A Modular Multi-Modal Specification of Real-Timed, End-To-End Voter-Verifiable Voting Systems

(E-Voting Seminar, Bern University of Applied Sciences)

Simon Kramer

University of Luxembourg
Institute of Mathematical Sciences, Chennai

March 29, 2011

UNIVERSITÉ DU LUXEMBOURG — aides à la formation recherche — Fonds National de la Recherche Luxembourg — MARIE CURIE ACTIONS — The Institute of Mathematical Sciences

Motivation

- Voting is the foundation of **democracy:** corrupt voting $\rightsquigarrow$ corrupt government.
- **Electronic voting** introduces new possibilities:
  1. *automation* of the voting process with networked computers (remotely accessible ballot-collecting, automated ballot-counting points);
  2. *ontological and epistemological guarantees* on the voting process thanks to modern cryptography.
- But: new possibilities $\rightsquigarrow$ new vulnerabilities.
- **Voting systems are societal-safety-critical systems!**
- Best practices are an ethical imperative: **formal methods.**

Goal

To obtain **a specification of real-timed electronic voting systems** that is:

- *intuitive,*
- *implementation-independent,*
- *consistent,*
- what we believe to be *up-to-date complete,*
- a *well-compounded single logical formula.*

Problem

1. the **conceptual complexity** of electronic voting
2. the difficulty of isolating a *pragmatically* sufficiently expressive (built-in *idioms*) **specification language** (set theory is no front-end option)

Solution

1. opt for **logical specification**
2. adopt a **principled methodology:**

   3 strategic (general) + 2 tactical (specific) principles

Methodology—*strategic principles*

1. **minimality—no semantic and syntactic overkill:**
   1.1 minimally sufficient semantic expressiveness of the specification language (Ockham's razor),
   1.2 minimally new specification code through *code reuse* (voter verifiability as trust-inducing accountability [KGO11]);
2. **modularity—separation of conceptual concerns:** top-down development of the specification applying a D&C strategy by splitting it up into semantically separate (security) sub-requirements;
3. **multi-modality—logico-linguistic fidelity—informal language transcribes into formal logic:** 1 logical operator for each key-modelling idiom, here modal idioms for modelling time, knowledge, and agent provability.

Methodology—*tactical principles*

1. **agent correctness:** the behavioural correctness of the voting-system-constituting agents
2. **data adequacy:** the soundness and (relative) completeness of the voting data processed by the system

Contribution

A formal specification of electronic voting systems that are accountable (and thus trustworthy) to their users that meets the following desiderata:

1. all our *goal criteria;*
2. being a *formal transcription* of a suitable natural-language formulation;
3. *automatic translatability* into standard first-order language, the most wide-spread *lingua franca* of Science;
4. *intra- and inter-comparability* w.r.t. sub-requirements and other specifications, respectively;
5. *implementability-proof by inspection* (counter-balancing results about the inconsistency of certain property pairs);
6. *implementation-verification parallelisability*.

Specification language

- specific **linguistic primitives** proper to voting systems;
- general **logical operators** including temporal, epistemic, and provability modalities.

Linguistic primitives

The primitives of our specification language are

- **logical constants** for the **individuals** in—and
- **relational symbols** for the **elementary facts** about—

voting systems.

The logical constants and relational symbols together form the *atomic propositions*.

Fixing the atomic propositions of a logic means instantiating the logic as a theory of a specific subject matter (here voting systems).

Logical constants

- **agent identifiers** $a, b, c, \mathtt{Tallier} \in \mathcal{A}$ where $|\mathcal{A}| \in \mathbb{N}$ and $\mathtt{Tallier}$ designates the tallier
- filled-in **ballots** $B \in \mathcal{B}$ where $|\mathcal{B}| \in \mathbb{N}$
- possible **vote results** $r \in \mathcal{R}$ where $|\mathcal{R}| \in \mathbb{N}$
- **real-time points** $t \in \mathcal{Q}$ where $|\mathcal{Q}| = |\mathbb{Q}|$

Relational symbols—unary symbols

- BBbalance, for expressing as the atomic proposition BBbalance(r) the elementary fact that the voting result r indeed corresponds to the balance of the tallier's, say, ballot book; BBbalance is a system-specific primitive;
- PA, for expressing as the atomic proposition PA(r) the elementary fact that the voting result r is being publicly announced;
- correct, for expressing as the atomic proposition correct(B) the elementary fact that B is a correctly filled-in ballot, which is type-checkable; correct is a system-specific primitive.

Relational symbols—binary symbols

- $=$, for expressing as the atomic proposition $a = b$ and $B = B'$ the elementary fact that the two agent identifiers a and b on the one hand and the two ballots B and B' on the other hand actually refer to one and the same agent and ballot, resp.;
- registrar, for expressing as the atomic proposition b registrar a the elementary fact that the agent b is a registrar of the agent a; thus a is a *legitimate voter;*
- inBB, for expressing as the atomic proposition B inBB b the elementary fact that the ballot B is an entry in b's, say, ballot book;

Relational symbols—binary symbols (continued)

- reports, for expressing as the atomic proposition reports(b, B) the elementary fact that the agent b reports the filled-in ballot B to the tallier `Tallier`.
- $[\cdot, \cdot]$, for expressing as the atomic propositions
  - $[t, t_1]$, for vote casting and registering,
  - $[t, t_2]$, for vote registering,
  - $[t, t_3]$, for vote reporting to the tallier,
  - $[t', t'']$, for public vote announcement,
  - $[t, t'']$, for the complete voting process,

  the elementary facts that the current time is within the respective time points

$$t < t_1 < t_2 < t_3 < t' < t'' \in \mathcal{Q}\,.$$

Relational symbols—ternary symbols

casts, for expressing as the atomic proposition casts(a, B, b) the elementary fact that the agent a casts the filled-in ballot B at the location of agent b.

Logical operators

- *propositional logic,* namely: $\neg$ (negation), $\land$ (conjunction), $\lor$ (inclusive disjunction), $\rightarrow$ (material conditional), $\leftrightarrow$ (material bi-conditional), and $\oplus$ (exclusive disjunction)
- *linear temporal logic with past [MP91],* namely:
  - $\overline{\Diamond}_{\leq 1}$, "at most once in the past"
  - $\overline{\Diamond}!$, "exactly once in the past"
  - $\overline{\Diamond}$, "once in the past"
  - $\overline{\bigcirc}$, "previous logical time"
  - $\overline{\Box}$, "so far"
  - 1, "now for the first time" $(1(\phi) := \phi \land \overline{\bigcirc\Box}(\neg\phi))$,
  - $\Box$, "henceforth"
  - $\bigcirc$, "next logical time,"
  - $\Diamond$, "eventually"

Logical operators (continued)

- *standard epistemic logic [FHMV95],* namely K_a "agent a knows that," with the following characteristic laws, ϕ and ϕ' denoting logical formulas:
  - $\mathsf{K}_a(\phi \rightarrow \phi') \rightarrow (\mathsf{K}_a(\phi) \rightarrow \mathsf{K}_a(\phi'))$ (Kripke's law)
  - $\mathsf{K}_a(\phi) \rightarrow \phi$ (truth law)
  - $\mathsf{K}_a(\phi) \rightarrow \mathsf{K}_a(\mathsf{K}_a(\phi))$ (positive introspection)
  - $\neg\mathsf{K}_a(\phi) \rightarrow \mathsf{K}_a(\neg\mathsf{K}_a(\phi))$ (negative introspection)
  - $\dfrac{\phi}{\mathsf{K}_a(\phi)}$ (necessitation);

Logical operators (continued)

- *a multi-agent provability logic [Kra08, Kra12],* namely P_a "agent a can prove to all other agents including herself that", with the following characteristic laws:
  - $\mathsf{P}_a(\phi \rightarrow \phi') \rightarrow (\mathsf{P}_a(\phi) \rightarrow \mathsf{P}_a(\phi'))$ (Kripke's law)
  - $\mathsf{P}_a(\phi) \rightarrow \phi$ (truth law)
  - $\mathsf{P}_a(\phi) \rightarrow \mathsf{P}_a(\mathsf{P}_a(\phi))$ (positive introspection)
  - $\dfrac{\phi}{\mathsf{P}_a(\phi)}$ (necessitation)
  - $\mathsf{P}_a(\phi) \rightarrow \mathsf{K}_a(\phi)$ (relation to knowledge);
- similar laws for more general provability operators $\mathsf{P}_{(a,b)}$

The specification

$$\begin{aligned}\text{Specification} := {} & \text{RolePlot} \land \\ & \text{Accountability} \land \\ & \text{Uncoercibility}\end{aligned}$$

where

$$\text{Uncoercibility} := \text{ReceiptFreeness} \land \text{Privacy}$$

Role plot

RolePlot :=
$\exists a \exists b \Box([t, t''] \rightarrow b \text{ registrar } a) \wedge$
$\forall a \forall b \Box(b \text{ registrar } a \rightarrow$
$\Box([t, t''] \rightarrow (b \text{ registrar } a \wedge$
$\neg(a \text{ registrar } b) \wedge$
$\neg(b = \texttt{Tallier}))))$

"During voting, registrar relationships are non-empty, persistent, asymmetric, and mutually exclusive w.r.t. the tallier property."

Roles

1. (legitimate) *voter*, i.e., agents $a \in \mathcal{A}$ such that

   $\text{voter}(a) := \exists b(b \text{ registrar } a)$;

2. *registrar*, i.e., agents $b \in \mathcal{A}$ such that

   $\text{registrar}(b) := \exists a(b \text{ registrar } a)$;

3. *tallier*, i.e., agents $c \in \mathcal{A}$ such that

   $\text{tallier}(c) := (c = \texttt{Tallier})$.

Accountability

Accountability := Abusefreeness $\wedge$ Auditability

Abusefreeness := $\forall a \Box(\text{correct}(a) \rightarrow \mathsf{P}_a(\text{correct}(a)))$

"For all agents a (there are finitely many of them), henceforth, if a is correct then a can prove (to all agents including herself) that she is correct."

Accountability (continued)

Auditability := $\forall a \Box(\neg\, \text{correct}(a) \rightarrow$
$\forall b \Diamond \Box \mathsf{P}_b(\neg\, \text{correct}(a)))$

"For all agents a, henceforth, if a is incorrect then all agents (including a) can eventually henceforth prove that a is incorrect."

Agent correctness

$$\mathsf{correct}(a) := a \ \mathsf{roleCompatible} \ \{ \textsc{Voter}, \textsc{Registrar}, \textsc{Tallier} \} \,,$$

$$\begin{aligned} a \ \mathsf{roleCompatible} \ \{\textsc{Voter}, \textsc{Registrar}, \textsc{Tallier}\} := & \ (\mathsf{caster}(a) \rightarrow \mathsf{voter}(a)) \land \\ & (\mathsf{voter}(a) \rightarrow \mathsf{correctVoter}(a)) \land \\ & (\mathsf{registrar}(a) \rightarrow \mathsf{correctRegistrar}(a)) \land \\ & (\mathsf{tallier}(a) \rightarrow \mathsf{correctTallier}(a)) \end{aligned}$$

where

$$\mathsf{caster}(a) := \exists B \exists b (\mathsf{casts}(a, B, b))$$

Voter correctness

$$\mathsf{correctVoter}(a) := \mathsf{noIncorrectCast}(a) \land \mathsf{AtMostOneCorrectCast}(a)$$

$$\mathsf{noIncorrectCast}(a) := \neg \exists B \exists b \overline{\Diamond}(\mathsf{incorrectlyCasts}(a, B, b)) \,,$$

where

$$\mathsf{incorrectlyCasts}(a, B, b) := \mathsf{casts}(a, B, b) \land \neg \, \mathsf{castCorrectness}(a, B, b) \,.$$

Voter correctness (continued)

$$\mathsf{AtMostOneCorrectCast}(a) := \exists_{\leq 1} B \exists_{\leq 1} b \overline{\Diamond_{\leq 1}}(\mathsf{correctlyCasts}(a, B, b)) \,,$$

where

$$\mathsf{correctlyCasts}(a, B, b) := \mathsf{casts}(a, B, b) \land \mathsf{castCorrectness}(a, B, b) \,.$$

$$\mathsf{castCorrectness}(a, B, b) := \mathsf{correct}(B) \land b \ \mathsf{registrar} \ a \land [t, t_1]$$

Registrar correctness

$$\mathsf{correctRegistrar}(b) := \mathsf{adequateBB}(b) \land \mathsf{adequateReporting}(b)$$

$$\mathsf{adequateBB}(b) := \mathsf{soundBB}(b) \land \mathsf{completeBB}(b)$$

$$\mathsf{soundBB}(b) := \forall B (B \ \mathsf{inBB} \ b \rightarrow \exists a \overline{\Diamond}(\mathsf{casts}(a, B, b)))$$

$$\mathsf{completeBB}(b) := \forall B (\exists a \overline{\Diamond}(\mathsf{casts}(a, B, b)) \rightarrow B \ \mathsf{inBB} \ b)$$

Registrar correctness (continued)

$$\mathsf{adequateReporting}(b) := \mathsf{soundReporting}(b) \land \mathsf{completeReporting}(b)$$

$$\mathsf{soundReporting}(b) := \forall B \overline{\Box}(\mathsf{reports}(b, B) \rightarrow ([t, t_3] \land B \text{ inBB } b \land \mathsf{correct}(B)))$$

$$\mathsf{completeReporting}(b) := \forall B((B \text{ inBB } b \land \mathsf{correct}(B)) \rightarrow \Diamond(\mathsf{reports}(b, B) \land [t, t_3]))$$

Tallier correctness

$$\mathsf{correctTallier}(c) := \mathsf{tallier}(c) \land \mathsf{adequateBB} \land \mathsf{noIncorrectPA} \land \mathsf{eventuallyExactlyOneCorrectPA}$$

$$\mathsf{adequateBB} := \mathsf{soundBB} \land \mathsf{completeBB}$$

$$\mathsf{soundBB} := \forall B(B \text{ inBB } \texttt{Tallier} \rightarrow \exists b \overline{\Diamond}(\mathsf{reports}(b, B)))$$

$$\mathsf{completeBB} := \forall B(\exists b \overline{\Diamond}(\mathsf{reports}(b, B)) \rightarrow B \text{ inBB } \texttt{Tallier})$$

Tallier correctness (continued)

$$\mathsf{noIncorrectPA} := \neg \exists r \overline{\Diamond}(\mathsf{incorrectPA}(r)) \,,$$

where

$$\mathsf{incorrectPA}(r) := \mathsf{PA}(r) \land \neg\, \mathsf{PAcorrectness}(r) \,.$$

$$\mathsf{eventuallyExactlyOneCorrectPA} := \mathsf{withinIntervalAtMostOneCorrectPA} \land \mathsf{rightAfterIntervalExactlyOneCorrectPA} \,,$$

where ...

Tallier correctness (end)

$$\mathsf{withinIntervalAtMostOneCorrectPA} := ([t', t''] \rightarrow \exists_{\leq 1} r \overline{\Diamond_{\leq 1}}(\mathsf{correctPA}(r)))$$

and

$$\mathsf{rightAfterIntervalExactlyOneCorrectPA} := ((\neg[t', t''] \land \overline{\bigcirc}[t', t'']) \rightarrow \exists! r \overline{\bigcirc \Diamond !}(\mathsf{correctPA}(r))) \,.$$

$$\mathsf{correctPA}(r) := \mathsf{PA}(r) \land \mathsf{PAcorrectness}(r)$$

$$\mathsf{PAcorrectness}(r) := \mathsf{BBbalance}(r) \land [t', t'']$$

Receipt-freeness

$$\text{ReceiptFreeness} := \text{Unanimity} \oplus \text{ExclusiveVoteProvability}$$

In an unanimous vote, all ballots that have been cast right after the casting-registering interval are identical.

$$\begin{array}{l}\text{Unanimity} := \\ \quad \Box((\neg[t, t_1] \wedge \overline{\bigcirc}[t, t_1]) \rightarrow \\ \qquad \forall B \forall B'(\begin{pmatrix} \exists a \exists b \overline{\Diamond}(\text{casts}(a, B, b)) \wedge \\ \exists a \exists b \overline{\Diamond}(\text{casts}(b, B', b)) \end{pmatrix} \rightarrow \\ \qquad\quad B' = B))\end{array}$$

Uncoercibility (continued)

$$\begin{array}{l}\text{ExclusiveVoteProvability} := \\ \quad \forall a \forall B \forall b \Box(\text{casts}(a, B, b) \rightarrow \\ \qquad \forall c \Box(\mathsf{P}_{(a,c)}(\exists b(\overline{\Diamond}\text{casts}(a, B, b))) \rightarrow \\ \qquad\quad c = a))\end{array}$$

"For all agents a, filled-in ballots B, and agents b, henceforth, if a casts B in the ballot box of b then for all agents c, henceforth, if a can prove to c that there is an agent b in whose ballot box a cast B then it is (only) a (herself)."

Privacy

$$\text{Privacy} := \text{Unanimity} \oplus \text{AnonymityAndSecrecy}$$

Anonymity and secrecy is defined as the exclusive knowledge of one's own vote w.r.t. both:

- the *act* $\exists b(\overline{\Diamond}\text{casts}(a, B, b))$—anonymity
- the *content* B (The ballot B occurs free in the formula $\exists b(\overline{\Diamond}\text{casts}(a, B, b))$!)—secrecy

of the vote.

Privacy (continued)

$$\begin{array}{l}\text{AnonymityAndSecrecy} := \\ \quad \forall a \forall B \forall b \Box(\text{casts}(a, B, b) \rightarrow \\ \qquad \forall c \Box(\mathsf{K}_c(\exists b(\overline{\Diamond}\text{casts}(a, B, b))) \rightarrow \\ \qquad\quad c = a))\end{array}$$

"For all agents a, filled-in ballots B, and agents b, henceforth, if a casts B in the ballot box of b then for all agents c, henceforth, if a knows that there is an agent b in whose ballot box a cast B then it is (only) a (herself)."

Specification properties

1. **Satisfiability:** by recursive inspection of the specification(!)
2. **Corollary:** non-contradiction of verifiability (provability) with
   2.1 privacy
   2.2 receipt-freeness;
3. Relation to **trust:**
   - accountability induces trust in the sense of [KGO11]:

     $$a \text{ sTrusts } b := \mathrm{K}_a(\text{correct}(b))\,;$$

   - accountability is provability of correctness, which implies knowledge of correctness;
   - hence, accountable voting systems are trustworthy.
4. Relation to other, voting-specific properties: *democracy, fairness, integrity, verifiable participation*.

Assessment

- a modular multi-modal specification of real-timed, universally end-to-end voter-verifiable voting systems, i.e., a formal but intuitive specification of real-timed voting systems that are accountable (and thus trustworthy) to their users;
- no full first-order logic is necessary;
- no real-time logic is necessary;
- modularity and multi-modality are crucial for the mental (and mechanical?) tractability of the specification.

Future work

- concrete *refinements* of our abstract specification towards more concrete implementation specifications such as a specification for the systems Prêt à Voter [Rya08] and Pretty Good Democracy [RT09];
- actual *verification* of concrete implementations w.r.t. these specifications.

R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi.
Reasoning about Knowledge.
MIT Press, 1995.

S. Kramer, R. Goré, and E. Okamoto.
Computer-aided decision-making with trust relations and trust domains (cryptographic applications).
Cryptology ePrint Archive, Report 2011/235, 2011.
http://eprint.iacr.org/.

S. Kramer.
Reducing provability to knowledge in multi-agent systems.
In *Proceedings of the LiCS-affiliated Intuitionistic Modal Logics and Applications Workshop*, 2008.
ftp://ftp.research.microsoft.com/pub/tr/TR-2008-90.pdf.

S. Kramer.
A logic of interactive proofs (formal theory of knowledge transfer).
Technical Report 1201.3667, arXiv, 2012.
http://arxiv.org/abs/1201.3667.

Z. Manna and A. Pnueli.
The Temporal Logic of Reactive and Concurrent Systems: Specification.
Springer, 1991.

P.Y.A. Ryan and V. Teague.
Pretty Good Democracy.
In *Proceedings of the Workshop on Security Protocols*, LNCS, 2009.

P.Y.A. Ryan.

Prêt à Voter with Paillier encryption.
Mathematical and Computer Modelling, 48(9-10), 2008.

Contact

`http://www.simon-kramer.ch`

`simon.kramer@a3.epfl.ch`