A Computer Scientist's Assessment of Online Voting

Steven Wolfman, Ph.D.
Department of Computer Science
University of British Columbia

E-volving Democracy and Online Voting
Saturday, 26 May 2012 from 2:00 PM to 5:00 PM (PT)
Vancouver, British Columbia

For me: intro with: not a computer security expert (but a CS expert, interested in elections and security); BELIEVE in the Internet to improve democracy, but not necessarily for voting.

Outline:

1. Thorniest issues: why CSists think online voting is a bad *bad* ***bad*** idea.
2. Bright side: what CS and online/electronic voting has to offer as positives.
3. (summary)
4. What would it take for me to trust online voting, were that possible?
5. Addendum: some items that seemed worth discussing once we heard from the panelists and audience.

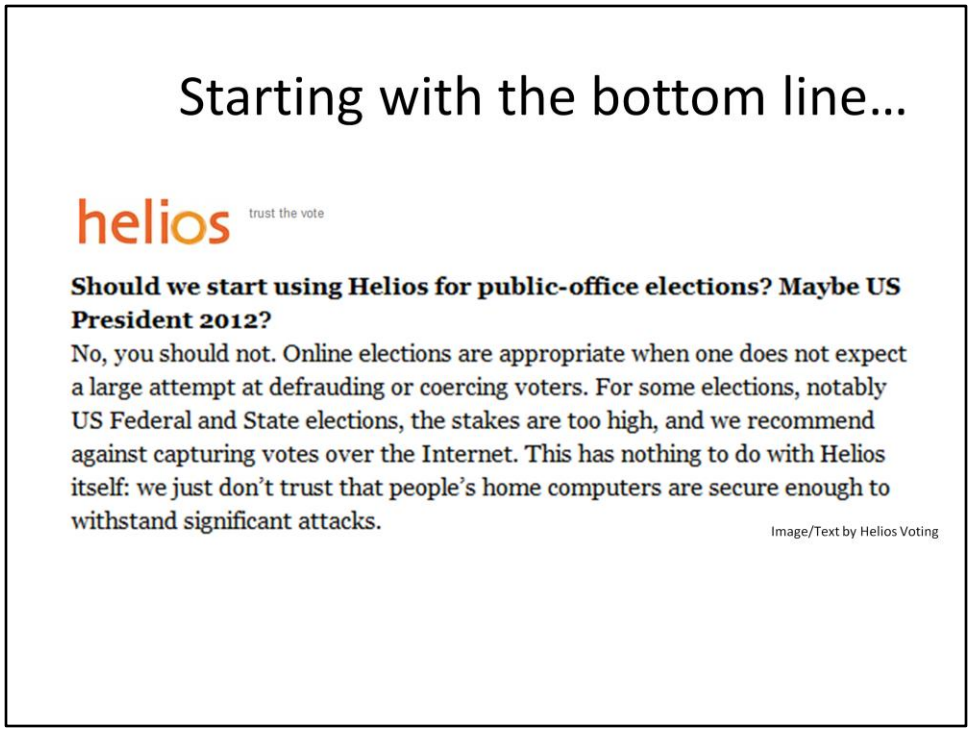
Good terms to toss in to educate people:

1. Attack surface: a way to think about security in terms of the points of attack available in a process (e.g., E2E increases the attack surface by making the process more complex).

2. Retail vs. wholesale fraud (**important**): retail fraud is the "storefront" level. I threaten/bribe you personally to coerce your vote. It requires a lot of legwork.

Wholesale fraud is fraud done centrally, requiring a "much small conspiracy" to achieve the same change in outcome in an election. A central problem with using computers and *especially* the Internet in an election is that it makes almost every step (almost the whole attack surface) subject to wholesale fraud. That's ***bad***.

ASIDE: I think proportional representation (e.g., party-list, STV (not my favorite), reweighted range, etc.) or good single-winner voting systems (e.g., approval, SODA, Condorcet-based, range, etc.) is ***more important*** than magically-secure online voting to have a positive effect on democracy.



Helios logo and quote from: http://heliosvoting.org/frequently-asked-questions/
This is copyrighted, but I believe its inclusion is under fair dealing.

Dutch Meyer (outstanding grad student in the Network, Systems, and Security lab at UBC CPSC) rough quote: "Computer Scientists are *arrogant*. We think we can do just about anything. And even we don't think we should do online voting."

ACM (and USACM) policy statement on electronic voting:

Background
Virtually all voting systems in use today (punch-cards, lever machines, hand counted paper ballots, etc.) are subject to fraud and error, including electronic voting systems, which are not without their own risks and vulnerabilities. In particular, many electronic voting systems have been evaluated by independent, generally-recognized experts and have been found to be poorly designed; developed using inferior software engineering processes; designed without (or with very limited) external audit capabilities; intended for operation without obvious protective measures; and

deployed without rigorous, scientifically-designed testing.

Recommendations

To protect the accuracy and impartiality of the electoral process, ACM makes the following recommendations:

· All voting systems -- particularly computer-based electronic voting systems -- embody careful engineering, strong safeguards, and rigorous testing in both their design and operation; and,

· Voting systems should also enable each voter to inspect a physical (e.g., paper) record to verify that his or her vote has been accurately cast and to serve as an independent check on the result produced and stored by the system. Making those records permanent (i.e., not based solely in computer memory) provides a means by which an accurate recount may be conducted.

Ensuring the reliability, security, and verifiability of public elections is fundamental to a stable democracy. Convenience and speed of vote counting are no substitute for accuracy of results and trust in the process by the electorate.

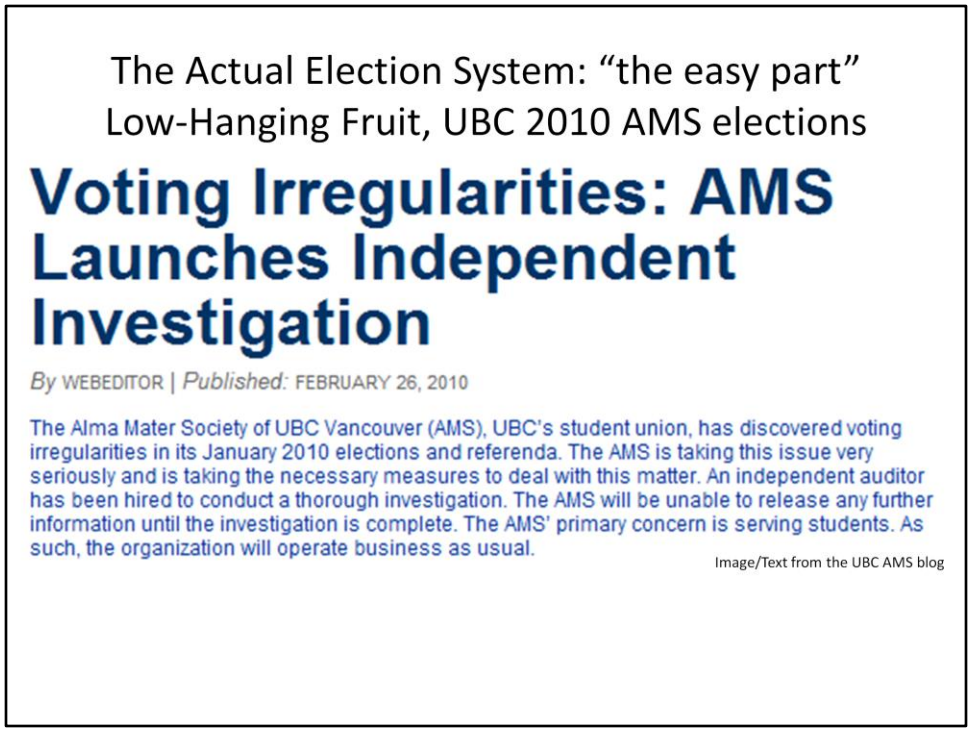
(This policy statement was adopted by both USACM and ACM's Council; therefore, it is an adopted position of ACM. More on this statement and the poll ACM conducted of its Members can be found on the weblog.)

ACM appended to this (for online voting):

Internet voting adds additional concerns about security, verifiability and auditability to those already known about electronic voting. USACM recognizes the difficulties faced by overseas and uniformed citizens when trying to vote, but cautions against voting systems that have no capability for a reliable post-election audit or recount. Delivering a blank ballot or blank registration form over the Internet poses fewer concerns than delivering completed ballots or registration forms; and would help reduce the time required to register and vote in accordance with local election laws.

There are steps that can be taken to reduce, but not eliminate, the risks associated with Internet voting. Using a dedicated Internet voting systems, like a kiosk system, where the computers are set up only for voting, can reduce security and reliability concerns. However, such systems need some means of preserving the ability to audit and/or recount the votes. At the present, paper-based systems provide the best available technology to do this.

The Actual Election System: "the easy part"
Low-Hanging Fruit, UBC 2010 AMS elections



Voting Irregularities: AMS Launches Independent Investigation

By WEBEDITOR | *Published:* FEBRUARY 26, 2010

The Alma Mater Society of UBC Vancouver (AMS), UBC's student union, has discovered voting irregularities in its January 2010 elections and referenda. The AMS is taking this issue very seriously and is taking the necessary measures to deal with this matter. An independent auditor has been hired to conduct a thorough investigation. The AMS will be unable to release any further information until the investigation is complete. The AMS' primary concern is serving students. As such, the organization will operate business as usual.

Image/Text from the UBC AMS blog

Let's start with a local example.

The screenshot/article/quote is from: http://www.ams.ubc.ca/2010/02/voting-irregularities-ams-launches-independent-investigation/
Again, I believe this use is under fair dealing.

Here I mean it's the "easy part" for the election system designer trying to secure an election system. You get to design it, you get to control it.. At least to an extent.

There are ***many*** problems beyond those I list here. (For example, say you have a cryptographic signing scheme plus legally mandated process in place to ensure that the software you distribute to voters as your voting app is the software you ***believe*** it is… but then at the last minute you discover a critical flaw in the software and believe you have a solid fix. Should you distribute the signed version or hot-swap the new software in? The US state of Georgia, for example, has faced this issue in their DREs. Estonia did in their 2011 election as well (see the OSCE/ODIHR report).)

Emphasize: HARD (impossible, in a sense) to secure these systems, but probably (maybe?) plausible to make attacks sufficiently expensive/complex and bugs sufficiently rare/recoverable. (But, the e-voting record is **not** promising!)

http://blogs.ubc.ca/ubcinsiders/2010/03/12/ams-electoral-fraud-the-technical-perspective/

The Voting Process (in techspeak)

The actual exploit itself is frustratingly simple. In fact, I'm frankly irritated I didn't discover the hole myself and point it out before voting closed. Bear with me a bit if things get technical, I'll do my best to make things clear.

Note: This information is based off my understanding of the system, the presentation from the EC at AMS Council, and some questions asked of the CRO. No guarantees are provided regarding the specific details of the exploit.

When a voter logged into the system, it asked for their CWL information. Their CWL username and password were then sent to UBC's authentication server, which then responded with information indicating if the login was successful as well as details about the student. One of these details is the user's student number. This number was then cross-referenced with a list provided by the Registrar's office to ensure the user was a valid AMS member who was eligible to vote.

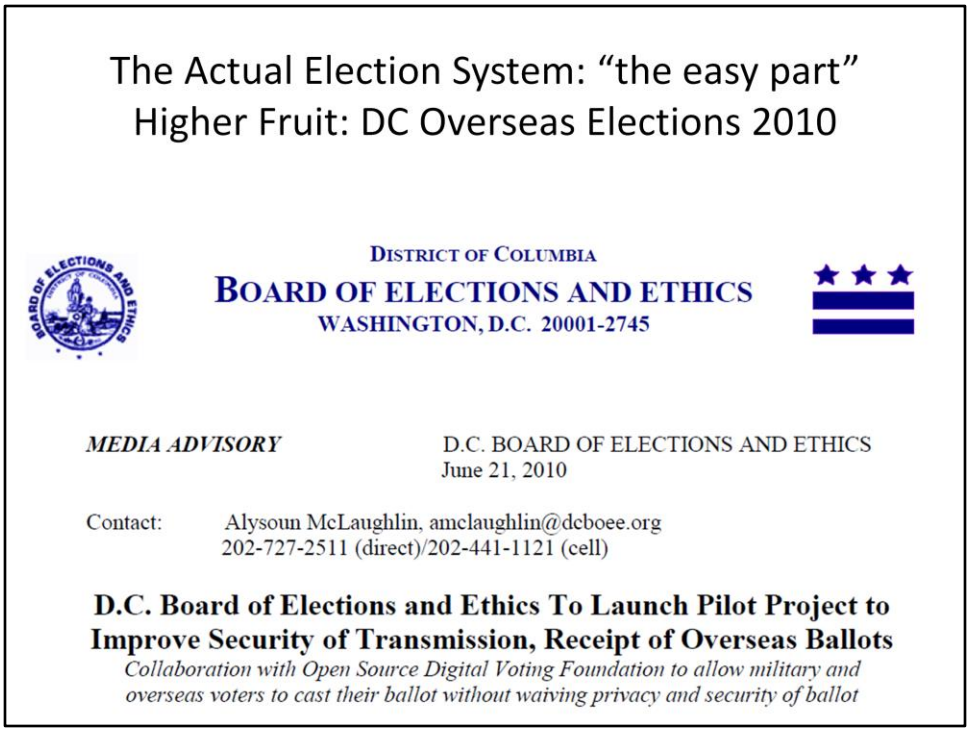
At this point, the system has determined that the student should be able to vote, and thus displays the ballot page. The user then fills out their choices on the page and clicks "Submit". The data entered in the form is then sent to the server where it is presumably validated and stored in a vote database. From comments made during the EA's presentation, we also know that a user's IP address and student number are stored along with their vote in the database.

The Problem

While the above might seem like a logical, straightforward, and superficially secure method of balloting, it suffers from one debilitating flaw. Specifically, validation of the voter is done before displaying the ballot, and *not when submitting the vote to the database*.

This means that all someone had to do was save the ballot page to disk, and they could submit it as many times as they liked! Yes, folks. It was *that* simple.

Of course, it would also be relatively trivial to create a script which would post ballots repeatedly to the server, thus allowing one to specify a desired degree of manipulation without doing any of the tedious forgery by hand. Since we know *at least* 731 invalid ballots were cast over a 4-hour span, I find it likely that a script was used in this process. Fortunately for us, the author of said script didn't feel it was necessary to hide his submissions and they were thus noticed with relative ease.



The Actual Election System: "the easy part"
Higher Fruit: DC Overseas Elections 2010

DISTRICT OF COLUMBIA
BOARD OF ELECTIONS AND ETHICS
WASHINGTON, D.C. 20001-2745

MEDIA ADVISORY

D.C. BOARD OF ELECTIONS AND ETHICS
June 21, 2010

Contact: Alysoun McLaughlin, amclaughlin@dcboee.org
202-727-2511 (direct)/202-441-1121 (cell)

D.C. Board of Elections and Ethics To Launch Pilot Project to Improve Security of Transmission, Receipt of Overseas Ballots

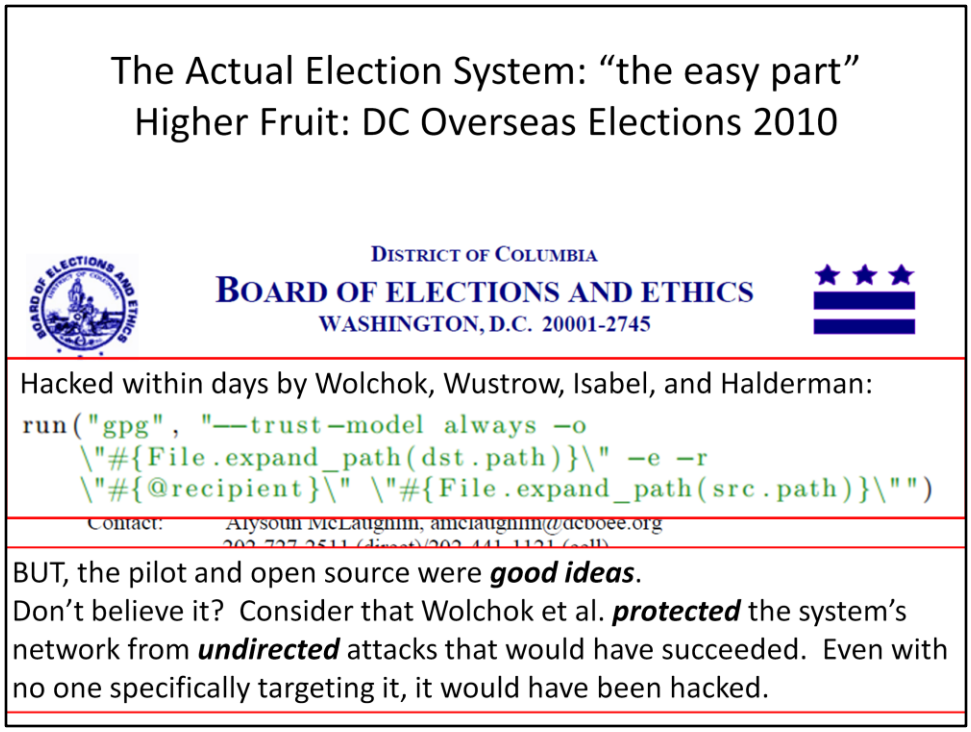
Collaboration with Open Source Digital Voting Foundation to allow military and overseas voters to cast their ballot without waiving privacy and security of ballot

Image clipped from: http://www.dcboee.org/popup.asp?url=/pdf_files/nr_423.pdf
Again, I believe this use is under fair dealing. Furthermore, this media advisory was obviously intended for reprint.

Now, how about something with a bit more clout. DC attempted to help overseas voters (particularly military) vote online. This is a huge problem where the "increased turnout" is about helping people who otherwise ***cannot*** vote because logistical issues delay their vote too long to count.

[Note: This is the continuation of a project killed by Computer Scientists' assessment back in 2004 (the SERVE project).]

The DC BOEE did ***the right thing*** by having an adversarial testing period and open source (+ open design, at least partially). (They should have held it further before the actual (cancelled) deployment and they should have announced it further before it happened and they should have had legal protection in place for "black hat" attacking teams, but they did do the right thing.)



Information and screenshot of code from the paper:
https://jhalderm.com/pub/papers/dcvoting-fc12.pdf
Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. "Attacking the Washington, D.C. Internet Voting System". In Proc. 16th Conference on Financial Cryptography & Data Security, Feb. 2012.

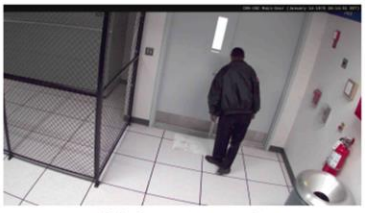
Unfortunately, the system was not good enough. A "shell injection attack" (putting a command inside some user input, in this case the extension of a file name being something complex not .pdf as expected) allowed the attackers tremendous control over the system. Note that single quotes (at the \" point) rather than double would have fixed this particular issue. Non-tech folks and tech folks both seem to like that point ☺

The comment from the bottom half is actually relevant to a *different* attack (not the one illustrate in the top box), but still worth thinking about.

There are ***many*** other reasons to use open source and adversarial testing. Wolchok et al. make a great one when they point out that people like them aren't going to do the social engineering (bribery, trash sifting, etc.) to get the source code, but the "bad guy" certainly will.

The Actual Election System: "the easy part"
Higher Fruit: DC Overseas Elections 2010
P.S. How Badly Hacked? All the votes, plus a bit..

(a) Voting server rack



(b) Security guard

(c) Typical workers, before attack

(d) Workers, after learning of attack

overseas voters to cast their ballot without waiving privacy and security of ballot

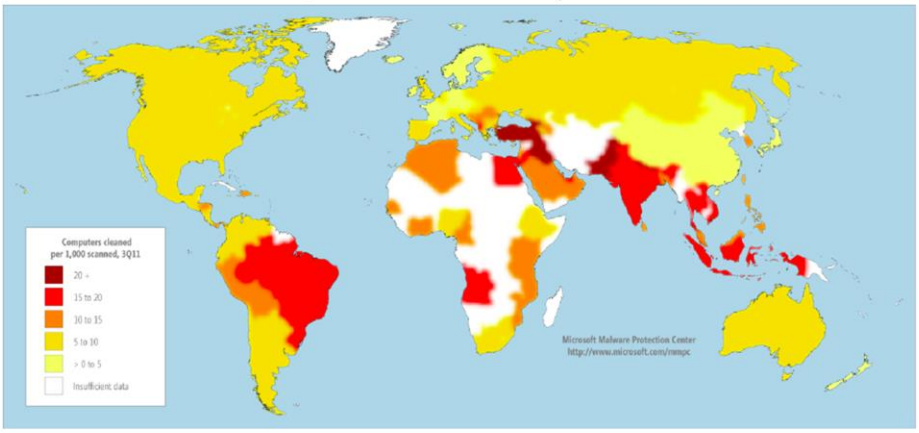
Really just here to emphasize to a non-technical crowd how serious such a compromise can be. These are shots from the security cameras, repurposed by the hackers to monitor the employees of the DC BOEE. Oops!

(Again, this came from the SECOND exploit described in the paper, not from the shell injection attack illustrated on the previous page.)

Also, note that just fixing the vulnerability used doesn't fix the problem. The attackers ID'd quite a number of different entry points that might have panned out to what they needed.

Client-Side Security: The HARD Part (your phone, my laptop, etc.)

Microsoft 2011 3rd Quarter malware detected per 1000 runs:



So, just get people to stop installing this stuff, right?

Stefan Savage pointed me at this report: http://www.microsoft.com/security/sir/default.aspx, in particular http://download.microsoft.com/download/C/9/A/C9A544AD-4150-43D3-80F7-4F1641EF910A/Microsoft_Security_Intelligence_Report_Volume_12_English.pdf

(Discuss retail vs. wholesale about here. See notes on opening slide.)

If the election system is the easy part, what's the hard part? ***Your phone, my laptop, etc.***

Here we can see an estimate (radical *under*estimate, I'd say) of the malware penetration across the world. Canada was pretty consistently around this time running at about 5 per 1000. Not too many computers, right? Well, first, it's an underestimate. Second, think about the tiny margins of victory that completely swing elections, at least in first-past-the-post systems like Canada has (and with modern "efficient" political canvassing).

How do you control the environment in which the user sees and manipulates the ballot? When you click the Hippo Party, did they really get your vote, or did the malware masking the real election system send a different vote? (Even with E2E, this can be an issue with a not-yet-satisfactory solution.)

Maybe an even better metaphor: imagine the “time-honored” practice of lying to people (e.g., with robo-calls) about when and where you can vote moved over to malware. You may have a *perfect* election system all set up, but if I can trick enough voters into missing the window during which they can vote, what do you do?

Stefan said this is the best reference he’s seen to give *some* kind of vaguely reasonable quantitative data on the prevalence of malware. BUT, note that (1) Stefan says there’s no methodologically defensible number out there that he’d subscribe to, (2) this is almost ***certainly*** a radical underestimate (e.g., not all computers are getting scanned, and the ones that are getting scanned seem less likely to be infected and (obviously) this doesn’t count malware that Microsoft’s detector does NOT find), and (3) StuxNet showed that a really organized, committed, well-funded group (whoever they were) can go head-and-shoulders above the run-of-the-mill malware.

Kanich, Kreibich, Levchenko, Enright, Voelker, Paxson, and Savage spam study in CACM 2009.

The lure to make more 'bots. Would ***you*** click?

Table 2. Filtering at each stage of the spam conversion pipeline for the self-propagation and pharmacy campaigns. Percentages refer to the conversion rate relative to Stage A.

Stage	Pharmacy		Postcard		April Fool	
A—Spam Targets	347,590,389	100%	83,655,479	100%	40,135,487	100%
B—MTA delivery(est.)	82,700,000	23.8%	21,100,000	25.2%	10,100,000	25.2%
C—Inbox delivery	–	–	–	–	–	–
D—User site visits	10,522	0.00303%	3,827	0.00457%	2,721	0.00680%
E—User conversions	28	0.000081%	316	0.000378%	225	0.000561%

No? Good for you, but enough will.

Images from http://cseweb.ucsd.edu/~savage/papers/CACMSpam09.pdf
Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage. "Spamalytics: An Empirical Analysis of Spam Marketing Conversion". Communications of the ACM, Sep 2009, 52:9(99-107).

Here's a bit of data from their analysis of one (real, reasonably large) spam campaign they infiltrated. This one was focused on making more 'bots (i.e., infesting more computers w/malware). Of the people who visited the silly banana site, 10% actually installed/ran the software. (Based on their experiments, may not have been a lot less than 10% of those who received the mail.. Hard to tell.)

This isn't quite the illustration of this that I'd like, but it ***is*** important to remember that "me doing a good job of cleaning up my yard" doesn't solve the problem. Everyone would need to, or at least an overwhelming percentage of the population. And do it consistently, keeping up with the latest threats and solutions.

Fixed the Server? Fixed the Client?
Then, fix the Internet.

Table View High School, polling station for the suburb of Flamingo Vlei; by Wikipedia User:Anrie, CC-BY-SA

A sort of "real-world" Denial of Service (DoS)

Distributed DoS:

Cheap 'botnets (e.g., infected user computers) attacking system infrastructure.

The image is under CC-BY-SA, satisfied by license on this document and by the citation above.

As of now, there's not a lot of ways around this one.

Interestingly, one of the most famous DDoS attacks was against Estonia (not their voting system, I think) in 2007.

Fixed Everything? Not forever. Cryptographic "Expiration Date"?

Key recovery attacks [edit]

Attacks that lead to disclosure of the key.

Image/Text from Wikipedia Block Cipher Security Summary

Cipher	Security claim	Best attack	Attack date	Comment
AES128	2^{128}	$2^{126.1}$ time, 2^{88} data, 2^{8} memory	2011-08-17[1]	Independent biclique attacks
AES192	2^{192}	$2^{189.7}$ time, 2^{80} data, 2^{8} memory		
AES256	2^{256}	$2^{254.4}$ time, 2^{40} data, 2^{8} memory		
Blowfish	2^{448}	4 of 16 rounds	1997[2]	
DES	2^{56}	2^{56} time	1998-07-17[3]	Broken by brute force, see EFF DES cracker. Off-the-shelf hardware is available for $10,000.[4]
Triple DES	2^{168}	2^{113} time, 2^{32} data, 2^{88} memory	1998-03-23[5]	
KASUMI	2^{128}	2^{32} time, 2^{26} data, 2^{30} memory, 4 related keys	2010-01-10[6]	The cipher used in 3G cell phone networks. This attack takes less than two hours on a single PC, but isn't applicable to 3G due to known plaintext and related key requirements.
Serpent-128	2^{128}	10 of 32 rounds (2^{89} time, 2^{118} data)	2002-02-04[7]	Linear cryptanalysis
Serpent-192	2^{192}	11 of 32 rounds (2^{187} time, 2^{118} data)		
Serpent-256	2^{256}			
Twofish	$2^{128}..2^{256}$	6 of 16 rounds (2^{256} time)	1999-10-05[8]	

(P.S. If we could only be sure how long this would take… I'm not sure it's a bad thing.)

Image from: http://en.wikipedia.org/wiki/Block_cipher_security_summary. Red indicates a practically "broken" cipher. Yellow is a "theoretical break", which essentially means it's untested, but also means in this case that it reduces the value from the security claim (dramatically in many cases) but not to levels that could plausibly be attacked with brute force for quite a while without radical advances or more breaks.

Finally, it wouldn't be that hard to record ***all*** (or at least an overwhelming majority of) the ballots in an Internet-based election. If you did a good job, I have no idea what they mean for now because they're encrypted.

But.. What about in 10 years? 20 years? 100 years?

I actually think this is fascinating. If we could control this, set a time-release for when the votes become available, it'd be utterly cool from an historical standpoint!

But.. We can't. We don't know when someone will crack a standard. (DES's cracking was somewhat predictable because it was brute force, but what if someone finds a flaw or we perfect quantum computation?)

This ties into privacy legislation in bizarre ways.

(It might also be a fun exercise to think about the current (as of 2012) proposal for Chinese ownership of a big chunk of the Canadian communications infrastructure in these terms.)

The Bright Side
(which doesn't resolve the issues above)

See your vote count in 3 easy steps...

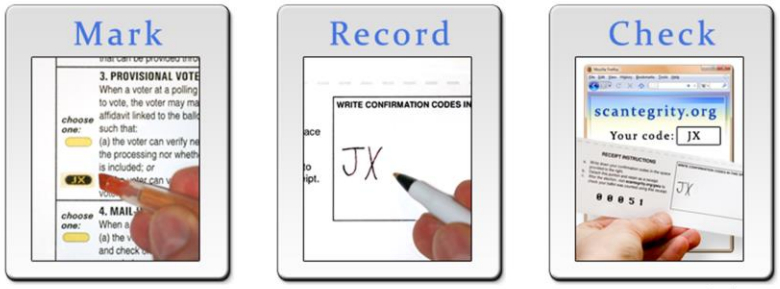


Image by the Scantegrity Project

End-to-End Verifiability: Scantegrity
(usable with standard optical-scan polling)

Image from http://www.scantegrity.org/, http://www.scantegrity.org/images/default/mark.png.

This image is under: http://www.scantegrity.org/wiki/index.php/GNU_Free_Documentation_License, NOT Creative Commons. Apologies if I screwed up including this in a CC document. ☹

NOTE: Audience Q here: can we convince people (sufficient technical literacy) to accept an E2E system? (Dunno.. But current election system is VERY complex!)

OK, so what's the good news?

Well, for one there's end-to-end verifiability, an absolutely foundational shift in how we might think about elections, the secret ballot, and trust of election administration authorities.

First (which I didn't do at the panel, oops!), what's E2E?

Quick version: E2E means everyone, everywhere can perform their own *independent* audit of an election. (That overstates a bit. Enough people really do have to check the receipts they get (or hand them off to another person or an organization that does the checking), and those checks are "independent" but also per-ballot.)

When you vote, what do you get to take away with you from the ballot box to prove how you voted?

Nothing!

Why not? It's the fundamental advance in voting that is the secret ballot. The secret ballot means that the election system is resistant to coercion. Boss B can grab you outside the polling place and threaten to break your kneecaps if you don't vote for Party B. You can go in and vote for Party A (party X?) and come out and tell Boss B that you really did vote for Party B. How can he tell? In a well-implemented secret ballot, he cannot. (Similarly, you can sell your vote (to multiple buyers supporting different candidates!), but cannot provide proof to them that you voted as you said you did. This is why cameras are banned from the ballot booth, BTW.)

Unfortunately, you then have to put a lot of trust in the counting system. That's because you cannot see someone, for example, write up a tick on the whiteboard when you say "I vote for Party A." In that situation, everyone can see that your vote was counted as it was cast.

E2E is a classification of (cryptographic) voting systems in which the voter gets to take a receipt away (JX above) that they can use to ensure that their vote was recorded as they cast it, but which they CANNOT use to prove to anyone else how they voted. (Additionally, the "decryption" process is verifiable by anyone from the posted ballots. The actual verification often requires CS/math expertise, but there are *at least* thousands of people in Canada qualified to do it, and each of them can independently perform the verification if they desire.)

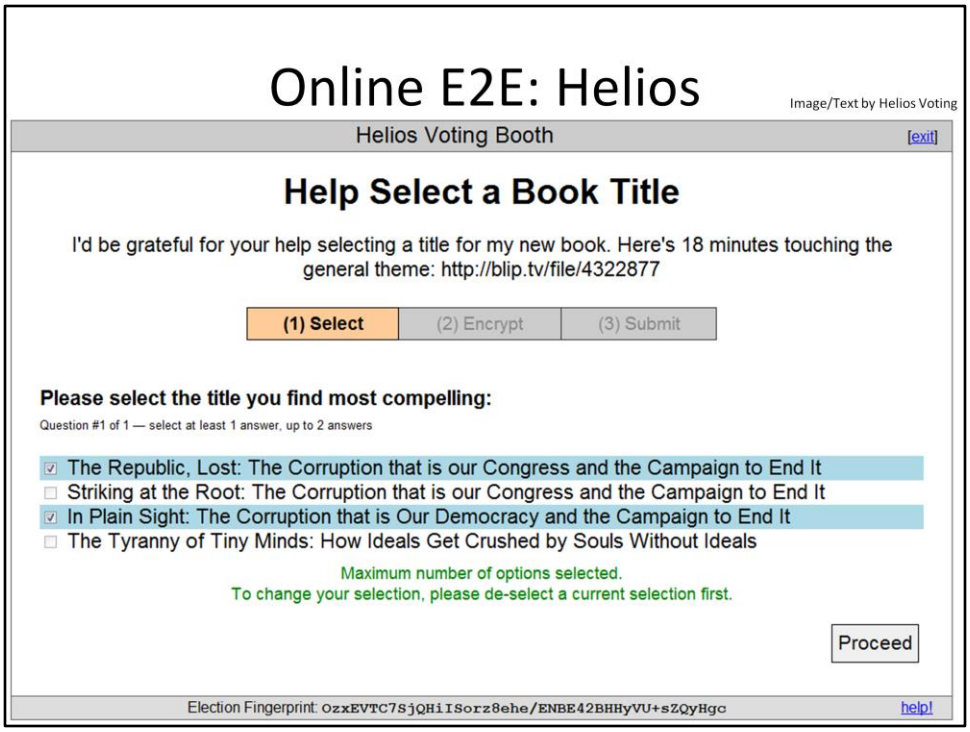
E2E is *awesome* in a paper-based election, because it gives you the best features of the secret and public ballots.

(Discuss only if needed: The voting authority has the cryptographic keys necessary to associate a voter (actually, a voter's receipt) with their identity. However, in a well-run system, you get a bunch of mutually distrustful parties (like the Liberals, NDP, and Conservatives plus Party X, FairVoting BC, UBC, etc.) to each take a "share" of the key. It's then set up so that unless, say, 13 of 20 key share holders ALL agree, no process requiring the key can proceed.)

All of this does mean that E2E adds nothing to the vote (and maybe detracts a bit) unless you have some basic level of trust in your government (i.e., as Christian Bull at WOTE 2011 roughly put it: you might think there's conspiracies out to get you in your government, but you don't think your government (including the political side and the bureaucracy) ***is*** a conspiracy out to get you). I firmly believe this of Canada.. Furthermore, if you DO NOT believe this, you probably don't see much way besides revolution to get to where democracy needs to be anyway.

Scantegrity, pictured here, has been used in one small governmental elections. Several such systems have seen use elsewhere. "Partial E2E" (if that's not an oxymoron) was used in Norway's online voting recently, per Christian Bull at EVT/WOTE 2011.

Online E2E: Helios



We can also do this online, ***BUT*** there's some hitches.

First, the client-side concerns above still stand. (Helios offers a verification step that can potentially get around this, but doing it right is somewhat painful.)

Second, there's ***no*** defense whatsoever against coercion attacks like those usable in vote-by-mail. (True of almost all online voting. Revoting (as in the Norwegian system) offers some defense.)

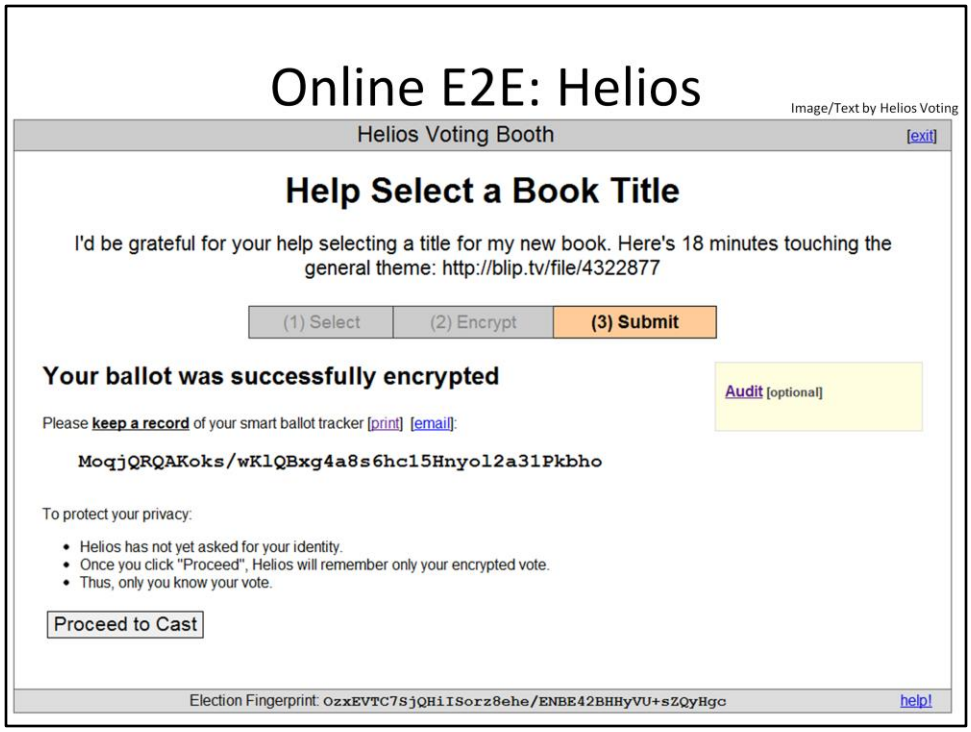
Aside: note the improved usability (which has been documented in other studies, see especially Voting Technology: The Not-So-Simple Act of Casting a Ballot). I cannot select additional options, when I select the second option, the message telling me I can select up to two changes to tell me I can't select more and tells me why. If I do try to select more, I assume (but don't know.. didn't check) that the system gives me specific feedback showing what the problem is. This is another ***big*** advantage of electronic voting.

Improved usability ("checked enough" helios shot?) and accessibility (Scytl w/audio hookup shot??)

Participation, MAYBE (leave to Fathima?) (25% of votes in Estonian 2011 election were online; 75% traditional)

Speed/precision (assuming system ***works***, recounts don't change result… mostly)

Online E2E: Helios



Here's my receipt.

If I click the audit button, it invalidates my ballot but allows me to, effectively, check if all was kosher to this point. I have not yet reviewed this process in detail to decide whether I believe it offers resistance to client-side attacks (but Ben Adida doesn't seem to think it *really* does, based on his comments at EVT/WOTE 2011 and the FAQ on Helios, where he's a (the, I think) principal designer).

More Bright Side

- Usability:
  reduced over/undervoting, multiple languages, cleaner and clearer ballots (worse in US), etc.
- Accessibility:
  audio systems, voting from home/care location, etc.
- Participation:
  not so well-established yet
- Speed/Precision:
  if we get it working, it works fast and consistently

Summary

- Servers are *hard* to protect
- Clients are *nearly impossible* to protect
- The Internet, as currently designed, is *impossible* to protect.

So Helios (or Scytl or …) is the Answer?

No. They're many steps in the right direction, but far from there. Helios was hacked from the client side, and remember…

helios trust the vote

Should we start using Helios for public-office elections? Maybe US President 2012?

No, you should not. Online elections are appropriate when one does not expect a large attempt at defrauding or coercing voters. For some elections, notably US Federal and State elections, the stakes are too high, and we recommend against capturing votes over the Internet. This has nothing to do with Helios itself: we just don't trust that people's home computers are secure enough to withstand significant attacks.

Image/Text by Helios Voting

What Would It Take?

- End-to-end verifiability and auditability
- Available well-managed polling stations/kiosks
- Open source and black hat testing
- Coercion resistance? (Via revoting? Hard w/E2E.)
- Clear technical, legal, and logistic plan for aggressively seeking fraud and recovering from it

Addenda to Panel Discussion

One Other Key Report:

National Academies Press (NAP): "Asking the Right Questions about Electronic Voting"

Canadian Organizations:

The Canadian equivalent to NAP is the Council of Canadian Academies, which has not weighed in on voting at this time.

ACM is an international organization, but a "Canadian version" might be CIPS: the Canadian Information Processing Society. (Has not weighed in on voting at this time, as far as I know.)

Bonus:
Online Banking vs. Online Elections

From Wolchok, Wustrow, Isabel, and Halderman's paper on the DC BOOEE election hacking:

Comparison to online banking While Internet-based financial applications, such as online banking, share some of the threats faced by Internet voting, there is a fundamental difference in ability to deal with compromises after they have occurred. In the case of online banking, transaction records, statements, and multiple logs allow customers to detect specific fraudulent transactions and in many cases allow the bank to reverse them. Internet voting systems cannot keep such fine-grained transaction logs without violating ballot secrecy for voters. Even with these protections in place, banks suffer a significant amount of online fraud but write it off as part of the cost of doing business; fraudulent election results cannot be so easily excused.

Text/screenshot from the paper: https://jhalderm.com/pub/papers/dcvoting-fc12.pdf
Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. "Attacking the Washington, D.C. Internet Voting System". In Proc. 16th Conference on Financial Cryptography & Data Security, Feb. 2012.