

Transparency and Access to Source Code in Electronic Voting

Joseph Lorenzo Hall
UC Berkeley, School of Information

This material is based upon work supported by
the National Science Foundation under Grant No. CNS-0524745.

Defining Electoral Transparency

- Transparency and the process of electing are the foundations of a representative democracy.
- My working definition of transparency:
 - A fully transparent election system is one that supports *accountability* as well as *public oversight, comprehension* and *access* to the entire process.

Defining Open and Disclosed

- Open Source
 - The license vs. the development model
- Disclosed Source
 - Akin to VoteHere's License Agreement
- Disclosure is in line with transparency needs; other aspects are not necessarily.

Goals

- What role could access to voting system source code play in increasing the transparency of voting systems?
- What are the risks and benefits of open source and disclosed source regimes for security and the market?
- If open source code offers measurable benefits, what barriers exist to the development of open source software in the voting systems environment?

Goals

- What business methods from the landscape of open source software may translate to voting systems?
- Are there alternatives to public disclosure of code or open source code requirements that might yield similar benefits in technology performance and increased transparency, but minimize potential risks posed by source code disclosure?

Enclosure of Transparency

- Voting technology has experienced an “enclosure of transparency”.
- The requirements placed upon elections have only increased.
- Mechanization has had profound consequences, good and bad.
- We end up with a very opaque system.

Implications of Source-Availability

- More people can examine the code
- Can build w/ debugging flags set
- Automated evaluation tools
- Software is not enough
 - Need access to the system in its running environment.

Negative Effects of Enclosure

- Voters cannot prove to themselves that their vote was cast.
- Election administrators are poorly positioned to evaluate their systems.
 - Rely on Federal cert. or do their own
 - Due to proprietary concerns, even the evaluation process is opaque.

Efforts to Increase Transparency

- State Level
 - States starting to require escrow, disclosure
 - CA: 2003 mandate, ACR 242, AB 2097
 - NC statute, Wisconsin AB 627
 - VVPRs (28 States)
- Federal Level
 - Holt Bill (H.R. 550), Conyers Bill (H.R. 533)

Benefits and Risks of OS

- Benefits
 - Source code transparency is increased
 - Fewer IP claims; more competition
 - Many more people can evaluate the code
- Risks
 - Exposure of vulnerabilities to public
 - Need to be able to handle found flaws close to an election (postpone+patch, paper backup, etc.)

Benefits and Risks of DS

- Benefits (same as OS, but)
 - IP regime is different (better for vendors), limited derivative works (needed for some evaluations)
- Risks (same as OS, but)
 - With mandated disclosure:
 - No more source code trade secrecy in the market
 - Narrower licensing options for source code IP
 - Legal complications with unilateral disclosure by government (*Ruckelshaus v Monsanto*, reverse FOIA/PRA lawsuits).

Open Source and the Market

- Open Source e-Voting Projects
 - Australia's eVACS
 - Released under GPL, used in 2 elections
 - *Not* open source development
 - Will abandon GPL for next release (disputed)
 - Voting Solution's ChoicePlus
 - Open Voting Consortium / Foundation
 - Open Voting Solutions

Open Source and the Market

- Some models are probably inappropriate for voting systems.
 - Web services, customization
- Some could make sense given a body of OS voting system code.
 - System integration, targeted development, dual/multi-licensing, hardware sales

Barriers to OS Voting Technology

- Regulatory: Changes trigger system recertification at all levels; certification is of an end-to-end system.
- Economic: Certification, contractual performance bonds are expensive.
- Organizational: Other pieces of a voting system business outside of code development need to be in place to field a product.
- Perceptual: Customers might not understand the debate around disclosure and system security; need a proven track record to make a sale.

Alternatives

- Transparency furthering vehicles
 - Limited disclosure of code
 - Audience (w/ public results), Scope
 - Federal certification process
 - Technological mechanisms
 - VVPR/BMDs – public verification of an record independent of the system.
 - Narrowing what has to be evaluated (work presented today, for example)

Open Questions

- What further policy mechanisms will work towards heightened transparency?
- Solutions outside current market?
 - “Community Source”
 - Leveling the playing field for OS/DS
 - Incentivizing OS
- How are “qualified individuals” chosen?
- How do we maintain high-quality evaluation?

Open Questions

- How do vendors move towards a disclosed source regime?
 - Rewrite from scratch, release to increasingly broad constituencies? Vet and release pieces of existing code?
- Should there be a market here?
- Is public disclosure appropriate for voting system software?
- What is the theoretical basis for transparency in a representative democracy?