Electing a University President using Open-Audit Voting

Ben Adida*, Olivier de Marneffe, Olivier Pereira, Jean-Jacques Quisquater

* Harvard University
Université catholique de Louvain

August 11, 2009
The UCL president election

May 2008

Université catholique de Louvain (Belgium) sets new rules for the election of its president

- ≈ 25,000 potential voters
  - ≈ 30 members of the academic senate were voting before
- Voting operations conducted through browser/email
  - Large number of voters
  - Geographic dispersion of the voters
  - High familiarity level of the voters with the Internet
  - Low-coercion environment
Talk Outline

- UCL election specifics
- Helios 1.0
- Challenges and Deployment
- Lessons and statistics
The UCL president election (cnt.)

Election specifics

- 1-out-of-\(n\) election
- Absolute majority is needed to win, two rounds maximum
- Vote is not mandatory
- Sophisticated vote weighting rules: \(\text{(simplified a lot)}\)
  - 4 categories of voters: Faculty, Researchers, Administrative Staff and Students
    - F have 61% of the electoral votes
    - R, A, S receive 13% each
    - restrictions apply on sufficient participation rates
  \(\Rightarrow\) the weight of each vote depends on the global turnout
The UCL president election (cnt.)

Election outputs (as in the bylaws)

- number of electoral votes received by each candidate
- number of voters in each category
- (results by category are secret)
How to make this work?

Observations

- A university is a nice place to try something new
- Voters aren’t necessarily computer scientists
- Voters have UCL email address, login/password, member card
- Open-source and free starting point system needed (trust, versatility, time frame)
Helios 1.0 [Adida 2008]

Helios Voting
Elections you can audit

If my vote is supposed to stay secret, how can I verify that it was counted correctly?

The Helios Voting System implements advanced cryptographic techniques to maintain ballot secrecy while providing a mathematical proof that the election tally was correctly computed.

We call this an open-audit election, because you or anyone else can audit it.

Check out our Frequently Asked Questions.

Create an Open-Audit Election

[Home] [Login/Register] [Learn] [Blog/Updates]

All content on this site is licensed under a Creative Commons License. If you redistribute this content, you should give credit to Ben Adida and Harvard University.

www.heliosvoting.org
Helios 1.0 [Adida 2008]

Principles

- Browser-only voting system
- Low-coercion elections
- Design kept as simple as possible:
  - Booth can be used as many times as desired
    - ElGamal encryption of 0/1 for each choice
    - Benaloh challenge
      - cast or audit, authenticate on cast
  - Sako-Kilian mixnet before decryption
  - Web bulletin-board shows votes and proofs for everything
- Deployed on Google App Engine
Technical Challenges (1/3)

Key management

- Vote confidentiality relies on control of ElGamal private key

  Move to distributed ElGamal

- Trustees are not computer scientists

Distribute trust among experts
Use LiveCD, disk- and network-free laptops
Monitoring/Audit by independent company
Technical Challenges (2/3)

Vote weighting

- Participation per category and weights are public
  But support of candidates per category is secret
  ⇒ We cannot open individual votes!

  *Move to homomorphic tally instead of mixnets*

- Not enough to hide support of candidates per category...

\[
w_F n_F + w_R n_R + w_A n_A + w_S n_s = n
\]

... has \( \approx 1 \) solution for UCL election parameters
(knapsack-style problem)

*Use smaller, approximate weights*

*Careful choice provided \( \approx 10^5 \) sol. for \( \approx 10^{-4} \) precision*
Technical Challenges (3/3)

Audit complaints arbitration

- Voters invited to complain if WBB looks wrong
- DoS through complaints?

  *Give voters a way to prove things are wrong*
  *Timestamp/sign everything as evidence*

- Voters usually not familiar with signature

  *Signed pdf files seem most usable*
  *Signature through PortableSigner*
  *UCL Root certificate deployed on all UCL machines*
Deployment Challenges (1/3)

Privacy matters

- Publication of privacy policies

  Help of law office

- Name of voters cannot appear on bulletin board

  Each voter receives an alias

- Google App Engine constraining: data sent out of EU

  Move to Django/PostgreSQL for free software stack

![Django](django.png)

![PostgreSQL](postgresql.png)
Deployment Challenges (2/3)

Usability

- Make voting process as straightforward as possible
  - Keep information available for curious voter
    - 2-level interface: basic vs. curious voter

Robustness and availability

- Each election round lasts 35 hours

  Use redundant in-house servers
  Use cloud computing (Amazon EC2)
Deployment Challenges (3/3)

Communication

▶ Meetings/presentations
  ▶ Election bylaws working group, Rector council, Academic council, Employees Union, . . .
▶ Voter education
  ▶ University newspaper, lunch-time demos, screencasts, . . .
  ▶ Test election (student projects, for university sponsoring)
▶ Support organization
  ▶ Phone/email support by UCL IT Department
  ▶ Voting offices, with election officers
Election Phases – Organization

Registration Phase
- Voters registration 2 weeks
  - registration website
  - generation of voters’ aliases
  - generation of credentials
- Test Election same 2 weeks

Voting Phases (Each two rounds)
- Voting period 2 days, from 8am to 7pm the next day
  - same interface as Test Election
  - credentials still accessible on registration website
- WBB Audit day 1 day, next to the voting period
  - voters check the web bulletin board (… and may complain)
Participation

- 5142 registered voters
  - Very useful for credential negotiation
  - Very useful for 1st bound on number of voters
- 10644 votes tallied
  - \( \approx 3000 \) votes for test election
  - \( \approx 4000 \) votes for each round
- max. 17 votes/minute, emails trigger vote
Voter behavior

- 1% vote more than once (last vote counts)
  
  *Quite controversial, no strong impact*

- 3% use voting offices
  
  *Mostly people unfamiliar with PC*
  
  *Quite over-dimensioned on our side*

- 30% check their vote on web bulletin board
  
  *Quite high!*
  
  *Decreases on 2nd round*

- 120 tickets raised by UCL support
  
  1. Credentials lost
  2. JVM missing, use of Win95, IE4, . . .
  3. Did I do everything correctly?

*Importance of testing with broad spectrum of people...*
Web Bulletin Board Audit days

- 7 complaints issued during 2 rounds
  1. I am just trying to vote after the deadline
  2. I want to test the procedure
  3. I switched my receipt with someone else in the printer

Convenience of voting server with public data only

Tally

- 1st round leader was < 2 electoral votes from majority

  no objection, clear majority on 2nd round
Conclusion

▶ 1st significant-outcome, multi-thousand-voters open-audit election successful

▶ Open-audit elections allow moving
  ▶ from election manipulation opportunity
  ▶ to voter verification opportunity

▶ Each election is a significant project on its own

Thanks to all the people at who supported it!

UCL, Harvard, ENS Cachan, BlueKrypt, Google, Nexxit, ...
Thank you!

https://election.uclouvain.be/test