Election Verifiability in Receipt-free Voting Protocols

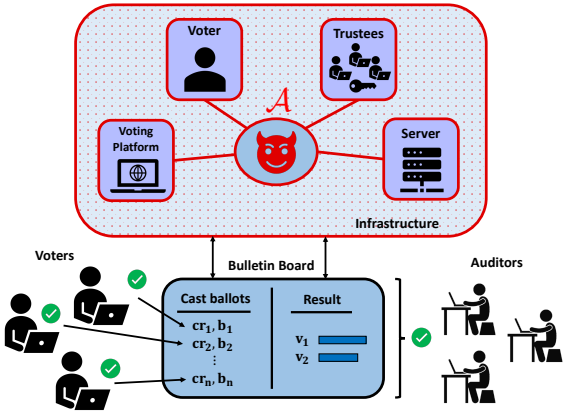
Sevdenur Baloglu[1], Sergiu Bursuc[1], Sjouke Mauw[2], Jun Pang[2]

[1] SnT, University of Luxembourg
[2] DCS, University of Luxembourg

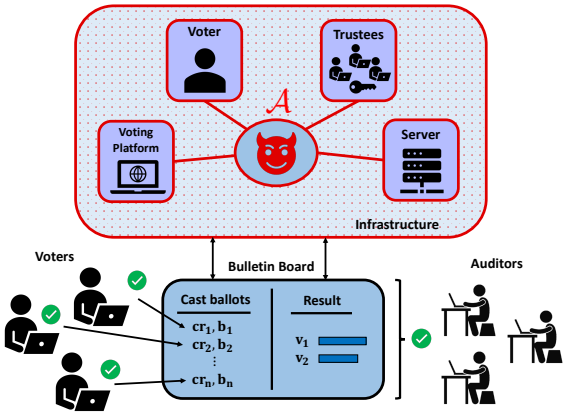
E-mail: *sevdenur.baloglu@uni.lu*

IEEE CSF23, July 10

End-to-end verifiability



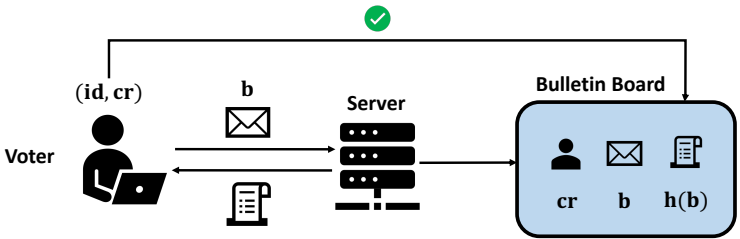
End-to-end verifiability



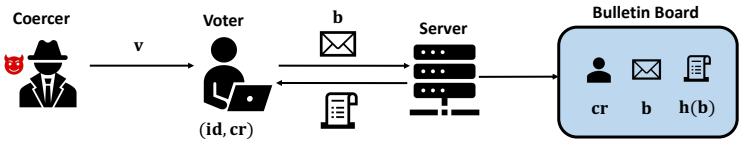
Goal: Automated and realistic formal verification of voting protocols

$$\mathcal{S} \models \Phi \qquad \text{Tamarin/ProVerif}$$

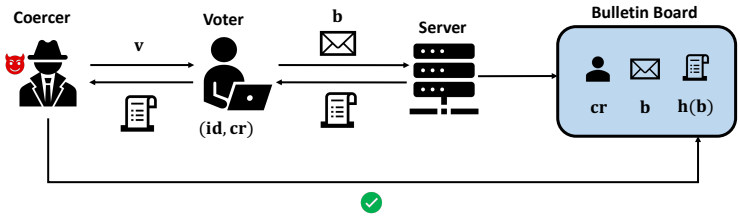
Individual verification



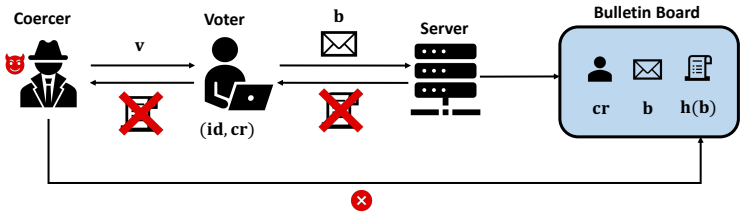
Coercion



Receipt-freeness vs. verification



Receipt-freeness vs. verification



Our contributions

1. General election verifiability definition
   - improves the state of the art
     - covers receipt-free systems
     - more closely captures end-to-end verifiability
   - allows automated verification with Tamarin/ProVerif
2. Specification and verification of receipt-free protocols
   - BeleniosRF: attacks and new proofs
   - Selene: new proofs

Specifications of protocols and properties

$$\mathcal{S} \models \Phi$$

$\mathcal{S}$: specification for receipt-free voting protocols

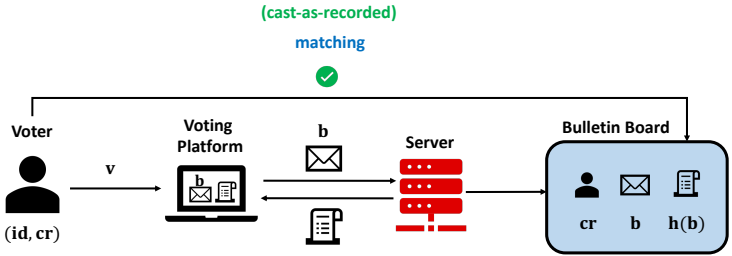
- $\mathcal{P}$: honest parties
- $\mathcal{A}$: corrupt parties and adversary
- $\mathcal{E}$: cryptographic primitives

Φ : specification of security

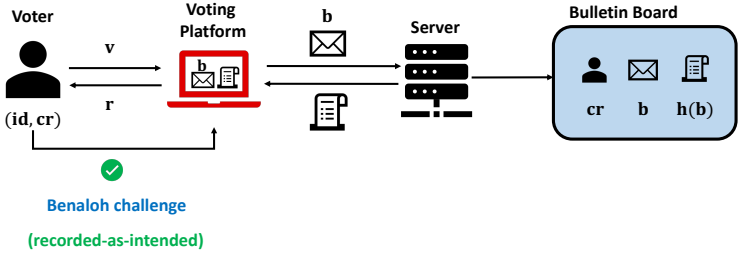
- Φ_{E2E} : end-to-end verifiability

$\Longrightarrow$ automated analysis with Tamarin/ProVerif

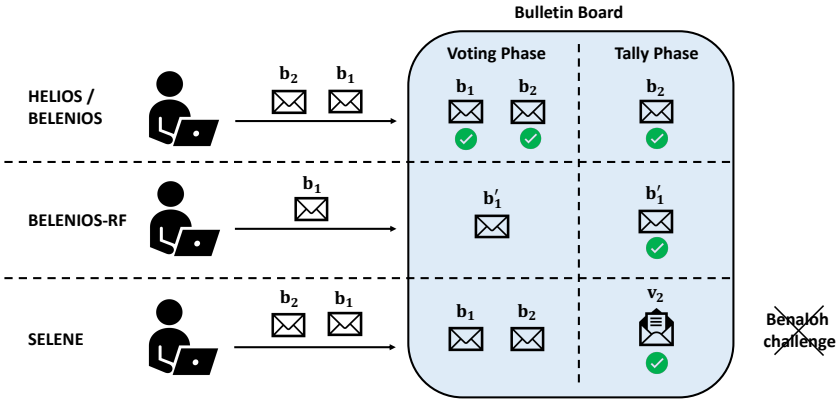
Individual verification



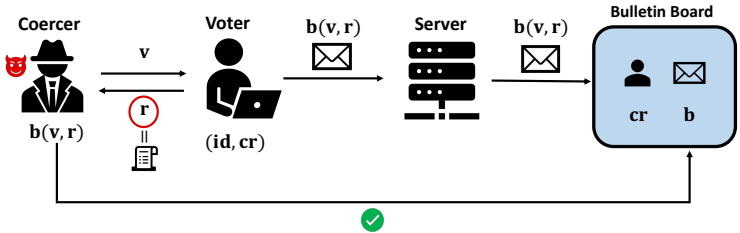
Individual verification



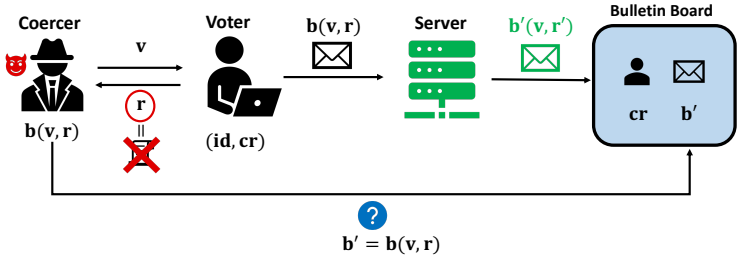
Prominent voting systems and receipt-freeness



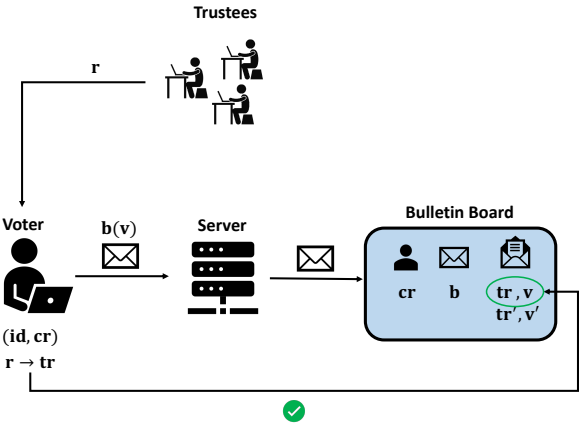
Helios/Belenios



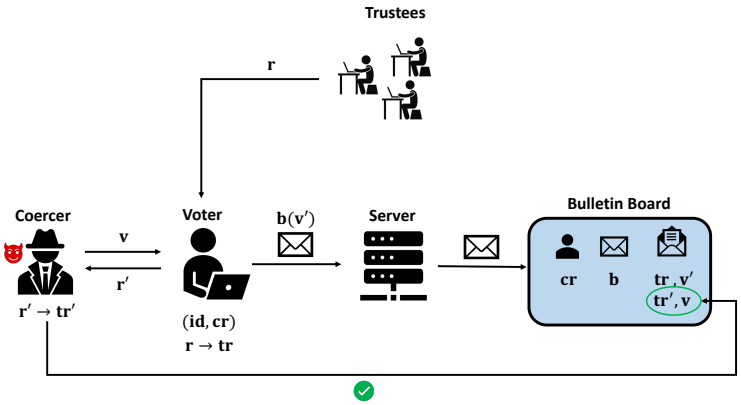
BeleniosRF


Coercer
v
r
b(v, r)
Voter
(id, cr)
b(v, r)
Server
b′(v, r′)
Bulletin Board
cr
b′
?
b′ = b(v, r)

Selene


Trustees
r
Voter
b(v)
Server
Bulletin Board
cr
b
tr , v
tr′, v′
(id, cr)
r → tr

Selene


Trustees
r
Coercer
v
r′
r′ → tr′
Voter
(id, cr)
r → tr
b(v′)
Server
Bulletin Board
cr
b
tr , v′
tr′, v

Previous work

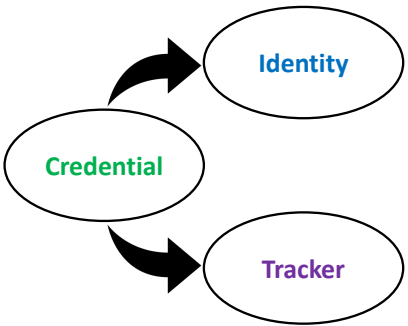
Cortier et al., "BeleniosVS: Secrecy and Verifiability Against a Corrupted Voting Device", CSF 2019.

+ allows automated analysis
- specific to BeleniosVS
- no revoting
- varies according to corruption scenarios

Baloglu et al., "Election verifiability revisited: Automated security proofs and attacks on Helios and Belenios", CSF 2021.

+ allows automated analysis
+ applicable to a broader class of protocols
+ allows revoting
+ independent of corruption scenarios
- credential specific (not voter identities)
- finds attacks related to registration procedure

Improved end-to-end verifiability definition



Identities: $\mathsf{id}_1 \quad \ldots \quad \mathsf{id}_n$

$\Updownarrow \quad \ldots \quad \Updownarrow$

Credentials: $\mathsf{cr}_1 \quad \ldots \quad \mathsf{cr}_n$

$\Updownarrow \quad \ldots \quad \Updownarrow$

Trackers: $\mathsf{tr}_1 \quad \ldots \quad \mathsf{tr}_n$

Symbolic definition

$\mathcal{S}$: the protocol specification (set of multiset rewriting rules or processes)

Φ : the property specification (based on events)

τ : an execution trace (sequence of events)

$$\mathcal{S} \models \Phi \iff \forall \tau \in tr(\mathcal{S}).\ \tau \models \Phi$$

Main challenge: general, simple and sound set of formulas Φ

Events

E-voting events:

- BBreg(cr)
- BBres(tr, v)
- Vote(id, v)
- Verified(id, cr, v)
- Corr(id)
- Reg(id, cr)
- Link(cr, tr)

	id_1	...	id_n
	$Reg(id_1, cr_1)$	...	$Reg(id_n, cr_n)$
BBreg:	cr_1	...	cr_n
	$Link(cr_1, tr_1)$	...	$Link(cr_n, tr_n)$
BBres:	(tr_1, v_1)	...	(tr_n, v_n)

Symbolic end-to-end verifiability

$$\Phi^{\diamond}_{\mathsf{E2E}} : \Phi_{\mathsf{iv}} \wedge \Phi_{\mathsf{cl}} \wedge \Phi_{\mathsf{eli}} \wedge \Phi^{\diamond}_{\mathsf{res}} \wedge \Phi_{\mathsf{cons}}, \quad \diamond \in \{\circ, \bullet\}$$

$$\Phi_{\mathsf{eli}} : \quad \mathsf{BBres}(\mathsf{tr}, \mathsf{v}) \implies \mathsf{Link}(\mathsf{cr}, \mathsf{tr}) \wedge \mathsf{BBreg}(\mathsf{cr}) \wedge \mathsf{Reg}(\mathsf{id}, \mathsf{cr})$$

Symbolic end-to-end verifiability

$$\Phi^{\diamond}_{\mathsf{E2E}} : \Phi_{\mathsf{iv}} \wedge \Phi_{\mathsf{cl}} \wedge \Phi_{\mathsf{eli}} \wedge \Phi^{\diamond}_{\mathsf{res}} \wedge \Phi_{\mathsf{cons}}, \quad \diamond \in \{\circ, \bullet\}$$

$$\begin{aligned}
\Phi_{\mathsf{eli}} : &\quad \mathsf{BBres}(\mathsf{tr}, \mathsf{v}) \implies \mathsf{Link}(\mathsf{cr}, \mathsf{tr}) \wedge \mathsf{BBreg}(\mathsf{cr}) \wedge \mathsf{Reg}(\mathsf{id}, \mathsf{cr}) \\
\Phi_{\mathsf{iv}} : &\quad \mathsf{Verified}(\mathsf{id}, \mathsf{cr}, \mathsf{v}) \wedge \Omega(\mathsf{id}, \mathsf{v}) \wedge \mathsf{Link}(\mathsf{cr}, \mathsf{tr}) \wedge \mathsf{BBres}(\mathsf{tr}, \mathsf{v}') \implies \mathsf{v} = \mathsf{v}' \\
\Omega(\mathsf{id}, \mathsf{v}) : &\quad \text{revote policy}
\end{aligned}$$

Symbolic end-to-end verifiability

$$\Phi^{\diamond}_{\mathsf{E2E}} : \Phi_{\mathsf{iv}} \wedge \Phi_{\mathsf{cl}} \wedge \Phi_{\mathsf{eli}} \wedge \Phi^{\diamond}_{\mathsf{res}} \wedge \Phi_{\mathsf{cons}}, \quad \diamond \in \{\circ, \bullet\}$$

$$\begin{aligned}
\Phi_{\mathsf{eli}} : &\quad \mathsf{BBres}(\mathsf{tr}, \mathsf{v}) \implies \mathsf{Link}(\mathsf{cr}, \mathsf{tr}) \wedge \mathsf{BBreg}(\mathsf{cr}) \wedge \mathsf{Reg}(\mathsf{id}, \mathsf{cr}) \\
\Phi_{\mathsf{iv}} : &\quad \mathsf{Verified}(\mathsf{id}, \mathsf{cr}, \mathsf{v}) \wedge \Omega(\mathsf{id}, \mathsf{v}) \wedge \mathsf{Link}(\mathsf{cr}, \mathsf{tr}) \wedge \mathsf{BBres}(\mathsf{tr}, \mathsf{v}') \implies \mathsf{v} = \mathsf{v}' \\
\Omega(\mathsf{id}, \mathsf{v}) : &\quad \text{revote policy} \\
\Phi_{\mathsf{reg}} : &\quad \text{consistency of credentials} \\
\Phi_{\mathsf{link}} : &\quad \text{consistency of trackers}
\end{aligned}$$

Soundness Theorem: $\Phi^{\diamond}_{\mathsf{E2E}} \implies \mathsf{E2E}$

Analysis: BeleniosRF

$b = \langle c, s, \pi \rangle$

$\mathcal{E}$: equational theory

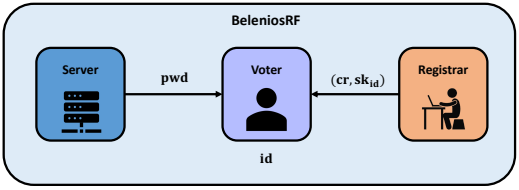
$\downarrow r$

$c \longrightarrow c'$

$s \longrightarrow s'$

$\pi \longrightarrow \pi'$

$b' = \langle c', s', \pi' \rangle$



Trust assumptions:

- either registrar or server is honest

Analysis: BeleniosRF

$b = \langle c, s, \pi \rangle$

$\mathcal{E}$: equational theory

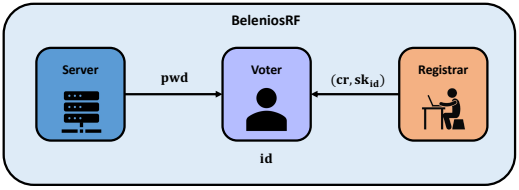
$\downarrow r$

$c \longrightarrow c'$

$s \longrightarrow s'$

$\pi \longrightarrow \pi'$

$b' = \langle c', s', \pi' \rangle$



Trust assumptions:

- either registrar or server is honest

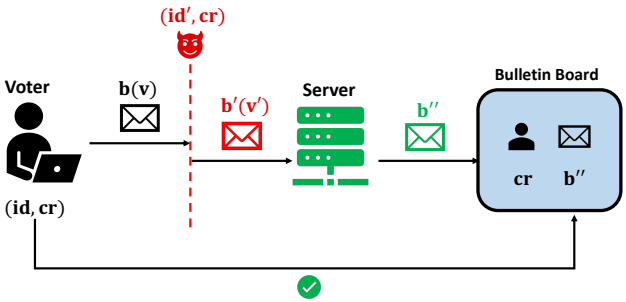
Attacks: (when the registrar is corrupt)

- IV attack
- clash attack

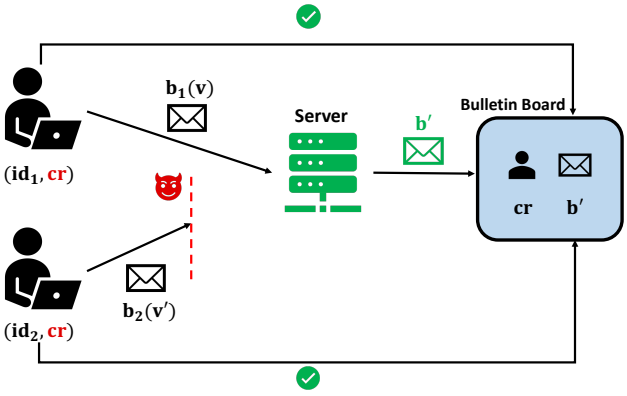
BeleniosRF: automated verification with ProVerif

Corruption Models		$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$
Trustees		H	C	C	C
Registrar		H	H	C	H
Server		H	H	H	C
Voting Platform		H	H	H	H
Belenios	$\Phi^{\diamond}_{\text{E2E}}$	✓	✓	✓	✓
BeleniosRF	$\Phi^{\diamond}_{\text{E2E}}$	✓	✓	✗	✓

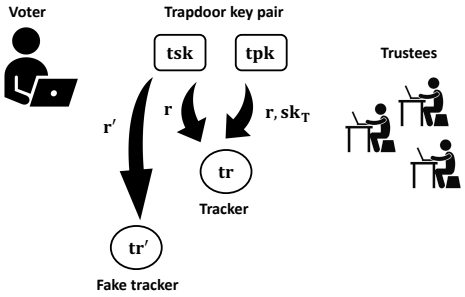
BeleniosRF: IV attack



BeleniosRF: clash attack



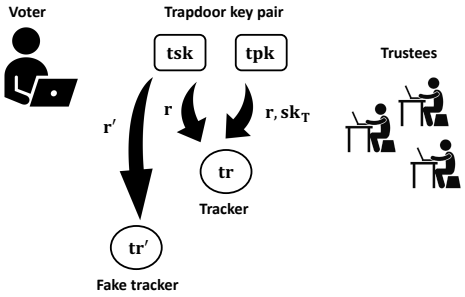
Analysis: Selene



Trust assumptions:

- either trustees or voting platform are honest

Analysis: Selene



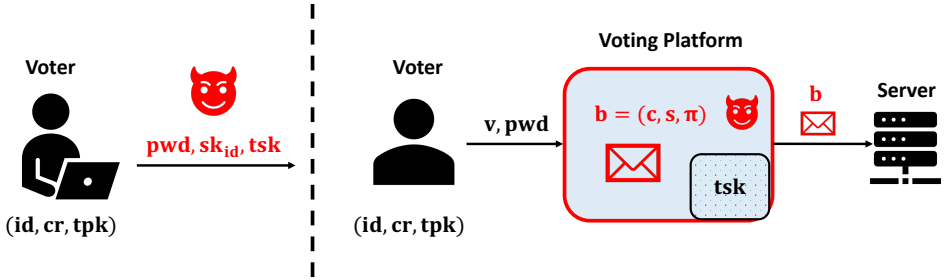
Trust assumptions:

- either trustees or voting platform are honest

Variants:

- **Hyperion:** simplified tracker construction (no trapdoor commitments)
- **SeleneRF:** improved receipt-freeness (based on rerandomised ballot)

Corrupt voter vs. corrupt voting platform



Selene: automated verification with ProVerif

Corruption Models	$\mathcal{A}'_1$	$\mathcal{A}'_2$	$\mathcal{A}'_3$	$\mathcal{A}'_4$	$\mathcal{A}'_5$	$\mathcal{A}'_6$	$\mathcal{A}'_7$	$\mathcal{A}'_8$	$\mathcal{A}'_9$	$\mathcal{A}'_{10}$
Trustees	H	H	H	H	C	C	C	C	C	C
Registrar	H	H	C	H	H	C	H	H	C	H
Server	H	H	H	C	H	H	C	H	H	C
Voting Platform	H	C	C	C	H	H	H	C	C	C
$\Phi^{\bullet}_{\text{E2E}}$	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
$\Phi^{\circ}_{\text{E2E}}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Conclusion

- a general formal verification framework for election verifiability that
  - allows automated proofs of security
  - accounts for receipt-free protocols
  - allows refined security according to adversary models
- the first symbolic verifiability analysis of BeleniosRF, Selene and its variants
  - new specifications
  - new attacker models and attacks
  - new security proofs
- neither BeleniosRF nor Selene satisfies strong end-to-end verifiability

Future work

- improvement of BeleniosRF against found attacks
  - confirmation message
- stronger end-to-end verifiability for Selene
- weaker trust assumptions
  - BeleniosRF when both the registrar and server are corrupt
  - Selene when both talliers and voting platform are corrupt
- improvement of the tools: Tamarin and ProVerif
  - BeleniosRF: $\mathsf{enc}(v, pk, r), r' \longrightarrow \mathsf{enc}(v, pk, r + r')$
  - Hyperion and Selene: more general models for exponentiation
- automated proofs of privacy and receipt-freeness

Thank you for listening!