CS 261 Week 12 Notes - E-Voting

Eric Kim

November 14, 2011

1 Why do Electronic Voting?

We want to do Electronic Voting (E-Voting) because it gives us operational efficiency. It's much easier to do an electronic election (say, using a DRE system) than a paper-based ballot, since a paper-based election has a lot of overhead (such as printing the ballots, making sure there are enough paper ballots, transporting the paper ballots, etc). In addition, E-Voting allows voting to be more accessible to those with special needs (say, vision-impairment).

In the past, those with impairments (motor, visual, etc) would have another person (friend, spouse, etc.) perform the vote for him or her. This isn't desirable, since the system is depriving that impaired voter of anonymity and independence. With an E-Voting system however, there can be special machines that are geared towards allowing those with impairments to still successfully cast a vote.

2 Why is E-Voting hard? (the 'voting' problem)

2.1 Problem 1: The Secret Ballot problem

All votes should be anonymous and untraceable - in addition, there shouldn't be a way to prove that a voter voted in a certain way. We want this in order to avoid vote coercion or blackmailing - consider a scenario where Bob is trying to force Alice to vote for Chuck. If Bob is unable to verify that Alice voted for Chuck, then it is impossible for Bob to force Alice to vote for Chuck, since Bob will be unable to verify Alice's vote.

So, we just can't maintain voting logs.

2.2 Problem 2: Confidence

We must be confident in the election results - more importantly, 'ordinary' people (i.e. non Ph.D people) should be able to hold confidence in the voting system. For instance, a voting system that uses a crazy amount of complicated cryptography would seem questionable to an ordinary member of the public. A voting system that requires experts to understand/verify the system isn't ideal.

Another problem with election confidence is that the election officials in charge are typically appointed by the party in power - thus, there is a potential conflict of interest there.

3 The (Initial) Rise of E-Voting

During the Florida 2000 election, the punch-card voting systems used caused a lot of highly-publicized problems. As a result, Congress allocated $4 billion in order to upgrade voting systems.

Since the money was only to be used within a certain time frame, DRE's (Direct Recording Electronic) systems were widely deployed. However, they were deployed before many computer scientists got involved with the systems.

4 The Paper Arrives (2006)

Just after many jurisdictions bought the DRE systems (with the Congress money), the Princeton paper was published in 2006. The people behind the paper managed to get their hands on one of the Diebold DRE systems (via a pretty shady means), and exposed many serious vulnerabilities with the system. Such vulnerabilities included the ability to rewrite the bootloader, steal/modify votes, and spread the malicious software via a DRE virus. Further follow-up surveys on other systems discovered vulnerabilities.

5 The Fall of E-Voting

The shift towards E-Voting systems have since reversed. Before Congress allocated $4 billion, E-Voting usage was about 33% . After the Congress allocation, the usage jumped to 66% - however, now it's back down to 33% . In addition, there has been a significant uptick in vote-by-mail votes. For instance, Oregon and Washington only has vote-by-mail. They claim that vote-by-mail elections are cheaper to perform.

6 ThreeBallot Voting System

Ron Rivest developed a ”cryptographic end-to-end voter verifiable” voting system called the ThreeBallot Voting System. It was an attempt to allow the voter to verify that his/her vote was tallied correctly, while providing verifiability and anonymity. However, it had some privacy issues (whoops).

7 Cryptography

We'd like to prevent undetectable large-scale vote stealing. Assume that we have a trusted voting official with a public key K_0 .

Say that each DRE machine publishes the encrypted votes (encrypted with public key K_0) $E_{K_0}(v_i)$ in a random order. How can we verify that the votes were correctly tallied?

7.1 Random Sampling

7.1.1 Analogy: Drug-Lord

Say you're a drug lord, and you're about to do a big, $100K deal. At the meeting place, the other party comes with 5000 stacks of $20 bills. You don't have enough time to check each $20 bill (the cops might come!), so instead you randomly select a small subset of the bills, and verify those. With a certain confidence interval, you can be sure that the bills are good.

7.1.2 In the voting system

When the user votes, the machine spits out a mag-strip card that contains $E_{K_0}(v)$. The user can then either accept it, or ask for the vote to be verified. In other words, can the system prove to me that the encryption was done correctly?

The hope is, if small fraction of the voting population asks to verify the vote (say, 10%), then with high probability any false encryptions (i.e. vote tampering) can be detected.

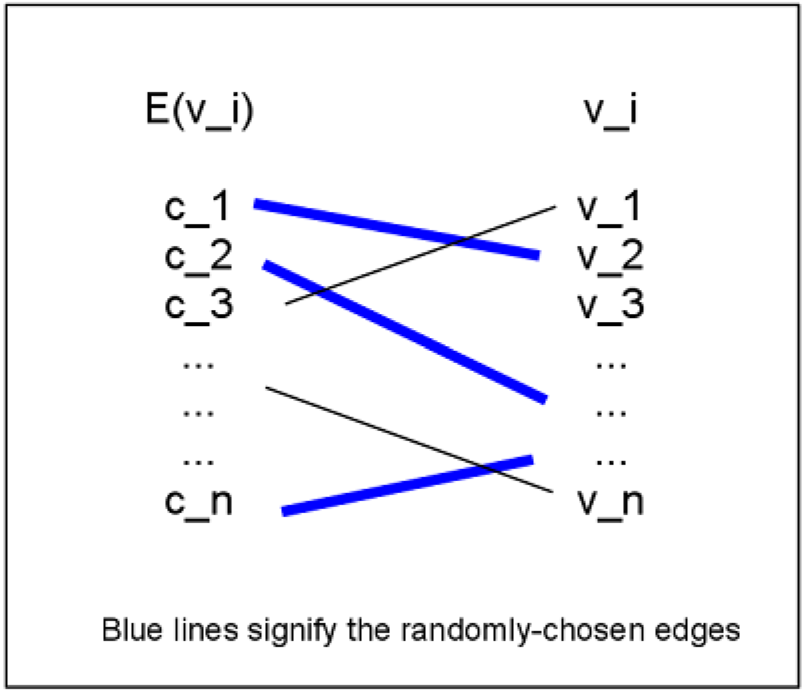
Note that, in order to preserve anonymity, if the voter asks for proof, then the vote is canceled, so the voter has to re-vote.

7.2 How to prove that the encryption was done correctly?

Any cryptographic encryption function $E_{publickey}$ is a deterministic function of the form: $E_{publickey}(v) = f(publickey, v, r)$, where r is the source of randomness. So, if the machine gives us r, then we can compute our own $f(p, v, r)$ to verify the machine's output $E_{publickey}(v)$.

7.3 How to tabulate $\{E_{K_0}(v_1), ..., E_{K_0}(v_n)\}$?

The trick is being able to decrypt in a verifiable way, in addition to preserving anonymity. The idea is: given the ciphertexts of the votes $\{c_1, ..., c_n\}$, first randomly permute the array of ciphertexts (to preserve anonymity). Then, the election official decrypts the ciphertext to an array of votes: $\{v_1, ..., v_n\}$. Then, the election official creates a (private) mapping from each c_i to its corresponding v_i . The voting official then randomly selects 1/2 of the edges (i.e. mappings from c_i to v_i), and provides proofs that each $c_i = v_i$ (same proof as from earlier).

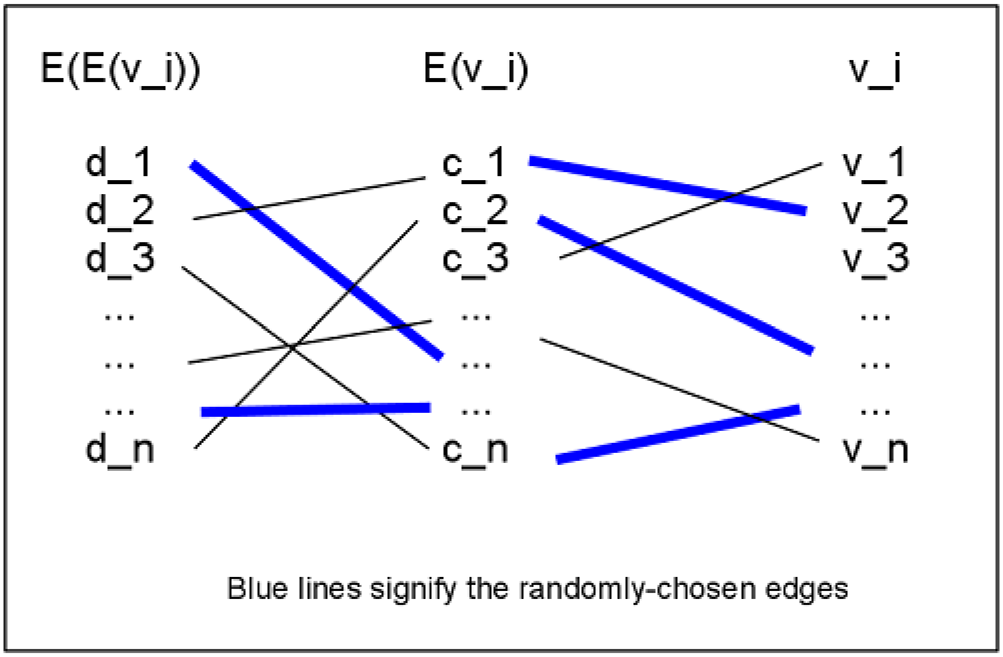


While this system offers great integrity (provided you completely trust the voting official), it means that only 1/2 of the votes get anonymity. Bummer!

The solution is to "chain" the above scheme:

7.4 A chain of indirection

Have the election official encrypt each c_i to get an array $\{d_1, ..., d_n\}$. Then, choose 1/2 of these edges such that there doesn't exist a path from a d_i to a v_i (in order to provide anonymity).



To choose the random 1/2 edges, you could either hold a public lottery, or a crypto-lottery.

7.5 The Catch

The problem is: the average person can't understand this system! So, the related question is: is it OK to have a system that is only verifiable by experts? Also, this system assumes that the voters are well-behaving (and competent) - what if a voter falsely claims that they've been cheated (i.e. they falsely claim that a vote proof was invalid)? Who do we trust?

7.6 Voke Revoking

With an in-person vote, once you hand in your vote it's done, since your name is not on the voted ballot. However, with mail-in votes, revocations are possible, since your name is on the envelop itself.

8 The Failure of the DRE Systems

An obvious question to ask is: why did the officials buy DRE's in the first place? Part of the reason is, vendors tend to become the election official's only friends, since officials tend to draw a lot of flak and criticism from the public. So, officials have grown to put a lot of trust in the vendors - thus, when a vendor (like Diebold) claims that their DRE system is "super safe and great", an official is likely to trust the vendor and purchase the systems.

In general, markets can be dysfunctional if buyers can't tell if a product is secure or not.

8.1 Economics Example: "Market for Lemons"

Say there is a used car lot - 1/2 of which are "lemons" (poor-quality cars), and the other 1/2 of which are "plums" (great-quality cars). A "lemon" is worth $4000, and a "plum" is worth $12000. Say the seller knows if a car is a lemon or a plum, but the buyer doesn't know.

When a buyer Bob walks into the lot, he doesn't know whether a car is a lemon or a plum - therefore, his best bet is to pay at most $8000. While this is the best possible action for Bob, this is bad for plum-owners - they can't sell their car to used-car lots, since they'd lose money from their $12000 investment! On the other hand, this situation greatly benefits lemon owners, since they can actually make a profit from their sucky cars! Thus, over time, plums start dropping out of used car lots until only lemons remain. Eventually, buyers will learn that used car lots usually only have lemons, so they'll learn to pay at most $4000. At this point, the market is "locked" into a lemon-only market - in other words, the market tends towards the lowest-common-denominator.

This analogy applies to the security industry - actual high-security products drop out of the market, until only insecure products (with pretty, glossy claims about security) remain.

One key assumption to this picture is that the seller knows the security strengths/weaknesses of all products (which is definitely questionable). So, a solution is to get independent analysis/auditors to look at products.

9 State of the Art

Currently, a large number of states has gone back to paper ballots. 2/3 of the US votes on paper ballots, and 1/3 of the US still votes on DRE's. This is depressing, since these systems all still have the published vulnerabilities, since it's very slow/expensive to re-certify new updates/patches to old systems.

However, to audit an election we can use the random sampling technique to quickly and efficiently perform election audits.