Election Verifiability Revisited: Automated Security Proofs and Attacks on Helios and Belenios

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 CSF'21, June 22

Introduction

Traditional Voting		Electronic Voting
paper ballots	$\xrightarrow{\text{e-voting}}$	electronic ballots
voting booths	$\xrightarrow{\text{e-voting}}$	voting platforms
ballot boxes	$\xrightarrow{\text{e-voting}}$	voting server
adversary has limited corruption abilities	$\xrightarrow{\text{e-voting}}$	adversary has extended corruption abilities

Election Verifiability

Electronic voting protocols should be **verifiable**:

- by voters to ensure their vote is counted in the final tally,
- by anybody to ensure the final outcome reflects the intention of eligible voters.

A system satisfying those is called **end-to-end verifiable**.

Our focus: Formal verification of election verifiability

Previous Work on formal verification of election verifiability

- S. Kremer, M. Ryan, and B. Smyth, "Election verifiability in electronic voting protocols," in ESORICS'10.
- V. Cortier, F. Eigner, S. Kremer, M. Maffei, and C. Wiedling, "Type-based verification of electronic voting protocols," in POST'15.
- V. Cortier, A. Filipiak, and J. Lallemand, "BeleniosVS: Secrecy and verifiability against a corrupted voting device," in IEEE CSF'19.

We propose an extension of Cortier et al. model: more general and covering new features desired in practice.

Limitations and Contributions

Features	Cortier et al. model	Our model
automated verification	✓	✓
soundness	✓	✓
general protocols and corruption scenarios	✗	✓
revoting	✗	✓
clash attacks	✗	✓
dynamic corruption	✗	✓
quantum-resistant protocols	✗	✗

We apply our definition to various corruption scenarios and verification procedures in Helios and Belenios, and we find **new attacks**.

Modeling protocols in Tamarin

Tamarin is based on multiset rewriting rules:

$$\text{rule}: \quad [\ \textbf{Premise}\ ] - [\ \text{Action}\ ] \rightarrow [\ \textbf{Conclusion}\ ],$$

where **Premise**, Action and **Conclusion** are sets of facts specifying terms.

Verifying protocols in Tamarin

Tamarin is based on multiset rewriting rules:

$$\text{rule}: \quad [\,\textbf{Premise}\,] - [\,\text{Action}\,] \rightarrow [\,\textbf{Conclusion}\,],$$

where **Premise**, Action and **Conclusion** are sets of facts.

- $\mathcal{S}$: the set of rules specifying the protocol,
- Φ : the formula specifying the desired property based on action facts.

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

E-voting Events as Action Facts

Tamarin is based on multiset rewriting rules:

$$\text{rule}: \quad [\ \textbf{Premise}\ ] - [\ \text{Action}\ ] \rightarrow [\ \textbf{Conclusion}\ ],$$

where **Premise**, Action and **Conclusion** are sets of facts.

We use action facts to record events:

- $\text{BBreg}(cr)$: eligible public credential
- $\text{Vote}(id, cr, v)$: the vote is cast by the voter
- $\text{Verified}(id, cr, v)$: the vote is verified by the voter
- $\text{BBtally}(cr, b)$: the ballot is tallied for cr
- $\text{Corr}(id, cr)$: corrupted voter

End-to-end Election Verifiability

$$\Phi^{\diamond}_{\mathsf{E2E}} = \Phi_{\mathsf{iv}} \ \wedge \ \Phi_{\mathsf{eli}} \ \wedge \ \Phi_{\mathsf{cl}} \ \wedge \ \Phi^{\diamond}_{\mathsf{res}}$$

Φ_{iv} : Individual verification implies the corresponding vote is tallied.

Φ_{eli} : Ballots correspond to eligible credentials.

Φ_{cl} : Individual verification by distinct voters implies distinct public credentials.

$\Phi^{\diamond}_{\mathsf{res}}$: Tallied ballots for **honest voters** cannot come from the **adversary**.

End-to-end Election Verifiability

$$\Phi^{\diamond}_{\mathsf{E2E}} = \Phi_{\mathsf{iv}} \;\wedge\; \Phi_{\mathsf{eli}} \;\wedge\; \Phi_{\mathsf{cl}} \;\wedge\; \Phi^{\diamond}_{\mathsf{res}}$$

Φ_{iv} : Individual verification implies the corresponding vote is tallied.

Φ_{eli} : Ballots correspond to eligible credentials.

Φ_{cl} : Individual verification by distinct voters implies distinct public credentials.

$\Phi^{\diamond}_{\mathsf{res}}$: Tallied ballots for **honest voters** cannot come from the **adversary**.

- $\Phi^{\bullet}_{\mathsf{res}}$: honest voters $\equiv$ not corrupted
- $\Phi^{\circ}_{\mathsf{res}}$: honest voters $\equiv$ (not corrupted) $\wedge$ (verify their ballot)

Election Verifiability Definition

$$\Phi^{\diamond}_{E2E} = \Phi_{iv} \ \wedge \ \Phi_{eli} \ \wedge \ \Phi_{cl} \ \wedge \ \Phi^{\diamond}_{res}$$

Φ_{iv} : Individual verification implies the corresponding vote is tallied:
$\mathsf{Verified}(id, cr, v) \ \wedge \ \Omega(id, cr, v) \ \wedge \ \mathsf{BBtally}(cr, b) \implies v = \mathsf{open}(b)$
$\Omega(id, cr, v)$: captures revoting policy.

Φ_{eli} : Ballots correspond to eligible credentials:
$\mathsf{Verified}(id, cr, v) \ \vee \ \mathsf{BBtally}(cr, b) \implies \mathsf{BBreg}(cr)$

Φ_{cl} : Individual verification by distinct voters implies distinct public credentials:
$\mathsf{Verified}(id, cr, v) \ \wedge \ \mathsf{Verified}(id', cr, v') \implies id = id'$

$\Phi^{\diamond}_{res}$: Tallied ballots for **honest voters** cannot come from the **adversary**:
$\mathsf{BBtally}(cr, b) \implies (\mathsf{Vote}(id, cr, v) \ \wedge \ v = \mathsf{open}(b)) \ \vee \ \Phi^{\diamond}_{adv}(cr)$

Election Verifiability Definition

$$\Phi^{\diamond}_{E2E} = \Phi_{iv} \;\wedge\; \Phi_{eli} \;\wedge\; \Phi_{cl} \;\wedge\; \Phi^{\diamond}_{res}$$

Φ_{iv} : Individual verification implies the corresponding vote is tallied:
$\mathsf{Verified}(id, cr, v) \wedge \Omega(id, cr, v) \wedge \mathsf{BBtally}(cr, b) \implies v = \mathsf{open}(b)$
$\Omega(id, cr, v)$: captures revoting policy.

Φ_{eli} : Ballots correspond to eligible credentials:
$\mathsf{Verified}(id, cr, v) \vee \mathsf{BBtally}(cr, b) \implies \mathsf{BBreg}(cr)$

Φ_{cl} : Individual verification by distinct voters implies distinct public credentials:
$\mathsf{Verified}(id, cr, v) \wedge \mathsf{Verified}(id', cr, v') \implies id = id'$

$\Phi^{\diamond}_{res}$: Tallied ballots for **honest voters** cannot come from the **adversary**:
$\mathsf{BBtally}(cr, b) \implies (\mathsf{Vote}(id, cr, v) \wedge v = \mathsf{open}(b)) \vee \Phi^{\diamond}_{adv}(cr)$

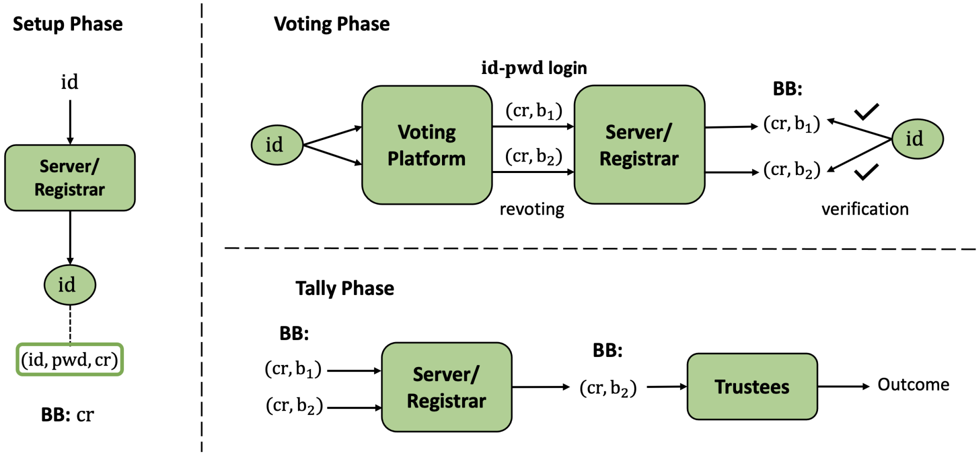
Protocol Specification

An e-voting protocol specification $\mathcal{S} = (\mathcal{P}, \mathcal{V}, \mathcal{A})$, where

- $\mathcal{P}$: the voting protocol procedures,
- $\mathcal{V}$: the individual verification procedures,
- $\mathcal{A}$: the corruption abilities of adversary.

$\mathcal{S}$ satisfies symbolic **election verifiability** if and only if $\mathcal{S} \models \Phi^{\diamond}_{\mathsf{E2E}}$.

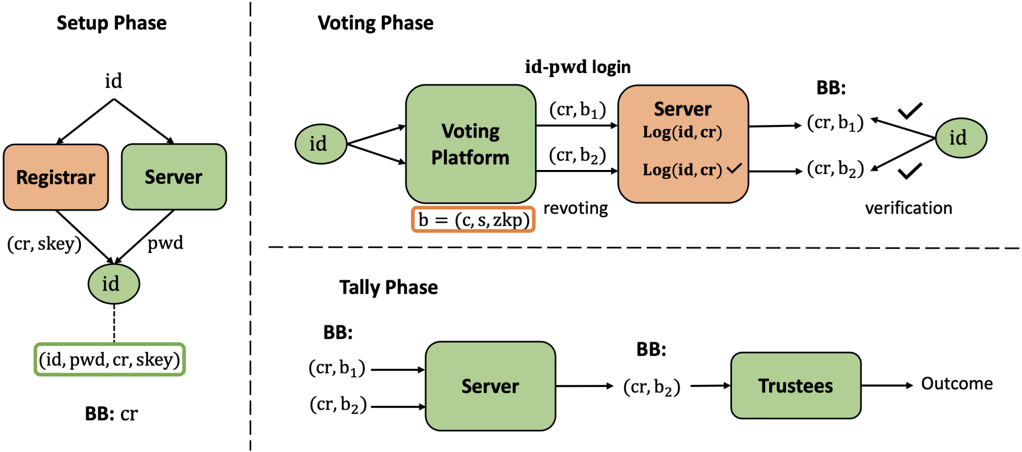
Helios


Setup Phase
id
Server/ Registrar
id
(id, pwd, cr)
BB: cr
Voting Phase
id-pwd login
id
Voting Platform
(cr, b1)
(cr, b2)
revoting
Server/ Registrar
BB:
(cr, b1)
(cr, b2)
id
verification
Tally Phase
BB:
(cr, b1)
(cr, b2)
Server/ Registrar
BB:
(cr, b2)
Trustees
Outcome

Helios and Belenios

- Main **limitation** of Helios:
  - Corrupted server can stuff ballots.
- Main **addition** of Belenios:
  - Belenios has additional signature mechanism.
  - Registrar generates a signature key pair for each eligible voter.
  - Trust is distributed between registrar and voting server.

Belenios



Setup Phase
id
Registrar
Server
(cr, skey)
pwd
id
(id, pwd, cr, skey)
BB: cr
Voting Phase
id-pwd login
id
Voting Platform
(cr, b1)
(cr, b2)
Server
Log(id, cr)
Log(id, cr) ✓
b = (c, s, zkp)
revoting
BB:
(cr, b1)
(cr, b2)
id
verification
Tally Phase
BB:
(cr, b1)
(cr, b2)
Server
BB:
(cr, b2)
Trustees
Outcome

Individual Verification and Adversary

Individual Verifications

$\mathcal{V}_1$	**last** ballot **anytime**
$\mathcal{V}_2$	**all** ballots **anytime**
$\mathcal{V}_3$	**last** ballot **in tally phase**
$\mathcal{V}_4$	**empty** ballot **in tally phase**

Corruption Scenarios

$\mathcal{A}_1$	trustees and voters
$\mathcal{A}_2$	trustees, voters and **server**
$\mathcal{A}_3$	trustees, voters and **registrar**
$\mathcal{A}_4$	trustees, voters, **server** and **registrar**
$\mathcal{A}_5$	trustees, voters, **server**, **registrar** and **voting platform**

Verification Results obtained by Tamarin

Helios

$\mathcal{V}_i / \mathcal{A}_j$	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_4$	$\mathcal{A}_5$
$\mathcal{V}_1$	✓	✗	✗	✗
$\mathcal{V}_3$	✓	✓	✓	✗

Belenios

$\mathcal{V}_i / \mathcal{A}_j$	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_5$
$\mathcal{V}_1$	✗	✗	✗	✗	?
$\mathcal{V}_3$	✓	✓	✓	✓	✗

Individual Verifications

$\mathcal{V}_1$	**last** ballot **anytime**
$\mathcal{V}_2$	**all** ballots **anytime**
$\mathcal{V}_3$	**last** ballot **in tally phase**
$\mathcal{V}_4$	**empty** ballot **in tally phase**

Verification Results obtained by Tamarin

Helios

$\mathcal{V}_i/\mathcal{A}_j$	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_4$	$\mathcal{A}_5$
$\mathcal{V}_1$	✓	✗	✗	✗
$\mathcal{V}_3$	✓	✓	(✓)	✗

Belenios

$\mathcal{V}_i/\mathcal{A}_j$	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	$\mathcal{A}_5$
$\mathcal{V}_1$	✗	✗	✗	✗	?*
$\mathcal{V}_3$	✓	✓	✓	(✓)	✗

Individual Verifications

$\mathcal{V}_1$	**last** ballot **anytime**
$\mathcal{V}_2$	**all** ballots **anytime**
$\mathcal{V}_3$	**last** ballot **in tally phase**
$\mathcal{V}_4$	**empty** ballot **in tally phase**

Clash Attacks against Helios

Clash attacks proposed by:

- R. Küsters, T. Truderung, and A. Vogt, "Clash attacks on the verifiability of e-voting systems," in IEEE S&P'12.

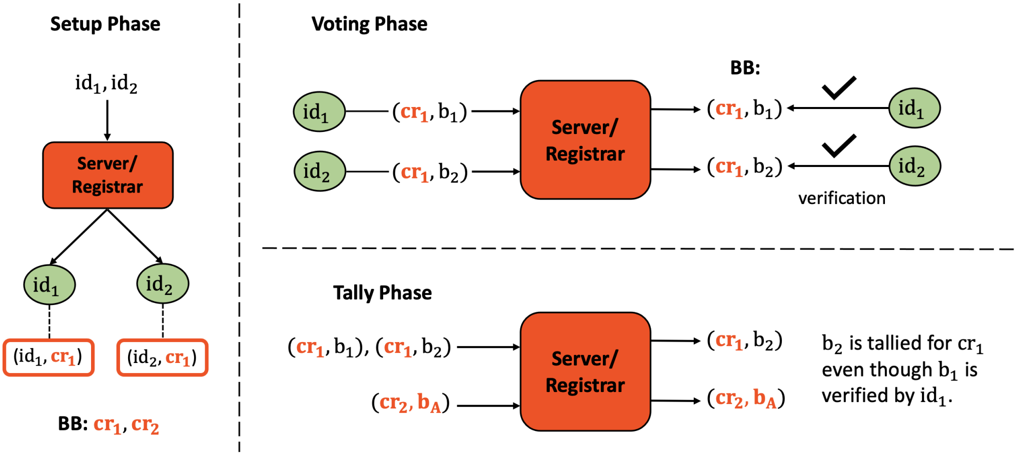
is mounted by clash on public credentials when **the server** and **voting platforms** are corrupted.

We show that it is not necessary to corrupt voting platforms, if

- **revoting** is allowed,
- voters **verify** their ballots **anytime**.

We discover this based on the property Φ_{cl} with Tamarin.

Clash Attack by corrupted server against Helios



Attack on Individual Verifiability against Belenios

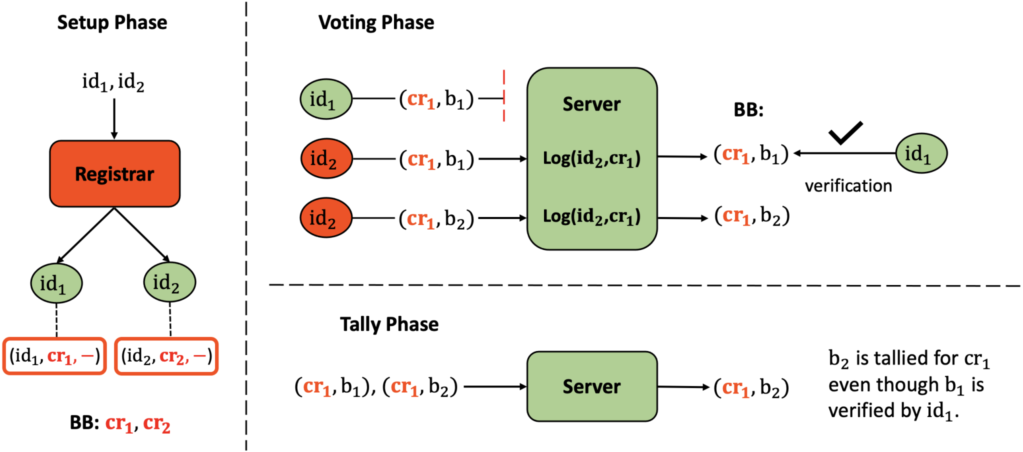
Belenios should be secure against ballot stuffing with **corrupted registrar**:

- The voting server verifies passwords, and checks the consistency on the logs.

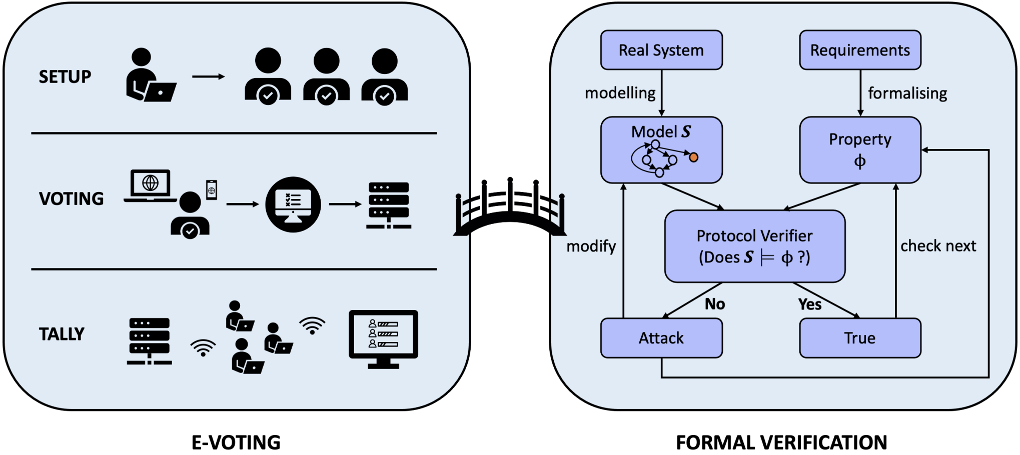
However, we find clash attack (Φ_{cl}) and attacks against individual verifiability (Φ_{iv}) and result integrity ($\Phi^{\bullet}_{res}$, Φ°_{res}):

Φ_{iv} : If voters verify the last ballot they cast, it should be counted.

Attack on Individual Verifiability against Belenios



Bridging the gap between practice and theory



Thank you for listening!