# Online verifiable elections with Helios

Stéphane Glondu

LORIA (Inria, CNRS, Université de Lorraine)

Libre Software Meeting, Geneva
July 10, 2012

# Outline

# Electronic voting

Elections are a security-sensitive process which is the cornerstone of modern democracy.

Electronic voting promises

- convenient, efficient and secure facility for recording and tallying votes
- for a variety of types of elections:
  from small committees or on-line communities...
  ...to public office (political) elections

Already used e.g. in Switzerland, France, USA...

# Two main families of e-voting

## Voting machines

- voters have to attend a polling station
- external authentication system (e.g. ID card)

## Internet voting

- voters vote from home
- using their own computer

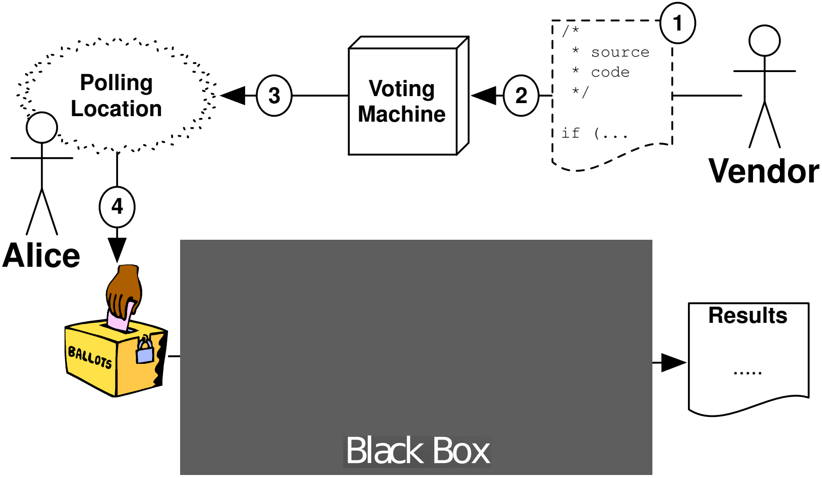
# A trust issue

In many systems in use today. . .

- the whole procedure is secret
  - secret specification
  - closed source software and/or proprietary hardware
  - audit restricted to (some) (supposedly honest) experts
  - . . .

  i.e. blind trust
- open source software/hardware is not enough!
  - the result should be verifiable independently
  - software should not matter
- people claim it's needed for security
  (security through obscurity)

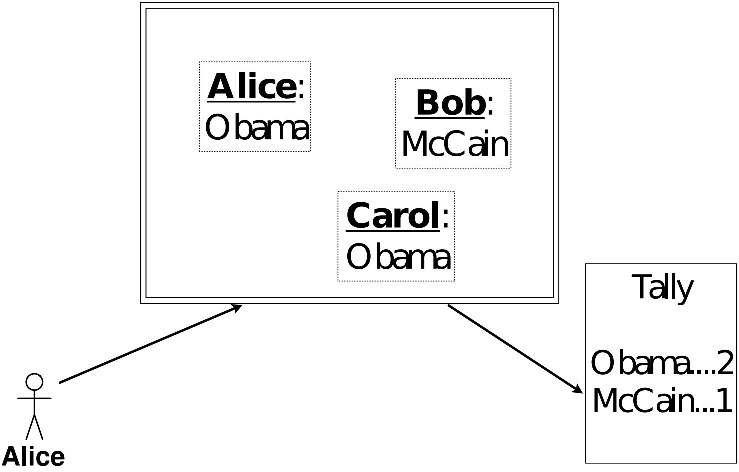
# A trust issue



# Properties

- Fairness: the result corresponds to the votes
- Eligibility: only legitimate voters can vote, and only once
- Individual verifiability: a voter can verify that her vote was really counted
- Universal verifiability: everyone can verify that the published outcome really is the sum of all votes

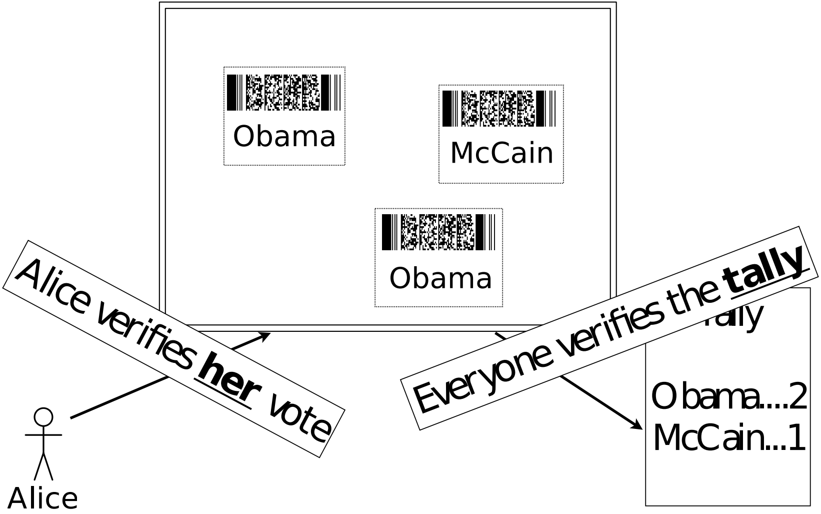
# Public ballots



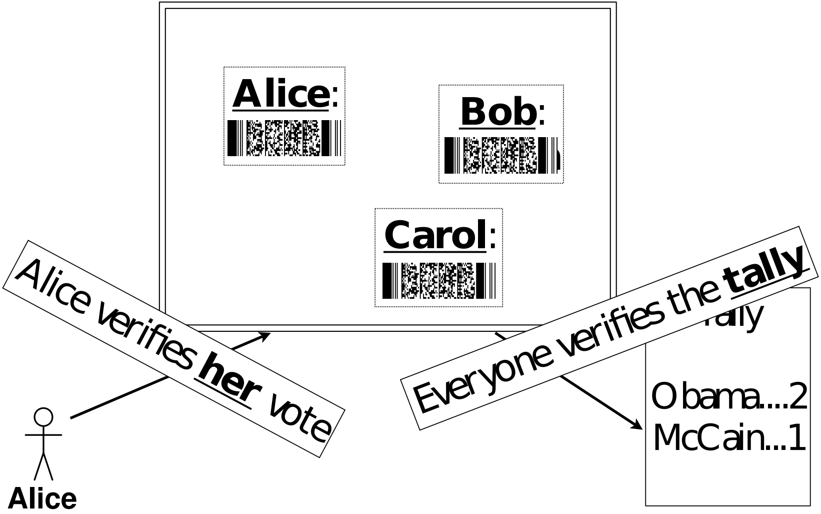
# Properties

- Fairness: the result corresponds to the votes
- Eligibility: only legitimate voters can vote, and only once
- Individual verifiability: a voter can verify that her vote was really counted
- Universal verifiability: everyone can verify that the published outcome really is the sum of all votes
- Privacy: the fact that someone voted in a particular way is not revealed to anyone else

# Anonymized public ballots



# Encrypted public ballots

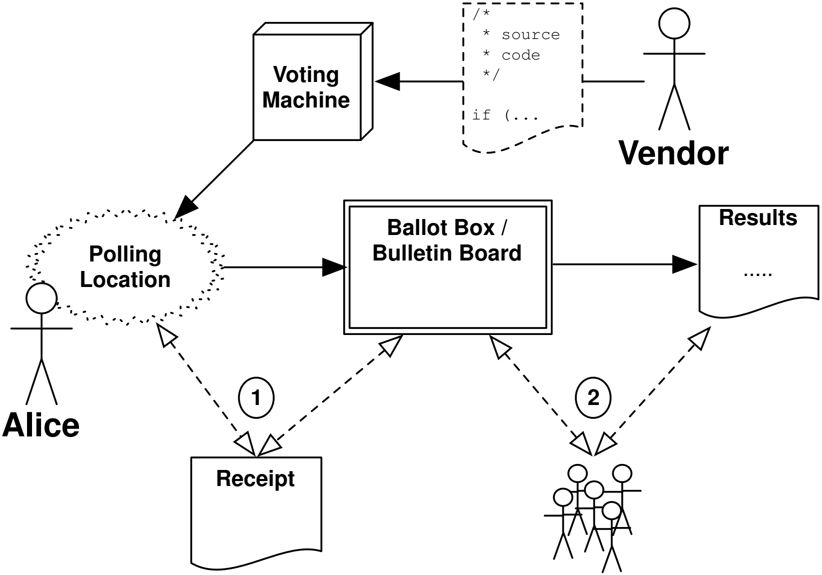


# Democratizing audits

- each voter is responsible for checking her receipt
- anyone (individual or organization) can audit the tally and verify the list of cast ballots

Verifiable elections

# End-to-end verification

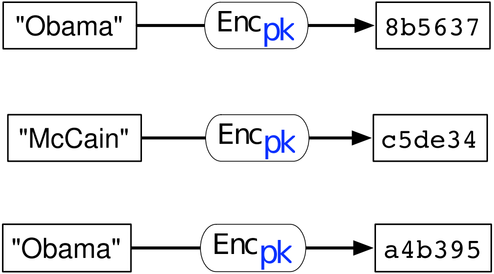


# Public key encryption

Public key: $\mathsf{pk}(A)$
Encryption: $\{m\}_{\mathsf{pk}(A)}$


Encryption with the public key and decryption with the private key.

# Randomized encryption



# Homomorphic encryption

- allows computations on encrypted messages without decrypting them

$$\{m_1\}_{\mathsf{pk}} \times \{m_2\}_{\mathsf{pk}} = \{m_1 + m_2\}_{\mathsf{pk}}$$

- for example: use the property

$$g^{m_1} \times g^{m_2} = g^{m_1 + m_2}$$

# A concrete voting system

## Phase 1: voting

| **Bulletin Board** | |
|---|---|
| Alice | $\{v_A\}_{\mathrm{pk}(S)}$ |
| Bob | $\{v_B\}_{\mathrm{pk}(S)}$ |
| ... | ... |

$v_A = 0$ or $1$

$v_B = 0$ or $1$

## Phase 2: tallying using homomorphic encryption

$$\prod_{i=1}^{n} \{v_i\}_{\mathrm{pk}(S)} = \{\sum_{i=1}^{n} v_i\}_{\mathrm{pk}(S)}$$

## Phase 3: decrypt the final result

*Only the final result needs to be decrypted!*

$\mathrm{pk}(S)$: public key of the election

# Cheating voters

- a malicious voter can cheat:

  **Result**:

  $\{v_A + v_B + v_C + v_D + \cdots\}_{\mathrm{pk}(S)}$ **Result**:

  $\{v_A + v_B + v_C + 100 + \cdots\}_{\mathrm{pk}(S)}$ **Result**:

  $\{v_A + v_B + v_C + v_D + \cdots\}_{\mathrm{pk}(S)}$
- hence, each voter must prove that her vote is 0 or 1 without revealing it
- it is possible with zero-knowledge proofs

# Cheating authorities

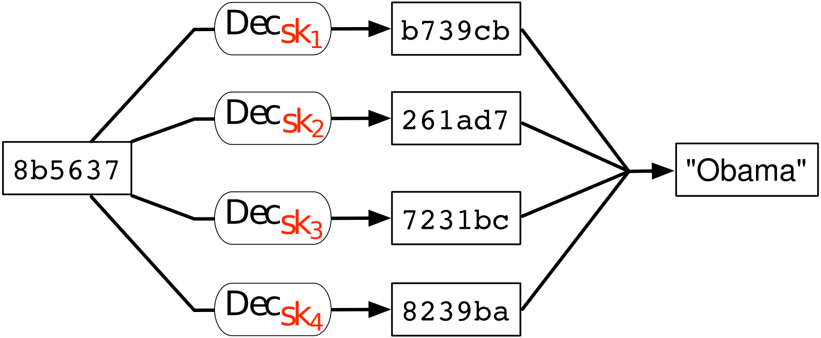
- malicious election authorities can cheat:

  **Result**: $\{v_A + v_B + v_C + v_D + \cdots\}_{\mathsf{pk}(S)}$

  | Bulletin Board | | |
  |---|---|---|
  | Alice | $\{v_A\}_{\mathsf{pk}(S)}$ | $v_A = 0 \text{ or } 1$ |
  | Bob | $\{v_B\}_{\mathsf{pk}(S)}$ | $v_B = 0 \text{ or } 1$ |
  | Chris | $\{v_C\}_{\mathsf{pk}(S)}$ | $v_C = 0 \text{ or } 1$ |
  | ... | ... | |

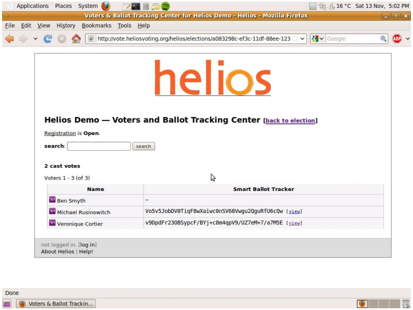
- can be mitigated by use of threshold decryption

# Threshold decryption



# Helios

http://vote.heliosvoting.org/




- developed by B. Adida *et al*
- used for:
  - university elections (Louvain, Princeton)
  - IACR board election

- libre version: `https://github.com/{benadida,glondu}/helios-server`
- better thought as an open specification for electronic voting
  - actively studied by the scientific communiyt

# Disclaimer

The security of Helios relies on the assumption that
the voter's computer can be trusted.

- Not suitable for political elections
  A corrupted machine may:
  - leak the choice of the voter
  - vote for a different candidate

  The same applies to systems currently deployed for political elections!
  - concrete attack by Laurent Grégoire on the system used by the French abroad
- Suitable for medium issue elections:
  - professional elections
  - scientific councils, students representatives, *etc.*
- To be compared with remote voting:
  - better guarantees than vote by mail

# Guaranteed properties

- Fairness: the result corresponds to the votes
- Eligibility (partial): voters vote only once
- Individual verifiability: a voter can verify that her vote was really counted
- Universal verifiability: everyone can verify that the published outcome really is the sum of all votes
- Privacy: the fact that someone voted in a particular way is not revealed to anyone else

# Mitigation for questionable properties

- LiveCD with minimal software and certificates
  - and documentation on how to build it by oneself
- voter-initiated audit before casting
  - using third-party software and/or hardware
  - possibly home-made
- honeypots

# Room for improvement

- resistance to ballot stuffing
- coercion resistance, ticket freeness
- everlasting privacy
- mixnets
- elliptic curve cryptography

# Conclusion

Electronic voting is possible without *blind* trust. . .

. . . but it is not ready to replace "traditional" voting

# Questions?

Contact:


- helios-voting@googlegroups.com
- steph@glondu.net


 Acknowledgements:

- Ben Adida, http://ben.adida.net/
- Véronique Cortier