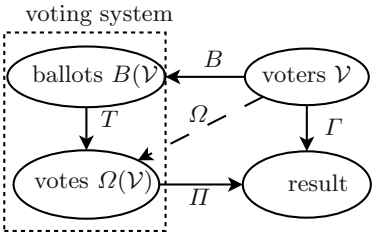

Fig. 2. Set-related model, relating all voters to all ballots, all votes and the final result

This can be publicly transformed into the election result (e.g. the number of seats for each party). Using the definition of $B(\mathcal{V})$ and $\tau \circ \beta = \omega$, we also have

$$\Omega(\mathcal{V}) = \{(o, n) \in \mathcal{O} \times \mathbb{N} \mid \exists v \in \mathcal{V}\colon \omega(v) = o \wedge n = (\#v \in \mathcal{V}\colon \omega(v) = o)\}\,.$$

Remark that we do not assume uniqueness of the votes: votes for the same candidate will usually[3] have the same form. The election result, e.g. the number of seats held by different parties in parliament, is obtained by a public transformation Π of the set of all votes $\Omega(\mathcal{V})$.

First we consider individual verifiability, requiring that each voter can verify that her ballot correctly captures her intended vote and has been included in the set of ballots to be counted. This is established by the following links:

1. the voter v can identify her ballot $\beta(v)$ (*cast-by-me*)
2. the ballot b contains the correct vote $\tau(b)$ (*contains-correct-vote*)
3. the ballot b is contained in the set of received ballots $B(\mathcal{V})$ (*recorded-as-cast*)

Following the concepts *cast-as-intended*, *counted-as-cast* and similar (see Sect. 2), we have named the first two links *cast-by-me* and *contains-correct-vote*, respectively. Both together correspond to the concept of *cast-as-intended*. Our third link matches the established notion of *recorded-as-cast*. Note that our definition of individual verifiability matches the concept of *ballot casting assurance* (see Sect. 2). By the link between the vote o and the set of all votes $\Omega(\mathcal{V})$, the voter knows that her vote is included in the tally. Still, the voter cannot pinpoint her vote as we assume that the votes are not unique.

Universal verifiability, on the other hand, requires the public to be able to verify that all received ballots have been counted correctly. This is established by the link between the set of received ballots $B(\mathcal{V})$ and the set of cast votes $\Omega(\mathcal{V})$, which matches the concept of *counted-as-recorded*. *Eligibility verifiability*, requiring anyone to be able to verify that only eligible voters cast a vote [20], is expressed by the link between the set of received ballots $\beta(\mathcal{V})$ and the set of voters $\mathcal{V}$. Public verifiability, thus, comprises linkability of all voters with the set of received ballots (eligibility verifiability) and linkability of received ballots and counted votes (universal verifiability or counted-as-recorded). Verifiability can thus be expressed as linkability of sets of voters, ballots and votes.

[3] This is not necessarily the case, e.g. in ThreeBallot [18] or in single transferable voting (STV).

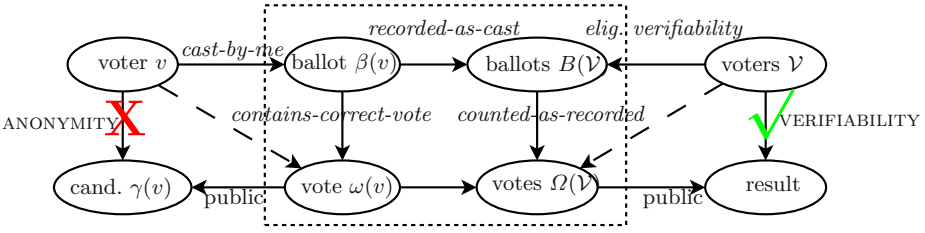

Fig. 3. (Un)linkability model

3.3 Unified (Un)Linkability Model

The individual-related and the set-related model are merged as shown in Fig. 3. This unified model captures VERIFIABILITY (desired linkability) as well as ANONYMITY (desired unlinkability) as follows. Since the link between the result and the set of all votes is public, verifiability is expressed by linkability of the set of all voters and the result (depicted with a "√" in Fig. 3). With respect to anonymity, we require unlinkability of a voter and her preferred candidate (depicted with an "X" in Fig. 3). The distinction between unlinkability of voters and candidates in the individual-related model and linkability of all voters with the election result in the set-related perspective is the key strength of our model. It explains how anonymity and verifiability can be combined in voting systems. Although there is an obvious trade-off between anonymity and verifiability, in our (un)linkability model, both are expressed by the same link in the individual-related and the set-related scenario, respectively, which shows the close relation between anonymity and verifiability.

The above model intuitively captures both anonymity and verifiability in terms of (un)linkability. It enables defining the notions *cast-as-intended*, *counted-as-cast*, *recorded-as-cast* and *counted-as-recorded* unambiguously. We also introduced the notions *cast-by-me* and *contains-correct-vote* to specify the term *cast-as-intended*, which has been interpreted in different ways to date. We now move on to the semantics of linkability and unlinkability.

4 Semantics of (Un)Linkability in Terms of (In)distinguishability

What does it mean that certain events in the voting system are (un)linkable? We argue that distinguishability is the natural concept to provide a semantics for the unlinkability properties. To explain this, we need the notions of a voting *protocol*, which specifies which types of messages should be exchanged by the various involved parties, and a protocol *run*, which is the instantiation of the protocol in a particular election. For reasons of clarity, we denote runs informally as sequences of messages $[m_1, m_2, m_3]$. Distinguishability then amounts to the observer's ability to spot differences between two runs. In particular, these often

are the actual run of the election and a hypothetical, slightly altered one. A formal definition of (in)distinguishability can be found in [11].

Intuitively, a voter should be able to distinguish between a run where her vote is counted correctly and a run where her vote is counted incorrectly (cast a, count a versus cast a, count b). This is a verifiability property. However, an attacker should not be able to distinguish between a run where the voter votes a and a run where the voter votes b (cast a, count a versus cast b, count b). This is an anonymity property. In the following, we give more precise descriptions for these properties. A piece of information e_1 from a protocol run is *linkable* to a piece of information e_2 for agent A if A can distinguish between a run containing e_1 and e_2, and a run containing e_1 and e_2'. For example, consider the linkability of a voter v and her vote $\omega(v)$. The *linkability of the voter to the vote* states that a run $[\ldots, v, \ldots, \omega(v), \ldots]$ is distinguishable from a run where the vote is modified $[\ldots, v, \ldots, \omega'(v), \ldots]$. From the perspective of the voter, it means that only *this* vote can be hers. Vice versa, the *linkability of the vote to the voter* states that a run $[\ldots, v, \ldots, \omega(v), \ldots]$ is distinguishable from a run where the vote was cast by another voter $[\ldots, v', \ldots, \omega(v), \ldots]$. From the perspective of the voter, it means that this vote can only be *hers*.

These distinguishability properties establish verifiability from the voter's perspective. Depending on the requirements of a particular system, the distinguishability should either hold only from the perspective of the voter, or from the perspective of anyone. For reasons of anonymity, we certainly do *not* want anyone to be able to distinguish the situations where a voter casts different votes. This leads to a desirable form of unlinkability. However, we cannot account for situations where a particular candidate does not receive any votes from others, as this would make directly observable whether the voter votes for this candidate. Therefore, it should be indistinguishable to the attacker whether two voters *swap* votes, i.e. the attacker cannot distinguish between a run where voter v_1 votes $\omega(v_1)$ and voter v_2 votes $\omega(v_2)$, and a run where voter v_1 votes $\omega(v_2)$ and voter v_2 votes $\omega(v_1)$ (cf. [8]).

Similar definitions can be derived for the set-related perspective. Using this semantics, it becomes possible to relate linkability to the observation capabilities of the attacker. In the next section, we will discuss such capabilities. Any formal protocol semantics can be used to formalize these distinguishability properties and reason about them, which is beyond the scope of this paper.

5 Components for Setting-Specific Adversary Models

As mentioned in Sect. 2, voting systems are considered in the presence of an adversary with specific capabilities. Thus, in order to apply our (un)linkability model and its semantics to existing voting protocols, we need an adversary model. However, a set adversary model (such as the Dolev-Yao model [9]) can be too strong or too weak for a specific situation.

For example, in an employees council election the CEO may have an interest to influence the results, but is not allowed inside the election office. Similarly, in a

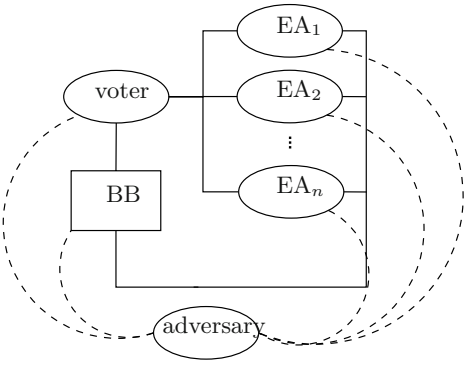

Fig. 4. Adversary communication model

national election it may be of interest to determine the impact of a single person with limited power, not an adversary who can control *all* communications. On the other hand, cryptographic schemes may turn out to be broken, and messages encrypted with such schemes may (eventually) be decrypted without using the decryption key (a particular risk for e-voting, as often, sensitive data is published to provide verifiability). None of these scenarios are adequately captured by the Dolev-Yao model, nor can they all be captured by any one adversary model.

Instead of adhering to one adversary model, we decompose the adversary's capabilities into specific components. With these components, one can easily define a fine-tuned adversary model for specific use cases.

Setting. A generic e-voting system (see Fig. 4) consists of voters and several election authorities (EAs). The voters are registered by the EAs and subsequently cast ballots which are processed by (possibly different) EAs. A public broadcast channel (bulletin board, BB) is used by the voters and the EAs in order to post messages for reasons of verifiability. Additionally, there are communication channels between each voter and the EAs, which may be cryptographically secured (e.g. by encryption, signing, blinding, etc.). We assume that the EAs communicate via the BB.

Adversary Capabilities. In addition to communication abilities (based on the communication model above, and inspired by the capabilities of a Dolev-Yao intruder), we consider cryptographic abilities. Furthermore, we distinguish between the adversary's abilities concerning existing communication channels, and his ability to create new communication channels (represented by dashed lines in Fig. 4). Thus, the capabilities of an adversary can be divied as follows:

I. capabilities concerning existing communication channels,
II. capability wrt. new communication channels,
III. cryptographic capabilities.

I. Capabilities Concerning Existing Communication Channels. We distinguish the following adversary capabilities regarding the ways in which the adversary can affect existing communication channels:

Ia. The adversary can detect channel usage.
Ib. The adversary can determine the sender of a message.
Ic. The adversary can eavesdrop on communication channels.
Id. The adversary can block communication channels.
Ie. The adversary can inject messages into communication channels.
If. The adversary can modify messages sent over communication channels.

An untappable channel provides perfect anonymity [15] and thus protects against an adversary with any of the above capabilities. An anonymous channel protects against an adversary with capability Ib. When analyzing voting schemes in light of the adversary capabilities, Ib and Ia-f shall not be considered for channels which are assumed to be anonymous or untappable, respectively.

II. Capabilities Concerning New Communication Channels. For the second category, we consider the following adversary capabilities:

IIa. The voter can send messages to the adversary.
IIb. An EA can send messages to the adversary.
IIc. The adversary can send messages to a voter.
IId. The adversary can send messages to an EA.
IIe. The adversary can post messages to the BB.

By repeated use of capabilities IIa-d, the adversary establishes one-way or two-way communications with groups of voters and/or groups of EAs. Capabilities IIa and IIb model voters or, respectively, EAs who cooperate with the adversary by leaking secrets (e.g. receipts proving a voter's vote), whereas IIc and IId model an adversary who coerces EAs or voters (e.g. by furnishing the voter with voting material in order to cast a specific vote), cf. [17].

III. Cryptographic Capabilities. For the third category, we do not assume that cryptography works perfectly (as opposed to the Dolev-Yao model [9]):

IIIa. The adversary can break cryptography providing computational security.

A voting scheme using cryptographic algorithms which provide information-theoretic security is secure even against an adversary with capability IIIa.

The attacker model determines which messages an attacker can observe and/or alter, and thereby which distinguishability properties hold. As we have seen in the previous section, these distinguishability properties determine in turn which messages are linkable for the attacker. Combining these three ingredients, real-life voting systems can be analysed in terms of (un)linkability, with respect to the combination of anonymity and verifiability.

Bob	
Chris	
Dora	X
Alice	
	fY92w7k

Fig. 5. Marked ballot form in Prêt à Voter

6 Case Study: (Un)Linkability of Prêt à Voter

The properties expressed as (un)linkability, the semantics of (in)distinguishability and the adversary model together form our comprehensive model of voting security. In this section, we analyze (un)linkability of the state-of-the-art voting scheme Prêt à Voter (PaV) [19].

PaV was originally developed by Ryan [19] and since then extended often. We analyze the version of [6]. The participants are voters,an election authority EA_1, and k tellers $EA_2, \ldots, EA_{k+1}$. Before the election, EA_1 generates the paper ballot forms consisting of two columns: the left column contains a candidate list determined by a cyclic offset from the base candidate ordering; the right column holds a random value at the bottom, the onion, which buries the information necessary for reconstructing the candidate ordering (see Fig. 5). To cast a vote, the voter registers at the polling station and randomly selects a ballot form. In the booth she marks the right column and then tears off the left one and shreds it. The right column is scanned and the image is send to the BB after the end of the election. Each teller performs an anonymizing mix and decryption by subsequently operating on the onions.

6.1 (Un)Linkability Analysis

First we consider verifiability. The correct processing of the ballots in the tallying phase can be publicly verified by randomized partial checking [14]. This establishes (probabilistic) linkability of the set of received ballots and the set of all votes (counted-as-recorded).[4] PaV does not provide a means to verify eligibility.

The voter can visit the BB to check that her receipt is correctly posted and hence correctly entered into the tallying process (recorded-as-cast). This means that the voter can distinguish her ballot from a ballot cast by another voter, which establishes linkability of ballot and voter (cast-by-me). However, the voter does not obtain any proof that the onion matches the candidate order in the left column and thus will be decrypted to the vote she intended to cast. Hence, linkability of ballot and vote (contains-correct-vote) is not provided.

[4] We refer to linkability of *sets* rather than individual ballots and votes, for which unlinkability under randomized partial checking has been proven in [12].

PaV allows for pre-election auditing[5] by revealing the construction of selected ballot forms. Thus, anyone can compute the onion as well as the offset for the candidate ordering and thereby verify that the ballot form was prepared correctly. This establishes a link between the ballot form and the set of candidates, in particular: the ballot's candidate ordering.

In PaV, the link between a voter and her ballot is ensured by the receipt; the link from one ballot to the set of ballots by the ballot box; the link from the set of ballots to the set of votes by the mixnet; the link from the vote to candidates by decryption after mixing, which also serves to link the set of votes to the result. Note that there is no link from cast ballots $\beta(v)$ to votes $\omega(v)$ – this link is hidden by the mixnet. The link from the received ballots $B(\mathcal{V})$ to the set of votes $\Omega(\mathcal{V})$ is probabilistic (as mentioned above). Finally, note the absence of a link between the set of all voters $\mathcal{V}$ and the set of all ballots $B(\mathcal{V})$, as PaV does not provide eligibility verifiability.

Next, we consider the different adversary capabilities, thus showing how our model can be used to analyze the security of PaV in terms of (un)linkability.

Capabilities Concerning Existing Communication Channels. Note that cast ballots do not contain any personal information of the voter. Also, individual ballots cannot be linked to the contained votes due to the mixing. If the adversary is restricted to public information, PaV thus provides unlinkability of voter and ballot and unlinkability of ballot and vote.

Consider an adversary attacking the communication channel from the voter (scanner) to the BB. Capability Ia allows the adversary to determine that the voter casts a ballot (but not what is on the ballot). Capability Ib enables the adversary to determine which ballot the voter cast (and so determine the validity of a vote). Using Ic, the adversary learns the link between the voter and her ballot. He can distinguish a) between this voter casting the ballot and another voter casting the ballot, and b) between the voter casting this ballot and the voter casting another ballot. The link thus works in both directions. Id enables the adversary to suppress ballots from being posted to the BB. This attack is detected if the voter checks the BB for her vote. With capability Ie, the adversary can post illegal ballots to the BB. An adversary with capability If is able to modify ballots sent to the BB. As for Id, such attacks can be detected by the voter.

Capabilities Concerning New Communication Channels. The ballots that are input to the voting system do not contain any personal information of the voter. However, if the voter forwards her receipt (IIa), the adversary can distinguish between the voter casting this ballot and the voter casting another ballot, and, thus, can link this voter to her ballot. Using capability IIb, ballot and vote can be linked: either EA_1 reveals the association between the candidate

[5] While auditing concerns linkability, it concerns ballots that are *not* cast, as opposed to verifiability (which concerns ballots that are cast). As the (un)linkability model does not distinguish one ballot from another, it cannot make this distinction. For completeness sake, we nevertheless address the linkability of audited ballots.

Table 1. Security of Prêt à Voter in terms of (un)linkability

capability	effects
Ia	determine existence of link voter → ballot
Ib	reveal link voter → ballot
Ic	reveal link: voter → ballot and ballot → voter
Id	ballot suppressed (detectable by voter)
Ie	eligibility compromised
If	ballot modified (detectable by voter)
IIa	reveal link: voter → ballot
IIb	reveal link: ballot → vote (for *all* ballots)
IIc	reveal link: voter → vote (chain voting attack)
IId	reveal link: ballot → vote (for *all* ballots)
IIe	eligibility compromised
IIIa	reveal link: ballot → vote (for *all* ballots)

ordering and the onion, or each teller $EA_2, \ldots, EA_{k+1}$ reveals his private key. An adversary equipped with capability IIc can furnish the voter with a marked ballot (obtained, for example, from EA_1, IIb) before she enters the polling station and coerce her to hand back an unmarked ballot form (IIa), thus proving that she cast the ballot provided by the adversary and linking her to the according vote (chain voting attack). If the adversary can send messages to an EA (IId) prior to the election, he can furnish EA_1 with the secret values for generating the ballot forms. Thus, the adversary knows the link between each ballot and each vote. Using IIe, the adversary can compromise eligibility by sending ballots to the BB as for Ie.

Cryptographic Capabilities. If the adversary can break the preimage resistance of the hash function (IIIa), he can trace back votes through the mix-net (cf. [6]), thus establishing a link between ballot and vote. If the adversary can break the encryption (IIIa), the ballot transformation is revealed and the adversary learns the link between each ballot and the corresponding vote.

(Un)Linkability of Prêt à Voter. We have seen that, depending on the adversary capabilities, different forms of (un)linkability are provided. The capabilities of category I can be used to establish a link between voter and ballot in PaV, thus anonymity is hurt in the presence of such an adversary. The capabilities of category III can be used to link voter and vote, thus destroying anonymity. For category II capabilities, both kinds of attacks are possible. Moreover, using either IIb, IId or IIIa, anonymity in PaV is broken, as each of these enables the adversary to uncover ithe link between ballot and vote for all ballots. The security of PaV in terms of (un)linkability is summarized in Table 1.

PaV offers linkability of the set of received ballots and the set of all votes (universal verifiability), but linkability of ballot and vote (contains-correct-vote) is provided only for *uncast* ballots. This approach assures the voter that her

actual vote will be correct without providing her with a receipt that could be used to prove it. As this approach does not reveal this link, this approach is far better suited for reconciling verifiability and anonymity than the traditional approach of having the voter check the correct form of her *cast* ballot.

7 Conclusion

In this paper, we resolved the confusion on the combination of verifiability and anonymity in voting systems by providing a comprehensive and general model of voting security. The model consists of:

1. an (un)linkability model capturing both anonymity and verifiability;
2. (informal) semantics of linkability in terms of distinguishability;
3. components for adversary models which describe the capabilities an attacker has for distinguishing and determining links.

The value of the (un)linkability model lies in the unification of seemingly different properties under a common terminology, enabling a clear visual representation of desirable and undesirable properties (see Fig. 3). The adversary components enable analysing and designing voting systems for particular environments, where the adversary capabilities deviate from regular assumptions.

Taken together, this model provides a unified approach to assess the security of voting schemes: it can be used for analyzing the level of anonymity and verifiability provided depending on the adversary capabilities assumed. This has been demonstrated with the Prêt à Voter system in a case study. Thus, different voting schemes can be compared in an intuitive, informal, yet precise way.

We recommend future work on formalizing the semantics, in order to enable automatic verification of properties. Also, additional case studies can lead to a visual representation of the differences between voting systems in terms of linkability properties.

Acknowledgements. The authors are grateful to the Center for Advanced Security Research Darmstadt (CASED) for supporting this collaboration. The research of the third author is supported by the research program Sentinels (`www.sentinels.nl`). Sentinels is financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

References

1. Adida, B., Neff, C.A.: Ballot Casting Assurance. In: Proc. 2006 USENIX/ACCURATE Electronic Voting Technology Workshop. USENIX Association, Berkeley (2006)
2. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections (extended abstract). In: Proc. 26th ACM Symp. on Theory of Computing, pp. 544–553. ACM, New York (1994)

3. Benaloh, J.: Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. In: Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop, EVT 2007, p. 14. USENIX Association, Berkeley (2007)
4. Burmester, M., Magkos, E.: Towards Secure and Practical e-Elections in the New Era. In: Advances in Information Security, vol. 7. Kluwer Academic Publishers, Dordrecht (2003)
5. Chaum, D.: Secret-Ballot Receipts: True Voter-Verifiable Elections. IEEE Security and Privacy 2(1), 38–47 (2004)
6. Chaum, D., Ryan, P.Y.A., Schneider, S.A.: A Practical Voter-Verifiable Election Scheme. In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 118–139. Springer, Heidelberg (2005)
7. Chevallier-Mames, B., Fouque, P.A., Pointcheval, D., Stern, J., Traoré, J.: On Some Incompatible Properties of Voting Schemes. In: Workshop on Trustworthy Elections, WOTE 2006 (2006)
8. Delaune, S., Kremer, S., Ryan, M.: Coercion-Resistance and Receipt-Freeness in Electronic Voting. In: CSFW, pp. 28–42. IEEE Computer Society, Los Alamitos (2006)
9. Dolev, D., Yao, A.C.C.: On the Security of Public Key Protocols. IEEE Transactions on Information Theory 29(2), 198–207 (1983)
10. Fujioka, A., Okamoto, T., Ohta, K.: A Practical Secret Voting Scheme for Large Scale Elections. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 244–251. Springer, Heidelberg (1993)
11. Garcia, F., Hasuo, I., Pieters, W., Rossum, P.v.: Provable anonymity. In: Proc. 3rd Workshop on Formal Methods in Security Engineering, pp. 63–72. ACM, New York (2005)
12. Gomulkiewicz, M., Klonowski, M., Kutylowski, M.: Rapid mixing and security of Chaum's visual electronic voting. In: Snekkenes, E., Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 132–145. Springer, Heidelberg (2003)
13. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 539–556. Springer, Heidelberg (2000)
14. Jakobsson, M., Juels, A., Rivest, R.L.: Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking. In: Proceedings of the 11th USENIX Security Symposium, pp. 339–353. USENIX Association, Berkeley (2002)
15. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Proc. ACM Workshop on Privacy in the Electronic Society, pp. 61–70. ACM, New York (2005)
16. Karlof, C., Sastry, N., Wagner, D.: Cryptographic Voting Protocols: A Systems Perspective. In: Proc. 14th USENIX Security Symposium, pp. 33–50 (2005)
17. Küsters, R., Truderung, T.: An Epistemic Approach to Coercion-Resistance for Electronic Voting Protocols. In: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy (S&P), pp. 251–266. IEEE Computer Society, Los Alamitos (2009)
18. Rivest, R.L., Smith, W.D.: Three voting protocols: ThreeBallot, VAV, and Twin. In: Proc. Electronic Voting Technology Workshop. USENIX (2007)
19. Ryan, P.Y.A.: A variant of the chaum voter-verifiable scheme. In: Proc. 2005 Workshop on Issues in the Theory of Security, pp. 81–88 (2005)
20. Smyth, B., Ryan, M.D., Kremer, S., Kourjieh, M.: Election verifiability in electronic voting protocols. In: Proceedings of the 4th Benelux Workshop on Information and System Security (WISSEC 2009). Louvain-la-Neuve, Belgium (2009)