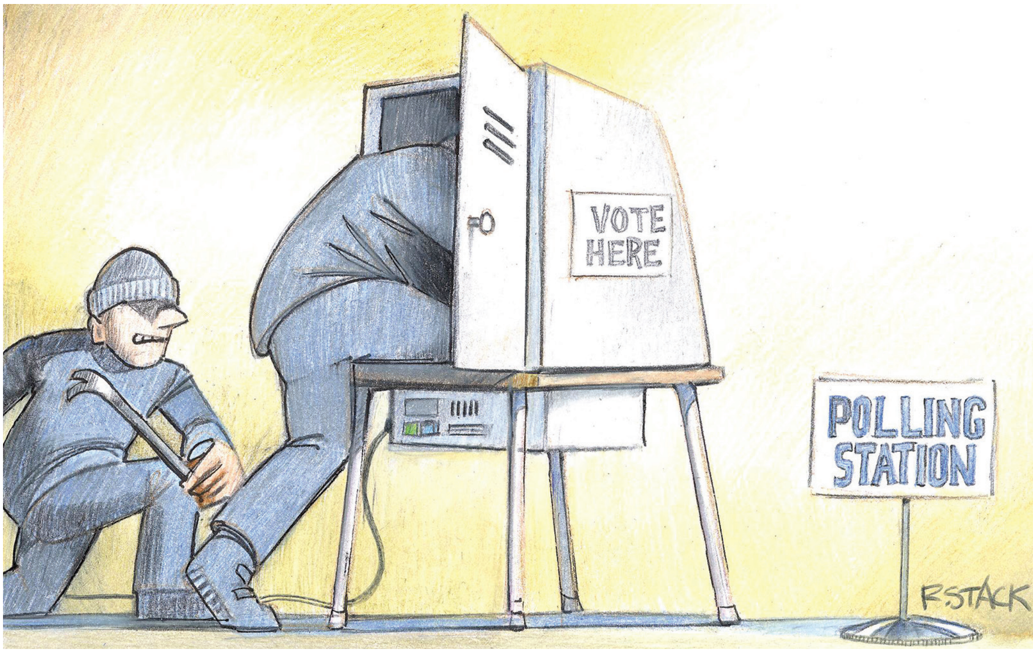
Editors: Mohamed Kaâniche, mohamed.kaaniche@laas.fr | Aad Van Moorsel, aad.vanmoorsel@ncl.ac.uk

End-to-End Verifiability in Voting Systems, from Theory to Practice

Peter Y.A. Ryan | University of Luxembourg
Steve Schneider | University of Surrey
Vanessa Teague | University of Melbourne



Since the dawn of democracy, societies have been experimenting with technological means to tackle corruption and avoid the need to trust officials. Excavations of Ancient Greece have revealed mechanisms that were clearly designed to ensure allotment: the randomness of the selection of people for office. In response to a rash of corrupted elections in the US in the late 19th century, countless devices were created that promised to provide incorruptible vote recording and counting. Thomas Edison even patented an electronic vote-recording device, and monstrous Metropolis-style lever machines persisted in some US states until very recently.

Throughout the history of democracy, there's been a battle between those trying to ensure the integrity of elections and those seeking to undermine them. The human ingenuity that has been poured into this war is truly impressive; see Andrew Gumbel's *Steal this Vote: Dirty Elections and the Rotten History of Democracy in America* for a highly entertaining—and somewhat terrifying—account.[1] The combat continues unabated, but now with new technology available to both sides. Cryptographers and those in information security have attempted to address the problem since the turn of the 21st century. Modern cryptography opens up a realm of new possibilities, but like all technology, cryptography and digital innovations are double-edged swords, opening up new threats.

Some argue that voting is a human activity that should remain in the traditional, even ceremonial realm: casting paper votes into ballot boxes and counting the resulting pile of ballots by hand. Others worry that any move to digital voting technology will enable systematic corruption. This position does hold some merit: it's true that any hasty, ill-thought-out innovation could result in disaster. Indeed, this has been demonstrated many times, such as with the California Top-to-Bottom Review of voting (https://www.sos.ca.gov/voting-systems/oversight/top-to-bottom-review.htm), where the team analyzing commercial voting systems in California declared that "virtually every important software security mechanism is vulnerable to circumvention." It's clear, then, that innovations must be developed with extreme care. But the argument that moving away from the traditional voting system will be disastrous is misguided.

End-to-End Verifiability

The promise of *end-to-end verifiability* (E2EV) gives us hope that digital technologies can provide benefits in terms of security, and not just in terms of convenience and usability. E2EV uses some of the novel properties of modern cryptography to offer something completely new and quite remarkable: the means for voters to confirm that their vote is accurately included in the tally while preventing any third party from determining how they voted, even with their cooperation. In essence, voters can privately create an encryption of their vote. All encrypted votes are posted to a public website, where voters can confirm that their vote is correctly recorded. The batch of encrypted votes is anonymized and decrypted in a universally verifiable fashion and can then be tabulated.

The fundamental challenge in public voting is how to reconcile the conflict between demonstrable integrity and ballot privacy. The E2EV solution is the classic computer science way of introducing an indirection: the encryption and decryption of votes. A short, gentle introduction to E2EV can be found at http://arxiv.org/abs/1504.03778.

Although E2EV sounds simple, it's really quite complex. The implementation of E2EV has to be sufficiently simple and usable for voters, election officials, and candidates to feel comfortable. A particularly delicate step is encrypting the ballot in such a way so that voters are confident that their vote has been correctly encoded without involving a third party. The most common approach to achieving ballot assurance is the *Benaloh challenge*: voters tell the device how they wish to vote, and this commits to an encryption. Voters can now challenge this—requiring that the encryption be opened—or cast their ballot. Voters are free to repeat this as many times as they wish until they feel confident that the device is behaving correctly. Of course, it's essential that the device not know in advance how voters will choose.

In recent years, we've seen such systems start to move from academic articles into the real world. In 2009, the Scantegrity II system, which uses the E2EV approach, was successfully used in municipal elections in Takoma Park, Maryland.[2]

vVote

Last November in Victoria, Australia, a system called vVote, based on the Prêt à Voter approach,[3] was successfully used by a section of the electorate. The system allowed for E2EV electronic voting in supervised polling places—the first time this was done in a politically binding statewide election—for voters with disabilities, such as vision impairment, and for Australian citizens voting remotely from London, England. Votes were cast privately in a voting booth and then transferred electronically to a central count. Because the electronic system ran in parallel with the traditional paper voting system, the final step in which the electronic votes were merged with the physical ones could be observed only by poll watchers who were present. Apart from that, all other steps could be verified by voters.

The key idea behind the Prêt à Voter approach, which vVote inherits, is to encode votes using a randomized candidate list, which ensures the secrecy of each vote and removes any bias. Once a ballot is marked by a voter, the candidate list is detached and destroyed. An encryption of the candidate order is preserved and used to extract the vote during tabulation.

This gives voters four steps of verification:

1. Before casting a vote, voters can confirm that the printed ballot with the randomized candidate list is properly constructed. When given a ballot, voters can choose to challenge it by demanding cryptographic proof of its correctness, which they can take home and verify. Voters can challenge as many ballots as they like before accepting one.
2. When the voting computer prints out their marked ballot, voters can check that the marks align properly with the randomized candidate list.
3. Once the candidate list is destroyed, voters leave the polling place with a receipt that includes their printed ballot and the encrypted candidate order. Voters can see that their ballot appears on a public list of accepted votes without revealing how they voted.
4. Anyone can verify that all the votes on the public list are properly shuffled and decrypted.

All of these steps—aside from the second—can be performed by or with the help of proxies of the voters' choice. Every aspect of the system is available for scrutiny: every check that voters perform with a computer can be independently recompiled, reimplemented, or performed by a completely independent party.

The source code for vVote is available at https://bitbucket.org/vvote. A nontechnical guide is available at http://electionwatch.edu.au/victoria-2014/click-here-democracy-e-vote-explained, and the complete system description and security analysis can be found in Chris Culnane and his colleagues' "vVote: A Verifiable Voting System."[4]

The vVote system was designed to handle up to hundreds of thousands of votes, though for this particular election, access to the system in the State of Victoria was restricted to 24 early voting centers and to voters with disabilities. In addition, voters in London, England, were able to use the system to cast their vote in a supervised polling place at the Australian High Commission. For these groups, 1,121 votes were cast using the system, more than the number of remote electronic votes cast in

2010, and with a quarter of the number of polling places available. A survey of the voters in London found that more than 75 percent agreed or strongly agreed with the statement that the system was easy to use.

Issues and Challenges

Although voter feedback seems to be fairly positive, there are some issues regarding existing E2EV techniques. The very concept of being able to verify a vote rather than blindly trusting a system is novel for voters and requires an effort by the authorities to educate and motivate the electorate. Usability remains a challenge for E2EV systems, as discussed in Fatih Karayumak and his colleague's "User Study of the Improved Helios Voting System Interfaces."[5] Verification needs to be simple enough so voters can understand its purpose and feel motivated to perform the checks in significant numbers. It's not sufficient for voters to simply follow the system's instructions—without performing any checks—as attackers could manipulate the code issuing the instructions.

The fundamental challenge in public voting is how to reconcile the conflict between demonstrable integrity and ballot privacy.

Another challenge is that a system can't simply be verifiable—it's essential that the system is actually verified randomly many times to ensure confidence in the result. In the case of the November 2014 election in Victoria, observation of the remote voters in London suggested that the majority did perform some check of the printed receipt against the candidate list, and around 13 percent of those using vVote checked receipts on the public website.[6]

There are a number of alternative commercial systems that claim to be verifiable but don't actually allow voters to perform their own checks. Of course, this can result in a more appealing "vote and go" user interface. With the iVote system, used in the 2015 state elections in Victoria's neighboring state of New South Wales, only a small number of chosen auditors could verify the system's output. Voters can check their own votes only by querying a database, instead of seeing the evidence themselves and checking it with their own machine as they can with E2EV voting.

One of the authors of this article co-discovered a serious security vulnerability in the 2015 New South Wales election. It was easily patched, but only after 66,000 votes had been cast.[7] Given that iVote's "verification" mechanism is unavailable for external review, there's a risk that it contains errors or security holes. This is important because trust in a small number of computers represents a potential avenue for undetectable, large-scale electoral manipulation if attackers can compromise that small set.

System Verification versus E2EV

It's important to note that the philosophy behind E2EV systems is quite different from what's usually meant by "system verification." In the latter, the idea is to perform a detailed analysis of a system's design and implementation against a set of required properties. Thus, as long as the verified code is running at execution time and the verification is complete and correct, the system should uphold the required properties. In practice, it's extremely difficult to achieve all this, especially due to the rather open, distributed nature of voting systems.

By contrast, E2EV seeks to ensure that the system execution is fully auditable. This idea is nicely captured in Josh Benaloh's maxim: "Verify the election, not the system." A related concept is Ronald L. Rivest and John P. Wack's notion of "software independence," which says that any error in the code that could result in a change in the outcome must be detectable at execution time (http://people.csail.mit.edu/rivest/RivestWack-OnTheNotionOfSoftwareIndependenceInVotingSystems.pdf). Of course, this doesn't mean that verification of the design and code should be neglected—it just means that the integrity of the outcome should not be dependent on assumptions about the correctness of the running code.

Another project is the End-to-End Verifiable Internet Voting Project (www.overseasvotefoundation.org/E2E-Verifiable-Internet-Voting-Project/News), which is examining E2EV in an attempt to define the real requirements of verifiability, so vendor systems that are not truly E2EV—but claim to be—can be differentiated from systems that are.

End-to-end verifiability represents a paradigm shift in electronic voting, providing a way to verify the integrity of elections by allowing voters to audit the information published by the system, rather than trusting that the system has behaved correctly. Recent deployments of E2EV systems in real elections demonstrate its practical applicability, and we hope to one day see E2EV as the normal expectation for electronic voting systems. ■

References

1. A. Gumbel, *Steal this Vote: Dirty Elections and the Rotten History of Democracy in America*, Nation Books, 2005.

2. R. Carback et al., “Scantegrity II Municipal Election at Takoma Park: The First E2E Binding Governmental Election with Ballot Privacy,” *Proc. 19th USENIX Conf. Security* (USENIX Security 10), 2010, p. 19.
3. P.Y.A. Ryan et al., “Prêt à Voter: A Voter-Verifiable Voting System,” *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, 2009, pp. 662–673.
4. C. Culnane et al., “vVote: A Verifiable Voting System,” arXiv:1404.6822, 2014; http://arxiv.org/abs/1404.6822.
5. F. Karayumak et al., “User Study of the Improved Helios Voting System Interfaces,” *1st Workshop Socio-Technical Aspects in Security and Trust* (STAST 11), 2011, pp. 37–44.
6. C. Burton, C. Culnane, and S. Schneider, “Secure and Verifiable Electronic Voting in Practice: The Use of vVote in the Victorian State Election,” arXiv:1504.07098, 2015; http://arxiv.org/abs/1504.07098.
7. V. Teague and A. Halderman, “Security Flaw in New South Wales Puts Thousands of Online Votes at Risk,” *Freedom to Tinker*, 22 Mar. 2015, https://freedom-to-tinker.com/blog/teaguehalderman/ivote-vulnerability.

Peter Y.A. Ryan is a professor of applied security at the University of Luxembourg. Contact him at peter.ryan@uni.lu.

Steve Schneider is a professor of computing and Director of the Surrey Centre for Cyber Security at the University of Surrey. Contact him at s.schneider@surrey.ac.uk.

Vanessa Teague is a research fellow in the Department of Computing and Information Systems at the University of Melbourne. Contact her at vjteague@unimelb.edu.au.