Le vote électronique : un défi pour la vérification formelle

Steve Kremer

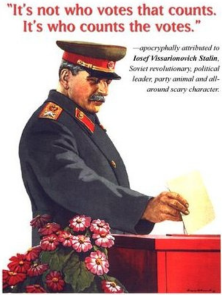
Loria, Inria Nancy

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 through to full-scale national elections

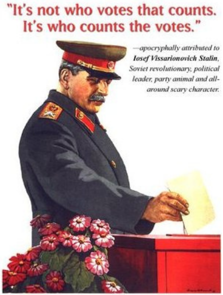


Electronic voting

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 through to full-scale national elections



E-voting may include :

- use of voting machines in polling stations
- remote voting, via Internet (i-voting)

Real-world Internet elections

Recent political legally binding Internet elections in Europe :

- parliamentary elections in Switzerland (several cantons)
- parliamentary election in Estonia (all eligible voters)
- municipal and county elections in Norway (selected municipalities, selected voter groups)
- parliamentary elections in in France ("expats")

But also banned in Germany, Ireland, UK

Even more professional elections

Attacks !

Attacks by Alex Halderman and his team :

- attack on pilot project for overseas and military voters : took control of vote server, changed votes, removed root kit present on server, . . .
- Indian voting machines : clip-on memory manipulator
- Re-programmed e-voting machine used in US elections to play pack-man

. . . and many more

Attacks !

Attacks by Alex Halderman and his team :

- attack on pilot project for overseas and military voters : took control of vote server, changed votes, removed root kit present on server, . . .
- Indian voting machines : clip-on memory manipulator
- Re-programmed e-voting machine used in US elections to play pack-man

. . . and many more

There exist also attacks on paper based remote voting, e.g. attack by Cortier *et al.* on a postal voting system used in CNRS elections

Vote privacy

Anonymity of the vote :
no one should learn how I voted

Vote privacy

Anonymity of the vote :
no one should learn how I voted

We may want even more :

Receipt-freeness/coercion-resistance :
I cannot prove to someone else how I voted

$\leadsto$ avoid vote-buying / coercion

Election transparency

In traditional elections :

- transparent ballot box
- observers
- ...

Election transparency

In traditional elections :

- transparent ballot box
- observers
- ...

In e-voting : End-to-end Verifiability

- Individual verifiability : vote cast as intended

  e.g., voter checks his encrypted vote is on a public bulletin board
- Universal verifiability : vote counted as casted

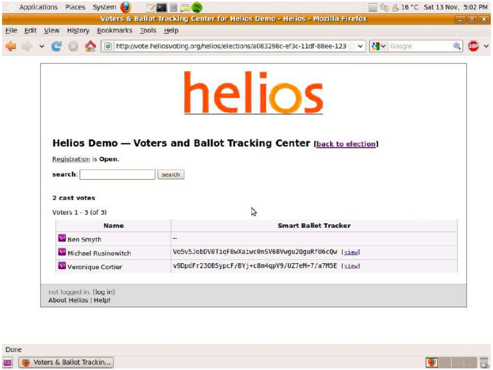
  e.g., crypto proof that decryption was performed correctly
- Eligibility verifiability : only eligible votes counted

  e.g., crypto proof that every vote corresponds to a credential

$\rightsquigarrow$ Verify the election, not the system !

Verifiable online elections via the Internet

http ://heliosvoting.org/



Already in use :

- Election at Louvain University Princeton
- Election of the IACR board (major association in Cryptography)

Phase 1 : voting

Bulletin Board	
Alice	$\{v_A\}_{pk(S)}$
Bob	$\{v_B\}_{pk(S)}$
Chris	$\{v_C\}_{pk(S)}$

$v_A = 0$ or 1
$v_B = 0$ or 1
$v_C = 0$ or 1

$pk(S)$: public key, the private key being shared among trustees.

Behavior of Helios (simplified)

Phase 1 : voting

$\xrightarrow{\{v_D\}_{pk(S)}}$

Bulletin Board

Alice	$\{v_A\}_{pk(S)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pk(S)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{pk(S)}$	$v_C = 0$ or 1

$pk(S)$: public key, the private key being shared among trustees.

Behavior of Helios (simplified)

Phase 1 : voting

Bulletin Board

Alice	$\{v_A\}_{pk(S)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pk(S)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{pk(S)}$	$v_C = 0$ or 1
David	$\{v_D\}_{pk(S)}$	$v_D = 0$ or 1

$pk(S)$: public key, the private key being shared among trustees.

Behavior of Helios (simplified)

Phase 1 : voting

Bulletin Board		
Alice	$\{v_A\}_{pk(S)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pk(S)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{pk(S)}$	$v_C = 0$ or 1
David	$\{v_D\}_{pk(S)}$	$v_D = 0$ or 1
...	...	

Phase 2 : Tallying using homomorphic encryption (El Gamal)

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

based on $g^a * g^b = g^{a+b}$

$\rightarrow$ Only the final result needs to be decrypted !

$pk(S)$: public key, the private key being shared among trustees.

This is oversimplified !

Bulletin Board	
Alice	$\{v_A\}_{pk(S)}$
Bob	$\{v_B\}_{pk(S)}$
Chris	$\{v_C\}_{pk(S)}$
David	$\{v_D\}_{pk(S)}$
...	...

$v_A = 0$ or 1

$v_B = 0$ or 1

$v_C = 0$ or 1

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

This is oversimplified !

Bulletin Board		
Alice	$\{v_A\}_{pk(S)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pk(S)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{pk(S)}$	$v_C = 0$ or 1
David	$\{v_D\}_{pk(S)}$	$v_D = 100$
...	...	

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

A malicious voter can cheat !

This is oversimplified !

Bulletin Board

Alice	$\{v_A\}_{pk(S)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pk(S)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{pk(S)}$	$v_C = 0$ or 1
David	$\{v_D\}_{pk(S)}$	~~$v_D = 100$~~
...	...	

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

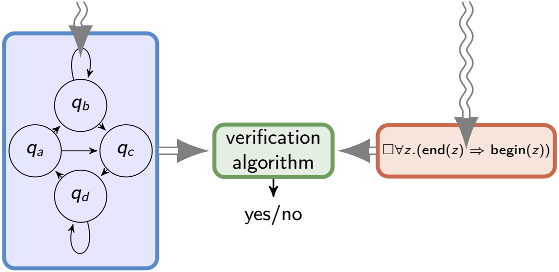
~~A malicious voter can cheat !~~

In Helios : use Zero Knowledge Proof

$$\{v_D\}_{pk(S)}, \text{ZKP}\{v_D = 0 \text{ or } 1\}$$

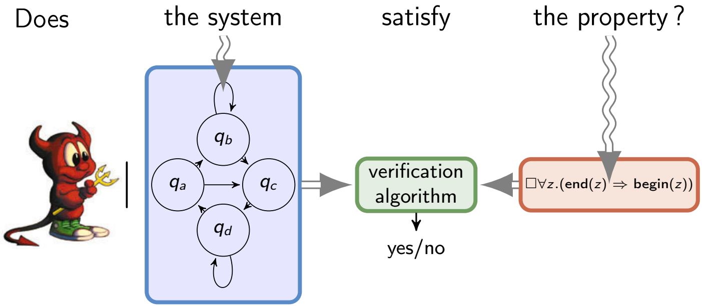
Formal verification of critical systems

Does the system satisfy the property ?



Formal verification of critical systems

Applied to security protocols :



Difficulties :

⇝ arbitrary attacker controlling the network

⇝ infinite state system

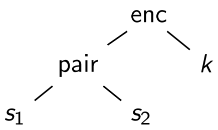
Techniques :

automated deduction, concurrency theory, model-checking, . . .

Symbolic analysis

Symbolic techniques (following [Dolev&Yao'82]) :

- messages = terms



- perfect cryptography (equational theories)

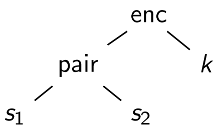
$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x \quad \mathsf{fst}(\mathsf{pair}(x, y)) = x \quad \mathsf{snd}(\mathsf{pair}(x, y)) = y$$

- the network is the attacker

Symbolic analysis

Symbolic techniques (following [Dolev&Yao'82]) :

- messages = terms



- perfect cryptography (equational theories)

$$\text{dec}(\text{enc}(x, y), y) = x \quad \text{fst}(\text{pair}(x, y)) = x \quad \text{snd}(\text{pair}(x, y)) = y$$

- the network is the attacker

Automated tools successfully found flaws in :

- Google's Single Sign-On protocol
- ISO/IEC 9798 standard for entity authentication
- commercial PKCS#11 key-management tokens
- ...

Modelling properties and properties

Protocols modelled in a process calculus with terms, e.g. the applied pi calculus

$$
\begin{array}{rcll}
P ::= & & 0 & \\
& | & \mathrm{in}(c, x).P & \text{input} \\
& | & \mathrm{out}(c, t).P & \text{output} \\
& | & \text{if } t_1 = t_2 \text{ then } P \text{ else } Q & \text{conditional} \\
& | & P \parallel Q & \text{parallel} \\
& | & !P & \text{replication} \\
& | & \text{new } n.P & \text{restriction}
\end{array}
$$

Modelling properties and properties

Protocols modelled in a process calculus with terms, e.g. the applied pi calculus

$$
\begin{array}{rcll}
P ::= & & 0 & \\
& \mid & \mathsf{in}(c, x).P & \text{input} \\
& \mid & \mathsf{out}(c, t).P & \text{output} \\
& \mid & \text{if } t_1 = t_2 \text{ then } P \text{ else } Q & \text{conditional} \\
& \mid & P \parallel Q & \text{parallel} \\
& \mid & !P & \text{replication} \\
& \mid & \text{new } n.P & \text{restriction}
\end{array}
$$

Properties

A process P satisfies φ if for any process A

$$A \parallel P \models \varphi$$

How to model vote privacy ?

How can we model “the attacker does not learn my vote (0 or 1)” ?

How to model vote privacy?

How can we model "the attacker does not learn my vote (0 or 1)"?

- The attacker cannot learn the value of my vote

How to model vote privacy ?

How can we model “the attacker does not learn my vote (0 or 1)” ?

- The attacker cannot learn the value of my vote
  $\rightsquigarrow$ but the attacker knows values 0 and 1

How to model vote privacy ?

How can we model "the attacker does not learn my vote (0 or 1)" ?

- ~~The attacker cannot learn the value of my vote~~

- The attacker cannot distinguish when we change the voter identity : $V_A(v) \approx V_B(v)$

How to model vote privacy ?

How can we model "the attacker does not learn my vote (0 or 1)" ?

- ~~The attacker cannot learn the value of my vote~~

- The attacker cannot distinguish when we change the voter identity : $V_A(v) \approx V_B(v)$
  $\rightsquigarrow$ but identities are revealed

How to model vote privacy ?

How can we model "the attacker does not learn my vote (0 or 1)" ?

- ~~The attacker cannot learn the value of my vote~~
- ~~The attacker cannot distinguish when we change the voter identity : $V_A(v) \approx V_B(v)$~~
- The attacker cannot distinguish when change the vote : $V_A(0) \approx V_A(1)$

How to model vote privacy ?

How can we model "the attacker does not learn my vote (0 or 1)" ?

- ~~The attacker cannot learn the value of my vote~~
- ~~The attacker cannot distinguish when we change the voter identity : $V_A(v) \approx V_B(v)$~~
- The attacker cannot distinguish when change the vote : $V_A(0) \approx V_A(1)$
  $\rightsquigarrow$ but election outcome is revealed

How to model vote privacy ?

How can we model "the attacker does not learn my vote (0 or 1)" ?

- ~~The attacker cannot learn the value of my vote~~
- ~~The attacker cannot distinguish when we change the voter identity : $V_A(v) \approx V_B(v)$~~
- ~~The attacker cannot distinguish when change the vote : $V_A(0) \approx V_A(1)$~~
- The attacker cannot distinguish the situation where two honest voters swap votes :

$$V_A(0) \parallel V_B(1) \approx V_A(1) \parallel V_B(0)$$

Also avoids the problematic case of unanimity !

[Kremer, Ryan '05]

Looking again at Helios

Bulletin Board		
Alice	$\{v_A\}_{pk(S)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pk(S)}$	$v_B = 0$ or 1

Looking again at Helios

Bulletin Board		
Alice	$\{v_A\}_{pk(S)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pk(S)}$	$v_B = 0$ or 1
Chris	$\{v_A\}_{pk(S)}$	

Vote-copying attack :
copying Alice's vote introduces a bias in the outcome

Weakness in Helios discovered when trying to prove the previous definition of anonymity

[Cortier, Smyth '11]

Challenges for automated verification

Security proofs for e-voting protocols out of scope of existing tools.

Challenges for automated verification

Security proofs for e-voting protocols out of scope of existing tools.

- New properties : observational equivalence
  Today : mature theory and verification tools for authentication and confidentiality

  $\rightsquigarrow$ both theory and verification tools for equivalence properties are still work in progress

Challenges for automated verification

Security proofs for e-voting protocols out of scope of existing tools.

- New properties : observational equivalence
  Today : mature theory and verification tools for authentication and confidentiality

  $\rightsquigarrow$ both theory and verification tools for equivalence properties are still work in progress

- New crypto primitives : complex equational theories, e.g. homomorphic encryption

  $$\mathsf{enc}(x_1, r_1, y) * \mathsf{enc}(x_2, r_2, y) = \mathsf{enc}(x_1 + x_2, r_1 \times r_2, y)$$

  where $*, \times, +$ are associative and commutative

  $\rightsquigarrow$ not (yet) supported by protocol verification tools

Challenges for automated verification

Security proofs for e-voting protocols out of scope of existing tools.

- New properties : observational equivalence
  Today : mature theory and verification tools for authentication and confidentiality

  $\rightsquigarrow$ both theory and verification tools for equivalence properties are still work in progress

- New crypto primitives : complex equational theories, e.g. homomorphic encryption

  $$\mathsf{enc}(x_1, r_1, y) * \mathsf{enc}(x_2, r_2, y) = \mathsf{enc}(x_1 + x_2, r_1 \times r_2, y)$$

  where $*, \times, +$ are associative and commutative

  $\rightsquigarrow$ not (yet) supported by protocol verification tools

Warning : verified protocol $\neq$ secure system !

Conclusion

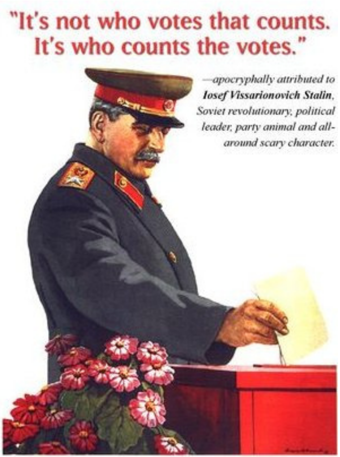
Some good systems exist

- Helios : anonymity and verifiability, but no coercion-resistance
  Belenios : variant of Helios developed at LORIA
- Civitas : verifiability and coercion-resistance
- End-to-end verifiable election systems in polling stations : Scantegrity, Prêt-à-Voter, . . .

Limitations

- Authentication in remote elections is based on credentials that are transferrable
- Untrustworthy voting clients (malware)
  - votes may be leaked
  - software changing votes

⇝ some mitigations exist, active research topic !


"It's not who votes that counts.
It's who counts the votes."
—apocryphally attributed to
Iosef Vissarionovich Stalin,
Soviet revolutionary, political
leader, party animal and all-
around scary character.

Thank you