

Election verifiability in electronic voting protocols^{*†}

Steve Kremer¹ / Mark Ryan² / Ben Smyth^{2,3}

¹LSV, ENS Cachan & CNRS & INRIA, France

²School of Computer Science, University of Birmingham, UK

³École Normale Supérieure & CNRS & INRIA, France

1 Electronic voting

Electronic voting systems are being introduced, or trialled, in several countries to provide more efficient voting procedures. However, the security of electronic elections has been seriously questioned. A major difference with traditional paper based elections is the lack of transparency. In paper elections it is often possible to observe the whole process from ballot casting to tallying, and to rely on robustness characteristics of the physical world (such as the impossibility of altering the markings on a paper ballot sealed inside a locked ballot box). By comparison, it is not possible to observe the electronic operations performed on data. Computer systems may alter voting records in a way that cannot be detected by either voters or election observers. A voting terminal's software might be infected by malware which could change the entered vote, or even execute a completely different protocol than the one expected.

2 Election verifiability

The concept of *election verifiability* or *end-to-end verifiability* that has emerged in the academic literature, aims to address this problem. It should allow voters and election observers to verify, independently of the hardware and software running the election, that votes have been recorded, tallied and declared correctly. One generally distinguishes two aspects of verifiability.

- *Individual verifiability*: a voter can check that her own ballot is included in the election's bulletin board.
- *Universal verifiability*: anyone can check that the election outcome corresponds to the ballots published on the bulletin board.

We identify another aspect that is sometimes included in universal verifiability.

- *Eligibility verifiability*: anyone can check that each vote in the election outcome was cast by a registered voter and there is at most one vote per voter.

^{*}This work has been partly supported by the EPSRC projects *UbiVal* (EP/D076625/2), *Trustworthy Voting Systems* (EP/G02684X/1) and *Verifying Interoperability Requirements in Pervasive Systems* (EP/F033540/1); the ANR *SeSur AVOTÉ* project; and the *Direction Générale pour l'Armement* (DGA).

[†]A long version can be found online <http://www.bensmyth.com/publications/10tech/>.

We explicitly distinguish eligibility verifiability as a distinct property.

3 Our contribution

In the full version, we present a definition of election verifiability which captures the three desirable aspects. We model voting protocols in the applied pi calculus and formalise verifiability as a triple of boolean tests Φ^{IV} , Φ^{UV} , Φ^{EV} which are required to satisfy several conditions on all possible executions of the protocol. Φ^{IV} is intended to be checked by the individual voter who instantiates the test with her private information (*e.g.*, her vote and data derived during the execution of the protocol) and the public information available on the bulletin board. Φ^{UV} and Φ^{EV} can be checked by any external observer and only rely on public information, *i.e.*, the contents of the bulletin board.

The consideration of eligibility verifiability is particularly interesting as it provides an assurance that the election outcome corresponds to votes legitimately cast and hence provides a mechanism to detect ballot stuffing.

A further interesting aspect of our work is the clear identification of which parts of the voting system need to be trusted to achieve verifiability. As it is not reasonable to assume voting systems behave correctly we only model the parts of the protocol that we need to trust for the purpose of verifiability; all the remaining parts of the system will be controlled by the adversarial environment. Ideally, such a process would only model the interaction between a *voter* and the voting terminal; *that is, the messages input by the voter*. In particular, the voter should not need to trust the election hardware or software. However, achieving absolute verifiability in this context is difficult and protocols often need to trust some parts of the voting software or some administrators. Such trust assumptions are motivated by the fact that parts of a protocol can be audited, or can be executed in a distributed manner amongst several different election officials. For instance, in Helios 2.0, the ballot construction can be audited using a cast-or-audit mechanism. Whether trust assumptions are reasonable depends on the context of the given election, but our work makes them explicit.

Tests Φ^{IV} , Φ^{UV} and Φ^{EV} are assumed to be verified in a trusted environment (if a test is checked by malicious software that always evaluates the test to hold, it is useless). However, the verification of these tests, unlike the election, can be repeated on different machines, using different software, provided by different stakeholders of the election. Another possibility to avoid this issue would be to have tests which are human-verifiable.

We apply our definition on three case studies: the protocol by Fujioka *et al.*; the Helios 2.0 protocol which was used with effect in recent university elections in Belgium; and the protocol by Juels *et al.*. This demonstrates that our definition is suitable for a large class of protocols; including schemes based on mixnets, homomorphic encryption and blind signatures. We also notice that Helios 2.0 does not guarantee eligibility verifiability and is vulnerable to ballot stuffing by dishonest administrators.