1-1-2004

But We Have to Protect Our Source: How Electronic Voting Companies' Proprietary Code Ruins Elections

Andrew Massey

Follow this and additional works at: https://repository.uchastings.edu/hastings_comm_ent_law_journal

Part of the Communications Law Commons, Entertainment, Arts, and Sports Law Commons, and the Intellectual Property Law Commons

Recommended Citation
Available at: https://repository.uchastings.edu/hastings_comm_ent_law_journal/vol27/iss1/6

This Note is brought to you for free and open access by the Law Journals at UC Hastings Scholarship Repository. It has been accepted for inclusion in Hastings Communications and Entertainment Law Journal by an authorized editor of UC Hastings Scholarship Repository. For more information, please contact wangangela@uchastings.edu.
"But we have to protect our source!": How Electronic Voting Companies' Proprietary Code Ruins Elections

by
ANDREW MASSEY*

I. Introduction .................................................................................. 234

II. Voting Systems and Open Source Code .................................... 236
   A. Voting Systems ............................................................................. 236
   B. Open Source Code ....................................................................... 238

III. Proprietary Code Creates Obstacles to Electronic Voting
   A. Proprietary Code Undermines Transparent and Accountable Government .......................................................... 241
   B. States Are Unable to Regulate Proprietary Code .................... 244
   C. Proprietary Code and Recounts ................................................ 245

IV. Open Source is the Best Solution ............................................ 247
   A. Open Source Makes Electronic Voting More Accountable and Secure .............................................................. 248
   B. Open Source Code Can Be Used to Conduct "Digital Recounts" ........................................................................ 250
   C. Open Source Makes Elections More Legitimate ...................... 253
   D. Open Source Code Should Be Legalized ................................. 254

V. Conclusion .................................................................................... 255

* J.D. Candidate 2005, U.C. Hastings College of the Law, B.A., History, 2001, U.C. Berkeley; Co-Editor-in-Chief, Hastings Communications and Entertainment Law Journal, 2004-2005. He wishes to thank his advisor, Professor David Jung for his essential critiques and suggestions; his brother, Steven, for the inspiration that lead to this note; his parents; and Cassandra, for putting up with him.
I. Introduction

"It's not the voting that's democracy, it's the counting..."
Tom Stoppard, *JUMPERS*

"I am committed to helping Ohio deliver its electoral votes to the president next year."
Walden W. O'Dell, chief executive of Diebold, Inc.

Walden W. O'Dell wrote what appears as a seemingly innocuous platitude in a fund-raising letter in support of President George W. Bush's 2004 re-election campaign, but this sentence has created a firestorm of controversy that continues to make headlines across the country. The uproar results from O'Dell's job as chief executive of Diebold, Inc., a major manufacturer of touch-screen electronic voting machines, otherwise known as Direct Recording Electronic devices ("DREs"). Unlike traditional paper-ballot systems, O'Dell's machines—and other manufacturer's machines like them—record votes in secret, meaning that the voting public has no idea whether its vote counted, and if it did count, whether it went to the person for whom the voters voted. In its current implementation, this new technology that lawmakers intended to correct the flaws in the 2000 Presidential election and restore legitimacy and trust to the electoral system has produced just the opposite: a system open to fraud, mistake, or error which could easily escalate into problems far worse than Florida's hanging chads.

The source of this problem is the proprietary source code that drives these paperless electronic voting machines. Unlike paper-based voting machines, DREs operate entirely by computer, meaning that at no stage of the election process can the public see the physical counting of the votes. Absent access to the source code that runs the DRE, the public has no way of knowing how—or if—the machine records and tabulates votes. As a result, a growing chorus of critics

4. Warner, supra note 2. I will use these terms interchangeably.
claim that without the ability to scrutinize the process, the public has no way to protect against malicious manufacturers, elections officials, or voters from "hacking" the machines to "deliver" votes to their chosen candidate.6

In practice, proprietary code-based DREs have proven to be error-ridden and prone to security weaknesses because the closed nature of the code has forced state agencies to protect manufacturers' intellectual property7 at the expense of a reliable voting system. The proprietary nature of the code requires a closed state review process that has not eliminated serious errors and security flaws8 because it limits the number of people testing the software.9 That closed process also contradicts public policy and American tradition favoring openness through transparent and accountable government.° As a result, the electorate is forced to rely upon arguably substandard machines to conduct one of the most important functions of our democratic system.

Activists and legislators have seized upon adding a paper receipt redundancy feature, or "voter-verified paper trail,"10 as a solution to


9. See, e.g., CAL. ELEC. CODE § 19206 (West 2004).

10. See, e.g., Cal. Secretary of State, 1 Procedures for Approving, Certifying, Reviewing, Modifying, and Decertifying Voting Systems, Vote Tabulating Systems, Election Observer Panel Plans, and Auxiliary Equipment, Materials and Procedures § 101 at http://www.ss.ca.gov/elections/vsp_procedures.pdf (last visited Jan. 20, 2004) ("In furtherance of open elections at each phase of the automated process of tabulating ballots, the Secretary of State requires that each election jurisdiction . . . provide access to that portion of the elections process.").

11. See Mercuri, supra note 6. See also Dill et. al., supra note 8, at §1. See also 149 CONG. REC. E1081-2 (daily ed. May 22, 2003) (statement of Rep. Holt) (introducing the
the potential for rigged or flawed DREs. I will argue against the paper trail solution, however, as it is (1) merely a “band-aid” for the real problem of proprietary code, (2) is subject to the same potential risks as existing paper-based voting systems, and (3) does nothing to further the general goal of restoring legitimacy to voting. By contrast, I will propose that states and the FEC mandate the use of open source code to ensure transparency and accountability mandated by law. In addition, states must eliminate escrow requirements to allow for public testing of the source code, which will further serve the policy goal of restoring trust and openness to the election system.

This note was written and submitted for publication in advance of the 2004 Presidential election in which DREs will face their most important test. The timing of the submission was deliberately made in the hope of avoiding being dismissed as a partisan post-election lament, or serving as the basis for a challenge to a specific vote count. Secure, reliable and fair elections are essential to American democracy, and should be in the interest of the general public independent of ideology or political affiliations. Transparent and accountable voting systems are crucial in achieving that end because, as the quote above from Tom Stoppard so wryly notes, vote counting is ultimately more important than vote casting. This note is therefore offered in the hope of contributing to the nonpartisan development of such a voting system and restoring legitimacy to the American electoral process.

II. Voting Systems and Open Source Code

A. Voting Systems

Currently five different kinds of ballot systems are used in the United States: hand-counted paper ballots, mechanical lever machines, computer punch-cards, optical scanners, and DREs. Of

these five machines, all but DREs rely on paper. The remaining machines require the voter to mark the ballot with a pen, pencil, or metal punch.

Although as of 2000 DREs only covered 10.7% of the population, a significant increase in use is all but assured and will probably occur in a very short period of time. Use of DREs has grown markedly in the past twenty years. Furthermore, growth in the use of DREs and optical scanners has come at the expense of paper ballots and aging lever machines. In addition, the near-Constitutional crisis brought about by the failure of punch-card machines in the 2000 Presidential election has produced both federal and state legislation financing their immediate replacement. The Help America Vote Act in particular has a 2006 deadline for the purchase of this new equipment.

States have turned to DREs because their unique touch-screen, computer-based voting procedure offers both voters and government a wide variety of benefits not found in any other voting technology. For voters, DREs offer the possibility of providing a ballot in an unlimited number of languages, multiple size type for the visually-impaired, photos of the candidates, text-to-speech technology for the blind, online “links” to a voter information packet, on-screen prompts to avoid undervotes, and much more. Also, the flexibility of the user

15. Id. at 2-3.
16. Id. at 5.
17. Id.
20. Id. at § 301(d).
21. Voting Technologies, supra note 13, at 5; The Caltech/MIT Voting Technology Project, Voting: What Is, What Could Be, 22, 25, 27, 59 (July 2001), at http://www.vote.caltech.edu/Reports/2001report.html [hereafter The Caltech/MIT Voting Technology Project]. There has been some concern about whether certain DREs make access by the disabled more difficult. In particular, proposals for a paper trail have been problematic for the visually-impaired. Recently, however, the U.S. Department of Justice issued an opinion to California Secretary of State Kevin Shelley suggesting that that
interface on DREs allows for many different kinds of ballots and for altering ballots to meet changing needs over time.\textsuperscript{22}

For government, DREs mean the elimination of paper which can become spoiled or altered.\textsuperscript{23} Paper ballots are also expensive and difficult to print, store, and transport.\textsuperscript{24} Furthermore, tabulation of the election results without some level of computerization can be extremely time-consuming and labor-intensive, often pushing costs above those necessary for the purchase and maintenance of DREs.\textsuperscript{25} Finally, and perhaps most importantly, electronic voting methods, including DREs, have been viewed by government as a means to encouraging voter turnout, particularly among young people who tend not to vote.\textsuperscript{26} All of these factors mean that DREs are likely to be purchased in great numbers in the next few years, making a resolution about questions of their reliability extremely urgent.

\textbf{B. Open Source Code}

Source code is the set of instructions that govern the abilities of computer software.\textsuperscript{27} A computer programmer writes the source code in a programming language that bears some resemblance to English, such as the popular C, C++, or Java.\textsuperscript{28} Once the source code is completed, the programmer "compiles" the code, a process by which the English-like programming language is translated into binary code (consisting of 1s and 0s) that the computer hardware can

\begin{quote}


26. \textit{Alternative Ballot Techniques: Hearing Before the House. Subcomm. on Elections of the Comm. on House Admin.}, 103rd Cong. (Sept. 22, 1994) (in which various state officials and elections experts testified on test projects including a telephonic voting system in New Mexico that produced higher turnout among young people). \textit{See also Residual Votes}, supra note 14, at 40. \textit{But see} Mercuri, supra note 6, at 48 (suggesting that "increased voter participation . . . for deploying Internet voting systems [is] . . . relatively insignificant," though citing only examples from the United Kingdom).

27. \textit{Medforms, Inc. v. Healthcare Mgmt. Solutions, Inc.}, 290 F.3d 98, 104 (2d Cir. 2002) ("Source code is a mathematical set of instructions that a computer converts into an executable program that can then be distributed and run by other computers.")

\end{quote}
understand. After the source code is compiled, it becomes a software program, such as a word processor, web browser, or flight simulator.

Once compiled, however, the source code cannot reliably be decompiled, making access to the exact original source code underlying the software program nearly impossible. A person with a compiled software program on a computer can use its features according to the abilities given to it by the source code, but that person cannot figure out how the software program works. The relationship is analogous to a person being able to drive an automobile without being able to open up the hood, take apart the engine, and determine how it makes the automobile operate. While a software program can be decompiled, the process rarely produces an exact copy of the original source code. Decompiling could give a software programmer enough information about the original source code to get a basic idea of how the program functions, but not enough information to duplicate that function. The decompiled software program might contain enough information, however, to inform a hacker how to exploit security flaws in the program's design.

Software developers utilize the inability to resurrect an exact copy of the source code through decompiling to protect the intellectual property within the source code, such as algorithms and creative software design. Thus, unsurprisingly, commercial software developers usually distribute only the compiled software program unaccompanied by the source code as one of the primary protections for their intellectual property and trade secrets. Software distributed without the source code is referred to as proprietary code. If a software developer distributed the source code along with the compiled software program, the recipient could easily recompile the source code, put his or her name on it, and sell it as his or her own.

29. Id.
30. Id. at 319.
34. Election Reform, supra note 31.
36. Id. at 320. Distributing compiled software programs without the source code is used along with the more traditional legal intellectual property protections of copyright and patent. Id.
The computer end-user would be completely unaware that the source code underlying a software program was not developed by the person whose name appears on the software.

Despite the loss of intellectual property protection, many software developers distribute their source code with the express intent that others make use of that source code; software distributed using this method is known as open source code. Open source code-based software may be distributed commercially or for free, and may be copyrighted or more often explicitly not copyrighted. More importantly, open source code allows access to the precise instructions for how the compiled software program will function, meaning that a programmer can determine the methods the software program uses to achieve its functionality.

Understanding how the software program operates has two important implications. First, programmers can scrutinize the source code to look for errors, poorly written code, or security weaknesses. In addition, programmers can improve the efficiency or design of the source code or add additional features to those already existing in the software. Second, if the source code is known to everyone, nefarious programmers cannot use the closed nature of proprietary code to include malicious code that would wreak havoc on a computer, and in turn the important human activities that rely increasingly on computers.

38. Id. at 320-21. "Open source software refers to a computer program whose source code is made available to the general public to be improved or modified as the user wishes." Jeffrey W. Seifert, Computer Software and Open Source Issues: A Primer CONGRESSIONAL RESEARCH SERVICE REPORT FOR CONGRESS, RL31627 at 1, (Nov. 5, 2001) available at http://www.ipmall.piercelaw.edu/hosted_resources/crs/RL31627.pdf.

39. Evans & Reddy, supra note 28, at 321-24. Open source software is typically distributed under two kinds of licenses: BSD or GPL. Under the BSD license, a user may modify, recompile, and distribute the source code, so long as the original copyright is acknowledged. By contrast, under the GPL license, a user may likewise modify, recompile and distribute the source code, but does not have to acknowledge a copyright, and more importantly, cannot claim any copyright for the altered source code. The GPL license is the more prevalent of the two schemes, and is used in the popular Linux operating system to facilitate input from programmers interested in improving the system. See id.

40. Id. at 319.


42. Seifert, supra note 38, at 1.

By contrast, proprietary code-based software programs operate on the "security through obscurity" model which presumes that a system is more secure the less anyone knows about it and its potential weaknesses. Advocates argue that security through obscurity "makes potential flaws more difficult to discover and therefore to exploit." Most security experts agree, however, that secret source code as security is inherently fragile, because once exposed, the damage is irreversible because the secret is known. In addition, to remain secret, the proprietary code must be available to only a select group of people, meaning that only a few programmers will scrutinize the code for errors, as opposed to the potentially limitless group who test open source code.

When used in conjunction with electronic voting, open source software can alleviate serious technical and security problems inherent to the use of proprietary code that can compromise the election process and cripple the system's legitimacy. In addition, proprietary code undermines public policy and the American tradition of transparent and accountable government. The next section shall address these two concerns.

III. Proprietary Code Creates Obstacles to Electronic Voting

A. Proprietary Code Undermines Transparent and Accountable Government

The public interest is protected "through having transparent and accountable government," and "openness in government has always been thought crucial to ensuring that the people remain in control of their government." Furthermore, the "right of the public to assure itself that the election process is free, fair, and transparent, is likewise of great public concern." Therefore elections, more than other government-sponsored activities, require a heightened form of transparency and accountability, to ensure the public their fundamental right to vote. Proprietary code is inappropriate for

44. Election Reform, supra note 31, at 26.
45. Id.
46. Id.
47. Id.
49. In re Sealed Case (Espy), 121 F.3d 729, 749 (D.C. Cir. 1997).
electronic voting because the closed nature of the code provides for neither transparency nor accountability.

As evidenced above, problems associated with proprietary source code are not unique to electronic voting; however, the election environment creates important additional requirements. Unlike using a word processor or playing video games, voting on a DRE (or any voting machine) constitutes the exercise of a fundamental right.52 Furthermore, the public voting process acts to preserve all other rights.53 Thus the legitimacy of the voting system has a direct correlation to a voter's perception that government uphold all other rights.54 A voting system operating in secret through proprietary code that masks how a DRE records and counts votes is fundamentally inappropriate to a public function of such vital importance. Indeed, the public already has great experience with unreliable voting systems and has seen how they can cause significant turmoil.55 Therefore, elections heighten the need for transparency and accountability beyond most other situations employing computer software.

Accountability for voting systems through traditional auditing practices of following a paper trail cannot easily be achieved because of the twin requirements of voter anonymity and the secrecy of the actual vote.56 Indeed, voter anonymity and secrecy are in fundamental conflict with the audit process.57 To audit a voting system, there must be a method to retrace the path from vote total to the original intent of each voter.58 Making that determination within the constraints of ballot secrecy and voter anonymity is nearly impossible, and attempts to do so often fail to produce satisfactory results.59 The most famous example of this problem is the endless recounts of pregnant and dimpled chads in Florida during the 2000 Presidential election.60 Using proprietary code adds yet another constraint to election

52. See, e.g., Yick Wo v. Hopkins, 118 U.S. 356, 370 (1886).
53. Id.
56. Mercuri, supra note 6, at 46; Aviel D. Rubin, Tadayoshi Kohno, Adam Stubblefield, and Dan S. Wallach, Analysis of an Electronic Voting System, 2 (July 23, 2003) at http://avirubin.com/vote.pdf (providing the first comprehensive independent analysis of DRE code by computer scientists other than approved government officials or contractors, or the developer).
57. Mercuri, supra note 6, at 46.
58. Id.
59. Id. See generally Bush, 531 U.S. 98.
60. The Caltech/MIT Voting Technology Project, supra note 21, at 17.
auditing by making it impossible to understand how the machine recorded the vote and subsequently tabulated it.\textsuperscript{61} Without access to the source code that runs the DRE, auditing becomes a pointless endeavor because all an auditor has to work with is potentially flawed election data produced by a black box in which it is impossible to see how it created that data.\textsuperscript{62}

Researchers at the Caltech/MIT Voting Technology Project, as well as the Congressional Research Service and others, have discussed creating redundancy by splitting the vote recording and tabulating features into two machines through a variety of methods.\textsuperscript{63} By dividing the recording and tabulating functions into two separate physical devices, the proposed systems would create data redundancy to allow for audits.\textsuperscript{64} Even bifurcating the vote-recording and tabulation processes into two machines would suffer from the same problem since the proprietary code on the machine recording the vote might contain malicious or error-ridden code. If the vote was not correctly recorded, it cannot subsequently be properly counted.\textsuperscript{65} Thus the flaw remains the same.

Many have likened DREs to bank ATM machines that preserve the secrecy of the customer’s transaction through pin-codes and bank cards, but ATMs are not subject to the same requirements of ballot secrecy and voter anonymity.\textsuperscript{66} Elections require a much higher level of secrecy because unlike ATM machines that print out paper receipts, election systems do not allow voters to take a copy of their completed ballots home from the polls out of fear of coercion and vote buying.\textsuperscript{67} Furthermore, unlike bank transactions where the bank knows which customer made a withdrawal, voting systems demand strict anonymity to prevent fraud.\textsuperscript{68} As a result, once the voter has completed voting, the ballot cast by the voter can never be traced back to the original voter.\textsuperscript{69}

The inability to audit a proprietary code-based DRE creates a further set of problems, most notably, a significant conflict of interest

\textsuperscript{61} See, e.g., Shamos, supra note 6.
\textsuperscript{62} Id.
\textsuperscript{63} The Caltech/MIT Voting Technology Project, supra note 21, at 44; Election Reform, supra note 31, at 20, 27-31.
\textsuperscript{64} The Caltech/MIT Voting Technology Project, supra note 21, at 44.
\textsuperscript{65} Id.
\textsuperscript{66} Holt, supra note 11.
\textsuperscript{67} The Caltech/MIT Voting Technology Project, supra note 21, at 13.
\textsuperscript{68} Mercuri, supra note 6, at 46-47.
\textsuperscript{69} Election Reform, supra note 31, at 2.
between those who manufacture the machines and the political organizations to which they contribute campaign money.\textsuperscript{70} Proprietary source code masks not only how DREs record and count votes, but how the programmer intended the machine to record and count votes through the inclusion of malicious code.\textsuperscript{71} While this possibility could create a sinister symbiotic relationship between elected officials and DRE manufacturers, an equally powerful countervailing incentive to maintain honesty exists for no other reason than exposure risks losing business in a very lucrative industry.\textsuperscript{72} Whether or not such conspiracies exist, proprietary code creates the impression of impropriety that casts a long shadow over the general goal of restoring trust and legitimacy to the electoral system.\textsuperscript{73}

B. States Are Unable to Regulate Proprietary Code

Proprietary code leaves to state elections agencies the task of maintaining trust in electronic voting because they alone can conduct testing and regulation of DREs.\textsuperscript{74} To protect manufacturers' intellectual property, the proprietary code can only be released to a select few expert testers, limited in number by statute and funding.\textsuperscript{75} As a result, very few people actually test the code for errors, thus limiting the amount of scrutiny the code will receive.\textsuperscript{76} Furthermore, without in-house experts to constantly test the numerous updates and revisions to the source code, state officials can only conduct occasional tests.\textsuperscript{77}

The lack of in-house experts also means that state officials are left with the task of regulating machines the officials themselves may not understand. In addition, state agencies left largely on their own to regulate the source code on the machines have found themselves unable to keep up with the rapid pace of alterations to the source code, leading to the use of uncertified software in some cases.\textsuperscript{78} While

\begin{itemize}
\item \textsuperscript{70} Warner, supra note 2.
\item \textsuperscript{71} \textit{Election Reform}, supra note 31, at 2.
\item \textsuperscript{72} See generally Warner, supra note 2. See also Lipton, supra note 18.
\item \textsuperscript{73} Statement, supra note 5, at S12673-4 (statement of Sen. Dodd).
\item \textsuperscript{74} See, e.g., Weber, 347 F.3d at 1101.
\item \textsuperscript{75} See, e.g., CAL. ELEC. CODE § 19206 (West 2004) ("For the purpose of assistance in examining a voting system the Secretary of State may employ not more than three expert electronic technicians at a cost to be set by the Secretary of State.").
\item \textsuperscript{76} Election Reform, supra note 31, at 26.
\item \textsuperscript{77} See, e.g., Cal. Secretary of State, supra note 10, at Article 15 § 1501 (authorizing biennial testing of electronic or computerized voting equipment).
\end{itemize}
the use of uncertified software was eventually discovered after the election, the mere fact that the DREs were running unapproved—and potentially malicious—software suggests that state agencies, even with the use of occasional expert testers, are not yet up to the task of regulating DRE software on their own. The inability of state agencies to regulate DRE software code became painfully evident when the Ohio Secretary of State, faced with problematic source code on already purchased machines, had no recourse but to ban DRE use outright. While drastic remedies such as outright bans will of course become unavailable once older machines have been discarded.

While time and experience will inevitably improve the ability of state agencies to regulate DREs and manufacturers, states cannot overcome the inherent structural limits imposed on testing proprietary code by the need to protect intellectual property. State agencies will never be able to have enough testers, to anticipate every potential hacker's mode of attack, or to think of every potential scenario that could cause error as compared to the limitless number of potential testers in the public who could work through those problems. Ultimately, the closed review procedure for proprietary code will always dampen the public's trust.

C. Proprietary Code and Recounts

The hidden, non-tangible ballot and tabulation process created by the proprietary code running in almost all DREs in turn creates a quandary when trying to make DREs comply with existing state elections codes premised on the use of mechanical equipment and paper ballots that the public can see. Specifically, current DREs make impossible public inspection of the vote tally in a recount, as well as the recount itself. For the purpose of this analysis, the California Elections Code will be used as a model.

Recounts are premised on the idea that an error has occurred in the tabulation of the vote as compared with the votes the voters actually cast. Ideally, to perform a recount, the voters would simply be returned to the precinct and asked how they voted, but with secrecy as an essential part of voting, such a recount would be

81. The Caltech/MIT Voting Technology Project, supra note 21, at 21.
82. CAL. ELEC. CODE §§ 15360, 15627, 15629-30 (West 2004).
83. Election Reform, supra note 31, at 8.
impossible. Therefore, current recounts are typically conducted by a hand count of the physical paper ballots. In the case of voting systems that use electronic tabulation, the ballots may be re-run through the system to see if the system produces the same result.

A manual hand-count is impossible if a voting system does not have paper ballots, because there is nothing physical to count. Currently, the voter who demanded the recount may "select whether the recount shall be conducted manually or by means of the voting system used originally, or both." Without a physical ballot to inspect, the voter demanding the recount has effectively lost the right to choose a manual recount and must settle for re-running the data through the voting system.

Re-running the data through the DRE will probably never produce a different result from the original tally, however, because there is little likelihood that the same software, when fed the same data, will not produce the same result. Unlike voting machines counting paper ballots, paperless DREs have little chance of mechanical error. Instead, DREs are vulnerable to malicious or poorly-written software that might intentionally or unintentionally fail to record the actual intent of the voter because the code itself does not tabulate the votes correctly. The purpose of recounts—to thwart voting machine error—simply cannot be achieved by running flawed data back through flawed software code. The result will always be the same: wrong.

The recount provisions in the Elections Code are violated if "recounts" on DREs cannot be done by hand, and will never reveal mechanical error by re-running the data. DRE manufacturers argue that election day summaries provide all the information necessary for recount, but there is no evidence that such summaries would not be affected by the same potentially flawed software that produced the erroneous tallies.

84. Mercuri, supra note 6, at 46-47.
85. CAL. ELEC. CODE § 15627 (West 2004).
86. Id. See also Election Reform, supra note 31, at 8.
87. The Caltech/MIT Voting Technology Project, supra note 21, at 21, 45.
88. CAL. ELEC. CODE § 15627 (West 2004).
89. The Caltech/MIT Voting Technology Project, supra note 21, at 21.
90. Id. at 17.
91. Id. at 45.
92. Id.
93. Id.
More problematic are the random 1% manual election day recounts required under the Elections Code, designed to catch machine irregularities on election day to avoid requiring candidates to pay for expensive and time-consuming recounts. This law cannot be complied with as written because manual recounts are impossible on paperless DREs. Using the summary as the manual recount device frustrates the language of the statute that provides for a "public manual tally of the ballots tabulated by those devices," because the language contemplates the ballots and the device being physically separate. If the ballot and the device are physically separate, counting the ballots by hand, and using that result as a check on the results tallied by the machine, creates an independent check on the reliability of the voting machine. On the other hand, if the DRE generates the summary report, the lack of redundancy renders the recount meaningless. As a result, elections officials cannot meet their obligations under the law, and the voters are not provided the election day safeguard the legislature undoubtedly intended to provide increased reliability to the vote.

IV. Open Source is the Best Solution

The serious potential problems with DREs are not inherent to the machines, but to the proprietary code that runs them. The mere act of opening the code to the public addresses the serious concerns advanced by critics by exposing the source of the problems these machines have thus far experienced, which in turn creates the transparency necessary to the election process. Specifically, open

94. CAL. ELEC. CODE § 15360 (West 2004).
95. See discussion infra in this section.
96. The Caltech/MIT Voting Technology Project, supra note 21, at 21, 45.
97. CAL. ELEC. CODE § 15360 (West 2004).
98. The Caltech/MIT Voting Technology Project, supra note 21, at 45.
99. Id.
100. Critics have included computer scientists and activists. Computer scientists have focused on the technical problems with DREs and have tried to use their expert analyses to prompt state governments to reconsider using the machines without modification. See, e.g., Neuman, supra note 43; Rubin, supra note 56. Activists, in addition to the technical problems, have focused somewhat on what can only be termed the "conspiracy theory" aspect of the debate. See Black Box Voting, at http://www.blackboxvoting.com (last accessed Feb. 16, 2004).
101. DREs have experienced technical problems of varying degrees of severity in almost every jurisdiction in which the machines are used in the United States. These include California, Florida, Maryland, Virginia, Georgia, and Texas. See, e.g., Ian Hoffman, E-voting Runs into Bumps in East Bay, OAKLAND TRIBUNE, Nov. 5, 2003, available at http://www.oaklandtribune.com/Stories/0,1413,82-1865-1745779,00.html;
source makes electronic voting transparent, more accountable, and secure; allows for the possibility of "digital recounts;" and makes elections more legitimate.

A. Open Source Makes Electronic Voting More Accountable and Secure

Switching to open source code enhances accountability and security of elections by eliminating the secrecy fundamental to proprietary code that operates in direct opposition to the "American tradition" of "open elections." The openness of open source code corresponds with the public requirements of voting by making the process transparent and thus possible to audit, by not only government officials, but the general public as well. In addition, publicly available open source code would undergo the scrutiny of a large number of testers, increasing the software’s reliability and relieving the burden on government.

Unlike secret proprietary code, open source code running on DREs could itself be directly audited, allowing the public as well as any government official (not just those privy to material protected by escrow provisions) the ability to review the code recording and tabulating votes. If made public, DRE source code could, at least in theory, be audited and tested by an infinite number of people an infinite number of times, but would almost certainly be reviewed by more people, more often than under laws restricting testing to a small handful of outside experts on a biennial basis. Increased testing would reduce the chances that the limited number of expert testers would overlook errors or security weaknesses in the source code. In addition, the risk that programmers might exploit the closed nature of proprietary code to hide malicious instructions is diminished somewhat by the number of eyes viewing the code.

Moreover, a public testing process would shift the burden of actually conducting the tests away from state agencies. That way,
state agencies could focus on managing and policing the process which, as regulatory bodies, they are probably better designed to handle. Furthermore, if government is not the sole tester of the software, the hysteria surrounding electronic voting would probably lessen. Results from one such review process bear out that claim.

Opening source code to the public also moves away from the "security through obscurity" model that critics have charged violates "[g]ood security principles [that] dictate that the analysis of a system should presume that all components are publicly known." This fragile form of defense has already proven a problem when Diebold, Inc. accidentally left the source code to one of its DREs on an unprotected FTP site on the Internet. Although the leak allowed computer security experts to analyze the code, which led to eventual improvements, the episode highlighted the weakness of security through obscurity given that one simple error compromised the entire security model. If the leak had occurred on the eve of an election, the results might have been catastrophic given the myriad coding errors and serious security flaws that numerous studies have discovered. Had Diebold made public its source code, many of the problems computer security experts subsequently discovered would have been eliminated in a more orderly review process. A review process would have also avoided the negative national and international publicity that Diebold—and electronic voting in general—garnered as a result of the leak and discovery that the source code already used in DREs was seriously flawed.

111. Kim Zetter, Aussies Do It Right: E-Voting, WIRED NEWS at http://www.wired.com/news/ebiz/0,1272,61045,00.html. The Australian Capital Territory's open source paperless DRE voting system was designed without the level of criticism and harsh debate found in the U.S. The design of the system involved an open review process that solicited assistance from the public. See infra Part IV C.
112. Trusted Agent Report, supra note 7, at 8.
114. See Rubin, supra note 56.
B. Open Source Code Can Be Used to Conduct "Digital Recounts"

While open source code does not allow for traditional hand recounts of paper ballots, the use of open source code, coupled with effective enforcement of using properly certified software, obviates the need for traditional hand counts. Since the ability to audit the system is the purpose of recounts, the hand-counting of paper is not necessary to achieve that purpose if an alternative means produces a relatively equally reliable result. Public availability of open source code will significantly reduce the errors and malfunctioning inherent in DREs to date. In addition, the code used on a DRE in an election can be compared to the publicly-available source code, creating a publicly-available means of comparison as well as compliance with state election codes requiring public recounts.

While paperless open source DREs do not eradicate the possibility of fraud or malicious software, "[t]he unfortunate reality is that the possibility of electoral fraud can never be completely eliminated, no matter which type of ballot is used." Furthermore, requiring paper ballots for all elections might stifle important innovation, such as increasing international interest in voting on the Internet. Moreover, paper does not make elections secure because paper has been the source of a long and much lamented history of ballot-stuffing, tampering, and other security woes in American elections.

117. Weber, 347 F.3d at 1103.
118. Election Reform, supra note 31, at 22.
119. See, e.g., CAL. ELEC. CODE § 15629 (West 2004). While laws requiring the recount process to be public also serve the purpose of placing a check on the fraud-prone hand-counting process, that check could also in theory be provided by a judge or other independent official. The public access to recounts provision in state elections codes probably also serves the purpose of providing transparency, government accountability, and legitimacy to the elections process by allowing direct public scrutiny over questionable ballot tabulation.
120. Weber, 347 F.3d at 1106 (emphasis in original).
121. See Bush, 531 U.S. at 134.
123. See Shamos, supra note 6.
Fearful at the loss of traditional paper ballots, a significant number of reformers, politicians, and pundits have seized upon adding a "voter-verified paper trail" to existing DREs in an effort to restore, in their minds, the ability to audit the system. The voter-verified paper trail would add a printer to existing DREs, that upon the conclusion of voting, would print a completed ballot that the voter must "verify" before the electronic ballot is cast. Proponents believe the paper print-outs create the ability to audit the system and increase reliability because, in their minds, paper is more secure than electronics. As a result of the controversy, proponents have introduced legislation in Congress to amend HAVA to mandate the use of paper in all voting machines.

Whether motivated by a genuine interest in improving the electoral system, naïveté, or technophobia, proponents of the voter-verified paper trail ignore their system's impracticality and flawed reliance on paper. Indeed, when the challenge of electronic voting required a complete shift in thinking for the digital age, far too many could not resist looking backwards at paper, thinking that past solutions could solve the fundamentally different problems posed by computerization. Furthermore, many proponents tend to focus on the visceral experience of voting without paper, rather than considering how DREs avoid many of the problems associated with paper ballots. For instance, for all the fanfare associated with voter-verified paper trails, only a few prototypes exist, and none have been tested in an actual election where their use might create new security problems and confusion for voters unaccustomed to the complexity of ballot verification. In addition, adding a new physical component to

124. See Mercuri, supra note 6 (claiming that the author invented the idea such that it should be termed "The Mercuri Method"). See also Dill et. al., supra note 8, at §1. The voter-verified paper trail should not be confused with the voter-verified audit trail, a term which describes any method by which the voter approves of the vote cast on a DRE through some supplemental procedure. That procedure does not necessarily have to include paper. See Election Reform, supra note 31, at 8.

125. Mercuri, supra note 6, at 47, 50 (including a diagram of how a voter-verified paper trail might work).


128. See, e.g., Holt, supra note 11, at E1081.

129. Election Reform, supra note 31, at 28.
DREs increases the cost and potential for mechanical malfunction.\footnote{130} Moreover, the paper trail effectively eliminates electronic recounts in favor of hand-recounts that would be exceedingly time-consuming and potentially more error-prone.\footnote{131}

Voter-verified paper trails also create questionable compliance with elections codes predicated on the use of a single ballot. The addition of the voter-verified paper trail in essence creates a second paper ballot to complement the electronic ballot, begging the question of which one should be used to count the vote. The recount provisions of the elections code are designed to facilitate a second count of the ballot to compare to the result of electronic tabulation, and do not designate which ballot—electronic or paper—should predominate.\footnote{132}

The California Secretary of State decided that "whether it includes a VVPAT [voter-verified paper audit trail] option or not, that the electronic vote should be the legally valid vote unless there is some sort of discrepancy between it and the permanent paper record," but "the paper record should be presumed to be more reliable than the electronic vote unless there is evidence it has been corrupted or is incomplete."\footnote{133} How the official record of the vote can change is not contemplated within the current elections code. Furthermore, allowing a secretary of state to arbitrarily change that designation depending on various circumstances of his or her own choosing seriously compromises the reliability of an election by essentially allowing the secretary of state to choose which set of results to prefer. Moreover, the secretary of state may invite countless legal challenges to an election by defeated candidates by publicly acknowledging that the electronic "legally valid vote" is officially less reliable than the paper record.

Ultimately, considerable support for voter-verified paper trails reflects a failure on the part of many officials to comprehend the paradigm shift in thinking that must accompany the change from paper ballots to paperless DREs. DREs can meet the demands of accountability and transparency, while accomplishing the auditing purpose of recounts, all through open source code without requiring a backward-looking paper solution. Elections do not require the needless formality of manual hand-recounts to achieve the purpose of

\footnote{130}{Id.}
\footnote{131}{Id. at 29.}
\footnote{132}{See, e.g., CAL. ELEC. CODE § 15630 (West 2004).}
\footnote{133}{CAL. SECRETARY OF STATE, supra note 23, at 38.}
a voting system: to count votes and allow for the collective democratic expression of public opinion. If the voters believe DREs achieve this purpose, as some studies suggest they do,\textsuperscript{34} then the voter-verified paper trail merely represents the attempts of an anxious mass eschewing self-criticism to cling to the vestiges of an archaic tradition that has no place in the digital age.

C. Open Source Makes Elections More Legitimate

Whether an election system succeeds or fails depends less on its ability to provide slightly higher reliability and more on whether voters perceive the system as legitimate. Open source code fosters legitimacy by mirroring the open and participatory system of democracy that elections are intended to facilitate. By allowing anyone to download and inspect the software, the entire population, either directly or through advocacy groups or the news media, could help ensure the code is error-free and not subject to security weaknesses.\textsuperscript{135} A testing group comprised of the entire voting public would obviate the need for mass reliance upon a few expert testers,\textsuperscript{36} and would replace faith in them with faith in one’s own inspection of the source code, or that of independent organizations or the news media.

Moreover, open source extends the concept of participatory democracy to one of the most fundamental elements of participatory democracy: voting. Like democracies themselves, the voting systems used by the public would only be as good as the collective experience and ability of the public to make that system trustworthy and effective, rather than the individual abilities of a few outside experts. Public involvement in the development and testing of the DRE source code would, in turn, create public responsibility and ownership, in both literal and figurative terms, over the source code that facilitates elections. Public responsibility and ownership would also eliminate the conflict of interest charges against manufacturers trying to influence the outcome of an election through legal means.\textsuperscript{137}

The world's first open source paperless DRE election in the Australian Capital Territory ("ACT") produced positive reactions

\begin{itemize}
\item[135.] See, e.g., Election Reform, supra note 31, at 26.
\item[136.] See, e.g., CAL. ELEC. CODE § 19206 (West 2004).
\item[137.] See, e.g., Warner, supra note 2.
\end{itemize}
from voters, suggesting that the negative press proprietary code-based DREs have received in the United States could have been avoided. Furthermore, the development process provides a model for how American DRE manufacturers and state governments could use a more open process based on open source code to enhance trust in the resulting voting system. Unlike American DREs, the ACT chose to run its system on open source software that was tested by a wide variety of people within and outside of government, in a highly public process. The testing process resulted in the identification of numerous errors, one of which would have seriously crippled the system if left in place.

In addition, the design team solicited advice and feedback on the system from representatives from the political parties, "special interest groups," and advocacy organizations. Thus, unlike their American counterparts who became opponents of paperless DREs, these groups in Australia were directly a part of the transition to electronic voting. By making these groups part of the development process, the ACT not only avoided turning the groups into opponents, but potentially gained public relations allies in efforts to win over the public. From the public's perspective, a broad-based coalition of groups developing the software negates the suggestion of government conspiracy—or worse, incompetence—that has plagued U.S. efforts.

D. Open Source Code Should Be Legalized

While opening the source code to the public would require very little effort, under current elections codes doing so is essentially "illegal." The Federal Elections Commission has promulgated, and the states have adopted, escrow provisions to lock up in secret any source code used to operate a voting system. Coupled with the use of compiled proprietary code, escrow provisions effectively lock the public out from access to the source code that determines the votes they cast. While these escrow provisions ostensibly act as part of the discredited "security through obscurity" model, they ultimately

139. Id. at 7-8.
140. Zetter, supra note 111.
141. ACT Electoral Commission, supra note 138, at 6-7.
142. See CAL. ELEC. CODE §19103 (West 2004). See also FEC, supra note 7.
143. See, e.g., Dill et. al., supra note 8, at § 2.2.
exist to protect the intellectual property rights and investment costs of proprietary source code developers.\footnote{See, e.g., Election Reform, supra note 31, at 26.}

Government may in many scenarios help companies protect their intellectual property when their systems are used for a public function, but in the case of elections, tipping the balance toward private financial interests at the expense of public access places profit ahead of democracy.\footnote{See discussion supra in Part IIIA.} Requiring open source code for computers performing all public functions might arguably stifle incentives for private development in certain markets,\footnote{See generally Evans & Reddy, supra note 28.} but that outcome is not readily apparent with electronic voting machines. For instance, the ACT put out bids for private companies to develop open source voting software.\footnote{ACT Electoral Commission, supra note 138, at 6.} In the United States, manufacturers make money from selling electronic voting machines to state elections agencies, not from the software in particular (although the Australian model suggests that companies can make money on open source election software). Moreover, many manufacturers rely heavily on commercial off-the-shelf software for their voting machines, suggesting that very little of the manufacturer’s intellectual property is protected in escrow.\footnote{Id. at 27.} Manufacturers could also make public the source code for the recording and counting of the votes, while retaining proprietary control over the user interface code that is arguably more important to a manufacturer in distinguishing their product from a competitor’s.

\section*{V. Conclusion}

With an important Presidential election hinging on vote results from DREs, looming Federal and state voting equipment funding deadlines, and increasing public pressure over DREs, government must decide how to solve this problem in relatively short order. To do so, it must cut through the conspiracy theories about election hackers and focus on the true source of the problem with DREs: proprietary source code. Proprietary code makes state agency testing of DRE software inadequate to assure the system will be error-free and safe from security flaws. In addition, the closed nature of proprietary code does not comport with the American tradition of openness in elections, based on transparent and accountable government.
Open source code provides a remedy to these problems by soliciting the much larger number of testers in the public who can leverage their combined knowledge and skills to create more reliable DRE software. More importantly though, the public process of testing and developing open source code gives the electorate ownership and responsibility for creating a reliable system, which in turn bolsters the resulting software’s legitimacy by allowing the public to affect its quality. In turn, that public review process will relieve state agencies from the burden of being the sole testers of the code, and allow them to focus on management and regulation of the review process for which they are better suited.

Using open source code will undoubtedly solve the technical problems with DRE software, and will probably restore the public’s faith in DREs, but it remains to be seen whether trust in the election system as a whole can be repaired by electronic voting after the disastrous 2000 Presidential election. Government must move quickly to counteract the voluminous negative press electronic voting has received. Moreover, even if open source software replaces proprietary code, the process by which government changes to open source may be just as important as using the open source code itself, as the successful collaborative Australian model suggests. Ultimately, if government is to succeed in its goal of restoring legitimacy to election, the electronic voting machines it deploys in several suspect Florida counties in November 2004 must perform flawlessly because an anxious nation will be watching.