Themis: An On-Site Voting System with Systematic Cast-as-intended Verification and Partial Accountability

Mikaël Bougon, Hervé Chabanne, Véronique Cortier,
Alexandre Debant, Emmanuelle Dottax, Jannik Dreier,
Pierrick Gaudry, Mathieu Turuani

IDEMIA, France
Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

CCS Conference, Los Angeles, Nov. 10th 2022

On-site e-voting

Main goal: enhance the trust v.s. pure paper-based voting.

Security targets:

1. Vote secrecy: no one can know who I voted for
2. Verifiability: no one can modify the result of the election

voting machine can be compromised

Requirements in IDEMIA's use context

- limited access to on-site technology (Internet, printers, ..)
- robustness, e.g. resist power outage
- expect difficult contexts (corruptions, false accusations, ..)

On-site e-voting

Main goal: enhance the trust v.s. pure paper-based voting.

Security targets:

1. Vote secrecy: no one can know who I voted for
2. Verifiability: no one can modify the result of the election

voting machine can be compromised

Requirements in IDEMIA's use context

- limited access to on-site technology (Internet, printers, ..)
- robustness, e.g. resist power outage
- expect difficult contexts (corruptions, false accusations, ..)

Themis overview

Limited access to technology through:

- pre-printed paper ballots ➡ do not need printers
- smart cards and voting machines ➡ from the service provider
- hash-chain for the electronic ballot-box's integrity ➡ monitored offline a posteriori

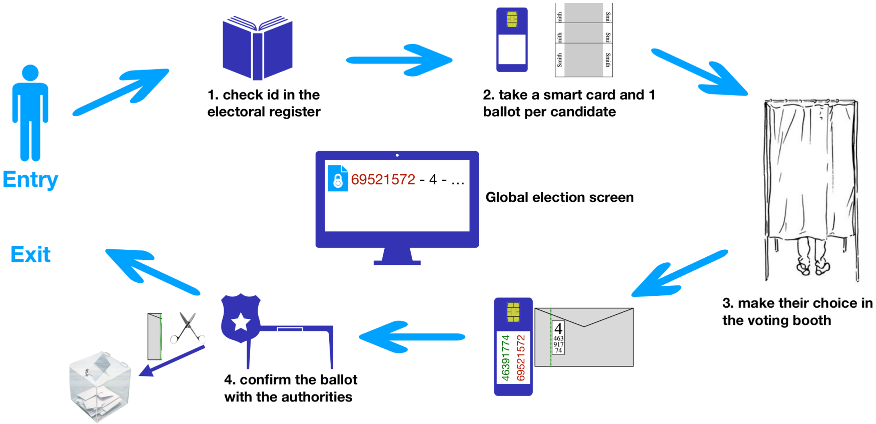
Robustness:

- verifiability (cast-as-inteded) and vote secrecy
- systematic cast-as-intended ➡ no need to trust smartcards
- can return to a pure paper-based voting system if needed

Difficult context:

- dispute resolution procedure to designate the culprit(s)
- proven to never wrongly blame someone
- require the corruption of several authorities to defeat vote secrecy or verifiability
- proven in symbolic models (ProVerif)

Themis polling station



Paper Ballot

Ballot contains :

- Candidate name with id X, plus verif. codes A and B s.t. $X = A + B \bmod n$
- SD_{paper} : a short digest (8 digits, unique) of

$$Digest_{paper} = Hash_{paper}(SN_{paper}, rand_{paper})$$

with SN_{paper} a unique serial number per paper ballot and $rand_{paper}$ a random value.

Ballot shows :

- Candidate name (aka. X), A, B, SD_{paper}
- QRCode 1 : SN_{paper}, $rand_{paper}$, with printer *signature*; (for dispute, does not break privacy)
- QRCode 2 : X, A, B, with printer *signature*.

Paper Ballot

Ballot contains :

- Candidate name with id X, plus verif. codes A and B s.t. $X = A + B \mod n$
- SD_{paper} : a short digest (8 digits, unique) of

$$Digest_{paper} = Hash_{paper}(SN_{paper}, rand_{paper})$$

with SN_{paper} a unique serial number per paper ballot and $rand_{paper}$ a random value.

Ballot shows :

- Candidate name (aka. X), A, B, SD_{paper}
- QRCode 1 : SN_{paper}, $rand_{paper}$, with printer *signature*;
  (for dispute, does not break privacy)
- QRCode 2 : X, A, B, with printer *signature*.

Inside the booth

Booth terminal:

- Waits for the voter's smartcard, and scan his paper ballot;
- Checks signatures (QRCodes) and data well-formedness;
- Shows the paper ballot's details on his screen (candidate, A, B)

Voter:

1. checks the screen v.s. his paper ballot, and confirm his choice
2. Place his paper ballot in the envelope (random direction)
   Envelope has a window to let see one of A or B, plus SD_{paper}.

SmartCard (in parallel to 2.):

1. Receives the paper ballot's QRCodes from the booth terminal;
   (also checks the signatures and well-formedness)
2. Sends the e-Ballot for display on the screen and in the chain.
3. On signed confirmation, shows SD_{elec} and SD_{paper} on card.

Inside the booth

Booth terminal:

- Waits for the voter's smartcard, and scan his paper ballot;
- Checks signatures (QRCodes) and data well-formedness;
- Shows the paper ballot's details on his screen (candidate, A, B)

Voter:

1. checks the screen v.s. his paper ballot, and confirm his choice
2. Place his paper ballot in the envelope (random direction)
   Envelope has a window to let see one of A or B, plus SD_{paper}.

SmartCard (in parallel to 2.):

1. Receives the paper ballot's QRCodes from the booth terminal; (also checks the signatures and well-formedness)
2. Sends the e-Ballot for display on the screen and in the chain.
3. On signed confirmation, shows SD_{elec} and SD_{paper} on card.

Electronic Ballot (concept)

e-Ballot contains :

3 ciphertexts c_X, c_A, c_B, and $\pi = \text{ZKP}(X \equiv A + B \bmod n)$

- proof that $X \equiv A + B \bmod n$
- proof that X, A, B are integers between $0..n-1$

Voter audit request: either A or B, chose secretly in the voting booth
$\Rightarrow$ ignored by the smartcard, the terminals and the server

Smartcard and e-Ballot audit:
$\Rightarrow$ e-Ballot created in the voting booth and added to the chain + screen
$\Rightarrow$ before confirmation, Voter must see his ballot on the screen
$\Rightarrow$ SmartCard provides the random used to encrypt the audited A or B

Auditors: see and check A or B plus random, for each ballot in the chain.

Ballot manipulation detected with probability 1/2 (on each ballot)

Electronic Ballot (concept)

e-Ballot contains :

3 ciphertexts c_X, c_A, c_B, and $\pi = \text{ZKP}(X \equiv A+B \bmod n)$

- proof that $X \equiv A+B \bmod n$
- proof that X, A, B are integers between $0..n-1$

Voter audit request: either A or B, chose secretly in the voting booth
$\Rightarrow$ ignored by the smartcard, the terminals and the server

Smartcard and e-Ballot audit:
$\Rightarrow$ e-Ballot created in the voting booth and added to the chain + screen
$\Rightarrow$ before confirmation, Voter must see his ballot on the screen
$\Rightarrow$ SmartCard provides the random used to encrypt the audited A or B

Auditors: see and check A or B plus random, for each ballot in the chain.

Ballot manipulation detected with probability 1/2 (on each ballot)

Electronic Ballot

e-Ballot contains :

- $c_X = enc(pk, X)$, $c_A = enc(pk, A)$, $c_B = enc(pk, B)$
  with pk the election's public key.
- 1 ZKP: π = ZKP($X \equiv A + B \bmod n$)
  $\Rightarrow$ also ensures A is odd, B is even, in $0..n-1$
  (note n is twice the number of candidates)
- A digest similar to the $Digest_{paper}$:

  $$Digest_{elec} = Hash_{elec}(SN_{paper}, rand_{elec})$$

  with SN_{paper} the paper's ballot (unique) serial number
  and $rand_{elec}$ a random value.
- A SmartCard's signature on this e-Ballot.

Loop until the short 8-digits digest SD_{elec} is unique in the ballot's chain.

Confirmation and election screen

Election screen and chain of blocks matches thanks to Auditors

Voter arrival, SmartCard connected :

- e-Ballot's state moved to **Under Confirmation**
- Voter + Official together confirms SD_{elec} and SD_{paper} (screen v.s. card, and paper ballot v.s. card)

Voter scans the envelope's window :

- Audited digit shown on the screen, move to **Scanned Code**
- Voter + Official together confirm the scan

SmartCard answers the challenge :

- Digit and random are shown for audit, move to **In Audit**
- SmartCard answers one signed challenge only, shows sign. otherwise
- SmartCard will be blamed if not answering the 1st challenge.

Confirmation and election screen

Election screen and chain of blocks matches thanks to Auditors

Voter arrival, SmartCard connected :

- e-Ballot's state moved to **Under Confirmation**
- Voter + Official together confirms SD_{elec} and SD_{paper} (screen v.s. card, and paper ballot v.s. card)

Voter scans the envelope's window :

- Audited digit shown on the screen, move to **Scanned Code**
- Voter + Official together confirm the scan

SmartCard answers the challenge :

- Digit and random are shown for audit, move to **In Audit**
- SmartCard answers one signed challenge only, shows sign. otherwise
- SmartCard will be blamed if not answering the 1st challenge.

Confirmation and election screen

Election screen and chain of blocks matches thanks to Auditors

Voter arrival, SmartCard connected :

- e-Ballot's state moved to **Under Confirmation**
- Voter + Official together confirms SD_{elec} and SD_{paper} (screen v.s. card, and paper ballot v.s. card)

Voter scans the envelope's window :

- Audited digit shown on the screen, move to **Scanned Code**
- Voter + Official together confirm the scan

SmartCard answers the challenge :

- Digit and random are shown for audit, move to **In Audit**
- SmartCard answers one signed challenge only, shows sign. otherwise
- SmartCard will be blamed if not answering the 1st challenge.

Confirmation and election screen

Audit from Voter + Officials :

- Both check together that the numbers match
- This confirms the Vote, e-Ballot move to **Confirmed**.

Paper ballot : keep candidate only, sent to the ballot box
SmartCard : reset and returned to the pool.

Precautions

- The e-Ballot is fixed prior to uncovering A or B.
  $\Rightarrow$ the SmartCard or "system" cannot change it anymore
- Voter and Official must agree on A or B prior sending the challenge
  $\Rightarrow$ limit later complain on the value sent
  $\Rightarrow$ the challenge sent is absolute and *must* be answered.
- The server is responsible for anything added to the chain or screen.

Confirmation and election screen

Audit from Voter + Officials :

- Both check together that the numbers match
- This confirms the Vote, e-Ballot move to **Confirmed**.

Paper ballot : keep candidate only, sent to the ballot box
SmartCard : reset and returned to the pool.

Precautions

- The e-Ballot is fixed prior to uncovering A or B.
  $\Rightarrow$ the SmartCard or "system" cannot change it anymore
- Voter and Official must agree on A or B prior sending the challenge
  $\Rightarrow$ limit later complain on the value sent
  $\Rightarrow$ the challenge sent is absolute and *must* be answered.
- The server is responsible for anything added to the chain or screen.

Observers

Does the presence of Observers change the result of the election ?

Maybe yes !

Electronic Ballot Box published and rechecked at the end;

Snapshots of the screen during the votes :

- Check QRCodes for signatures and data inconsistencies.

Compare Snapshots with the Electronic Ballot Box;

Compare nb. of e-Ballots with the register

Optional Audits :

- Risk-limiting audits on the paper ballots;
- SmartCard and Terminals can be audited;
- Destructive audit of some (random) paper ballots.

Observers

Does the presence of Observers change the result of the election ?

Maybe yes !

Electronic Ballot Box published and rechecked at the end;

Snapshots of the screen during the votes :

- Check QRCodes for signatures and data inconsistencies.

Compare Snapshots with the Electronic Ballot Box;

Compare nb. of e-Ballots with the register

Optional Audits :

- Risk-limiting audits on the paper ballots;
- SmartCard and Terminals can be audited;
- Destructive audit of some (random) paper ballots.

Dispute resolution (confirmation fails)

Phase 1 (in the polling station)

- *Preserves* the vote secrecy
- Partial opening of the envelope (uncovers 1st QRCode) i.e. SN_{paper} and $rand_{paper}$, with printer *signature*.
- Checks data v.s. corresponding SmartCard's records
- Checks e-Ballot box v.s. SmartCard records

Phase 2 (outside of the station, external Auditors)

- Paper ballot (inside it's envelope) plus the SmartCard are kept for further (offline) analysis
- Could be e.g. an attack attempt from inside the polling station
- Could be e.g. a forged fake paper ballot inside the envelope
- Complete analysis uncovers the paper ballot completely
- But the SmartCard will still never reveal the vote by herself.

Dispute resolution (confirmation fails)

Phase 1 (in the polling station)

- *Preserves* the vote secrecy
- Partial opening of the envelope (uncovers 1st QRCode) i.e. SN_{paper} and $rand_{paper}$, with printer *signature.*
- Checks data v.s. corresponding SmartCard's records
- Checks e-Ballot box v.s. SmartCard records

Phase 2 (outside of the station, external Auditors)

- Paper ballot (inside it's envelope) plus the SmartCard are kept for further (offline) analysis
- Could be e.g. an attack attempt from inside the polling station
- Could be e.g. a forged fake paper ballot inside the envelope
- Complete analysis uncovers the paper ballot completely
- But the SmartCard will still never reveal the vote by herself.

Model overview (ProVerif)

Flexibility :

- All interactions goes through channels, including non-electronic ones
- Each scenario describes it's honesty/dishonest assumptions
  $\Rightarrow$ Models derived from a single, main one (for reachability prop.)

Observers :

- Not modeled as agents but as restrictions (consistency properties)
- Only traces where the Observers are satisfies are considered

End-to-end Verifiability (part 1)

Individual verifiability (aka. recorded-as-intended)

Combines cast-as-intended (after confirmation) and recorded-as-cast

Assuming :

1. All checks from Voter & Auditors succeed
2. Paper ballot was well-formed
3. Voter do not trust authorities or the 'system'

Prop: Each voter is assured that some valid ballot containing his intended vote exists for him in the database

This is split in two subproperties to ease ProVerif analysis :

1. **recorded-as-intended without voters to ballots injectivity**
   i.e. allows to wrongfully associate two voters to one same ballot.
2. **no-clash-attack**
   i.e. two happy voters cannot share the same ballot.

End-to-end Verifiability (part 2)

Eligibility

Entry the polling station is not part of the protocol
Neither is the link with the record (human check)
$\Rightarrow$ remains only : **no-ballot-stuffing**

Two ways to ensure :

1. Through local authorities, by comparing the number of paper and electronic ballots;
2. By design if the scenario allows it, and counting is only a safeguard $\Rightarrow$ targeted here

Counted-as-recorded

The tally is unspecified in the protocol, so this property is not analyzed;
Usual tallying methods are expected to work as usual here.

End-to-end Verifiability (part 2)

Eligibility

Entry the polling station is not part of the protocol
Neither is the link with the record (human check)
$\Rightarrow$ remains only : **no-ballot-stuffing**

Two ways to ensure :

1. Through local authorities, by comparing the number of paper and electronic ballots;
2. By design if the scenario allows it, and counting is only a safeguard $\Rightarrow$ targeted here

Counted-as-recorded

The tally is unspecified in the protocol, so this property is not analyzed;
Usual tallying methods are expected to work as usual here.

Modeling difficulties

Probabilities for recorded-as-intended

- Attack should be detected with only prob. 1/2 only;
- **Assumption :**
  - adversary cannot anticipate which code will be audited;
- **ProVerif model :** both codes are audited (honest agents)
  - i.e. models two runs inside one;
  - adversary failure means he failed at least to one of both audits.

Assumption is easy to prove in the modeled scenario, but side-channels attacks (e.g. camera in the booth) would break it;

This allows to abstract the probabilities away from the model.

Modeling difficulties

Arithmetic of $X = A + B$ mod n

Over-approximation through events and restrictions :

- Model each agent verification over X through an event
- For $X =^? A + B$ mod n : event isSum(X, A, B)
- For $X \neq^? A + B$ mod n : event isNotSum(X, A, B)

Define a set of restrictions to model the few and only extra deductions that ProVerif needs when building the Horn Clauses, e.g. :

$$\texttt{isSum}(x, a, b) \wedge \texttt{isSum}(x, a, b') \rightarrow b = b'$$

$$\texttt{isSum}(x, a, b) \wedge \texttt{isNotSum}(x, a, b') \rightarrow b \neq b'$$

This also shows a (over-approximated) set of deductions on X, A, B that this protocol needs to be secure.

Query sample (and strongly simplified)

From recorded-as-intended :

$$\begin{array}{ll} & \text{event HappyUser}(\text{Voter},\text{Candidate},SD_{elec},A,B) \\ \wedge & \text{event Snapshot}(SD_{elec},\ \text{data}) \\ \wedge & \text{event isSum}(X,A,B) \quad \Rightarrow \quad \text{Candidate} = \text{GetName}(X) \end{array}$$

With

- HappyUser : Voter confirmed with SD_{elec} and think he voted for Candidate;
- Snapshot : an Observer spotted a ballot with SD_{elec} on the election screen;
  $\Rightarrow$ thanks to QRCode audits, it is assumed to contain consistent data
- GetName : function from candidate *id* to real name.

Analysis results

Individual verifiability holds when (both conditions) :

1. The election screen can be trusted and matches the e-Ballot box
   - either because the server is honest
   - or Observers are present to monitor the server
2. The paper ballot was well-formed
   - either because the Printing Authority was honest
   - or the Devices were honest and thus, checked it.

Note: Voter needs addition modulo if the cart or terminal is dishonest.

No-ballot-stuffing holds when both the local authorities and the smartcard are honest

- Fallback to counting ballots if only the authorities are honest;
- Dishonest authorities can let through false voters in the process.

Analysis results

Individual verifiability holds when (both conditions) :

1. The election screen can be trusted and matches the e-Ballot box
   - either because the server is honest
   - or Observers are present to monitor the server
2. The paper ballot was well-formed
   - either because the Printing Authority was honest
   - or the Devices were honest and thus, checked it.

Note: Voter needs addition modulo if the cart or terminal is dishonest.

No-ballot-stuffing holds when both the local authorities and the smartcard are honest

- Fallback to counting ballots if only the authorities are honest;
- Dishonest authorities can let through false voters in the process.

Analysis result (Individual verifiability)

Participants											
Observer		1	1	1	1	0	0	0	0	0	0
Printing Authority*		1	0	0	0	x	x	x	x	x	x
Local Authorities		x	x	x	x	x	x	x	x	x	x
Devices	Smart card		1	x	0	x	0	1	x	0	1
	Terminals	x	1	0	1	0	1	1	0	1	1
	Server		x	x	x	0	0	0	1	1	1
Results											
NI-recorded-as-intended		✓	✓ (5min)	✗*	✗*	✗	✗	✗†	✗*	✗*	✓ (5min)
No clash attacks		✓	✓	✓	✓	✗	✗	✗	✓	✓	✓

* property proved if the voter verifies $\text{verif}_a \neq \text{verif}_b$ and $\text{id}_{cand} = \text{verif}_a + \text{verif}_b$

† property proved if the display of the global election scree is consistent with the content of the hashchain

Analysis result (no-clash-attack)

Participants				
Observer		x	x	x
Printing Authority		x	x	x
Local Authorities		0	1	1
Devices	Smart card	x	x	1
	Terminals	x	x	x
	Server	x	x	x
Results				
No ballot stuffing		✗	✓⋆	✓

⋆ requires to compare the number of paper and electronic ballots to prevent malicious additions/deletions. Cannot be proved in ProVerif

Vote Privacy

Biprocess for privacy

- Assume Alice and Bob audit their first code, A resp. A';
- It will be revealed, so must not change through the processes;

$$\begin{aligned} \mathrm{P} = \mathrm{C} \mid & \mathrm{Alice}(\mathrm{diff}[\ \mathrm{ballot}(X, A, B_1),\ \mathrm{ballot}(Y, A, B_3)\ ]\ ) \\ \mid & \mathrm{Bob}(\ \mathrm{diff}[\ \mathrm{ballot}(Y, A', B_2),\ \mathrm{ballot}(X, A', B_4)\ ]\ ) \end{aligned}$$

with $X \equiv A + B_1 \equiv A' + B_4 \bmod n$ and $Y \equiv A + B_3 \equiv A' + B_2 \bmod n$

Problem: the restrictions for arithmetic creates an over-approximation !

Solution with both a ProVerif lemma and a hand proof :

1. the IsSum(..) relation is preserved from left to right in this biprocess
2. hand-proof to lift this to vote privacy w.r.t. arithmetic operations.

Analysis results

Vote privacy holds (in general)
versus a single corrupted entity;

Noticeable exceptions:

- Local authorities provide an invalid paper ballot to a targeted Voter
  $\Rightarrow$ observe if he returns
- Similar for the printing authority with a local accomplice.
- Auditing rejected ballots might reduce the risk

Participants		Dis. local		Dis. printer		
Printing Authority		1	1	0	0	0
Local	Authorities	0	0	1	1	1
	Accomplice	x	x	1	0	0
Devices	Smart card	1	1 dis.	x	1 dis.	1
	Voting term.	1		x		1
	Confirm. term.	1		x		1
	Server	x	x	x	x	x
Results						
Privacy		✓[*] (3h57)	✗	✓	✗	✓[**] (47min)

[*] assume that local authority cannot forge fake ballots.
[**] assume random audits to detect fake paper ballots in the stack.

Accountability divided

Defendability

Dispute resolution always end with a blame accusation
⇒ Honest participant expects **not to be blamed**.

In some cases, a group of participants is to be blamed, meaning that one of them was guilty (but not necessarily the others)

All scenarios proved, covering all exit cases for the dispute resolution;
Some scenario cannot blame one single agent, but a group among which one is guilty :

- Mainly due to fake paper ballots in the process;
  - from the Printing Authority or a local agent ?
  - from the Voter, armed with a pair of scissors ?
- Further, human-level analysis might better point the culprit.

Accountability divided

Defendability

Dispute resolution always end with a blame accusation
⇒ Honest participant expects **not to be blamed**.

In some cases, a group of participants is to be blamed, meaning that one of them was guilty (but not necessarily the others)

All scenarios proved, covering all exit cases for the dispute resolution;
Some scenario cannot blame one single agent, but a group among which one is guilty :

- Mainly due to fake paper ballots in the process;
    - from the Printing Authority or a local agent ?
    - from the Voter, armed with a pair of scissors ?
- Further, human-level analysis might better point the culprit.

Accountability divided

Contestability

The Voter always terminates with: either a success; or a dispute resolution; or a return to booth.

No liveness (the Voter can always continue), but not possible with ProVerif and easy to check by hand.

Card-capture resistance

Each time the dispute resolution holds, a card is captured. However :

1. Occurs only if system is dishonest or a fake paper ballot is used;
2. Fake paper ballots leading to a card captured have specific shapes;
   $\Rightarrow$ countermeasures ?
3. Honest Voters are not subject to use fake paper ballots by accident.
   $\Rightarrow$ with honest terminal or audit of the ballots

Accountability divided

Contestability

The Voter always terminates with: either a success; or a dispute resolution; or a return to booth.

No liveness (the Voter can always continue), but not possible with ProVerif and easy to check by hand.

Card-capture resistance

Each time the dispute resolution holds, a card is captured. However :

1. Occurs only if system is dishonest or a fake paper ballot is used;
2. Fake paper ballots leading to a card captured have specific shapes;
   $\Rightarrow$ countermeasures ?
3. Honest Voters are not subject to use fake paper ballots by accident.
   $\Rightarrow$ with honest terminal or audit of the ballots

Conclusion

On-site voting protocol with systematic audits and dispute resolution;
Large ProVerif modeling and analysis, despite modular arithmetic.

Questions ?