A comprehensive analysis of Belenios

Alexandre Debant, *and Véronique Cortier*

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

GT-MFS

Fréjus, March 21st 2022

E-voting today

Recherche

Les Echos

CONNEXION S'ABONNER

À la une Idées Économie Politique Élections Monde Entreprises Tech-Médias Start-up Bourse Finance - Marchés Ré

Hauts-de-Seine : Neuilly-sur-Seine met en place un système de vote électronique

La mairie de Neuilly-sur-Seine va tester un système de vote électronique pour permettre aux habitants d'arbitrer des décisions local
imaginée par l'association fran

LE TEMPS

SE CONNECTER SEF

RUBRIQUES EN CONTINU BLOGS VIDÉOS CHAPPATTE MULTIMÉDIA EPAPER/PDF

Accueil › Suisse › Le vote électronique fera son retour en 2022

DROITS POPULAIRES ABONNÉ

Le vote électronique fera son retour en 2022

Après la découverte de failles en 2019, tous les projets de
pendus. La Poste a cependant
éveloppe à Neuchâtel un système
nettra à des hackers

Consulter le journal

Le Monde

Se connecter S'abonner

ACTUALITÉS ÉCONOMIE VIDÉOS OPINIONS CULTURE M LE MAG SERVICES

LES DÉCODEURS · RÉGIONALES & DÉPARTEMENTALES

Partage

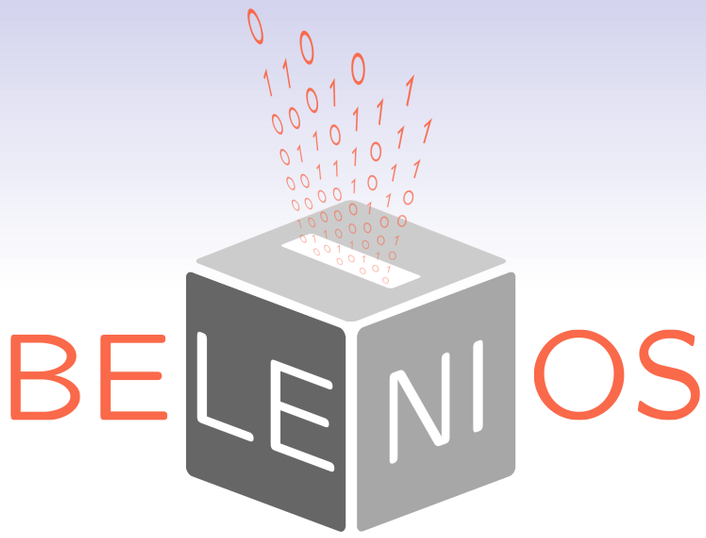
Elections régionales 2021 : le vote électronique, remède à l'abstention ?

Après un premier tour marqué par une abstention historique, des membres de la majorité ont appelé à moderniser les scrutins, pour voter plus facilement, et donc de mobiliser davantage les électeurs.

Par Assma Maad et Clément Perruche

Publié le 25 juin 2021 à 18h40 - Mis à jour le 26 juin 2021 à 16h42 · Lecture 7 min.

Belenios



General information

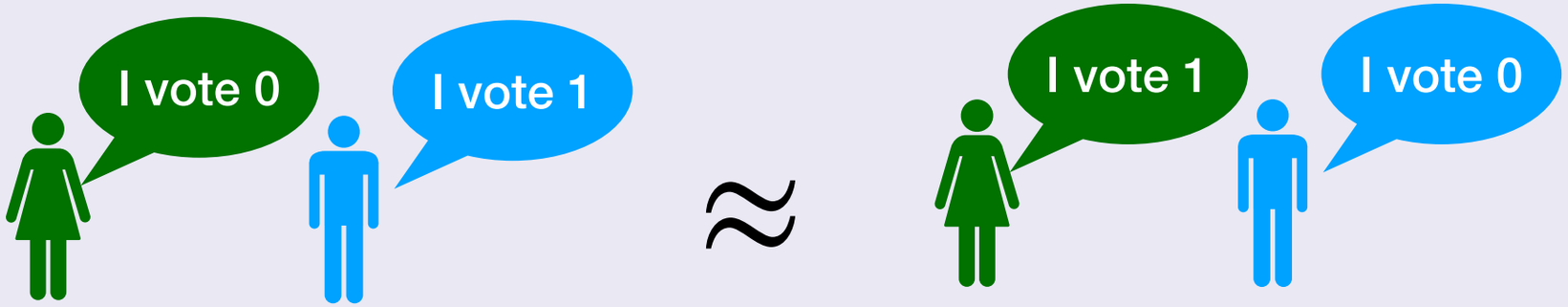
- developers: Véronique Cortier, Pierrick Gaudry, Stéphane Glondu
- context: developed for associative or professional elections
- +2000 elections in 2021, +110 000 ballots
- multi-languages platform: French, English, Spanish…

Technical details

- re-vote
- homomorphic tally and/or mixnets
- threshold decryption (k ou of n decryption trustees)
- weighted votes
- …

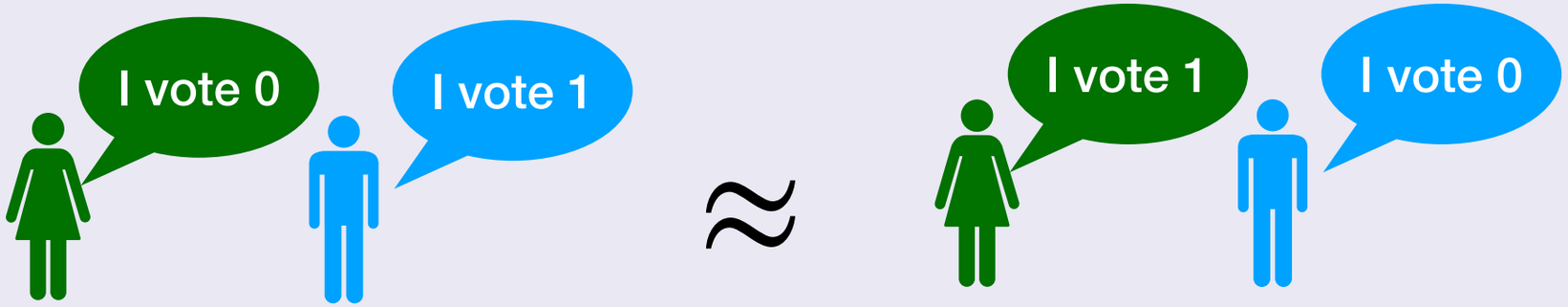
Security properties

Vote secrecy - no one is able to learn who I voted for!



Security properties

Vote secrecy - no one is able to learn who I voted for!



Verifiability - no one is able to modify the result of an election!

- Eligibility: all the counted ballots belong to legitimate voters
- Individual verifiability: if I see my last ballot on the bulletin board, it will be counted
- Universal verifiability: the result corresponds to the content of the ballot box

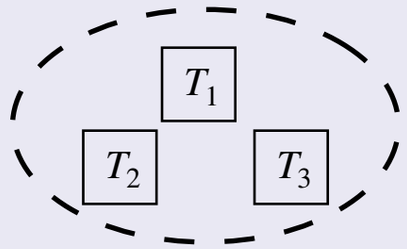
A complex environment…

A lot of participants… with different roles…

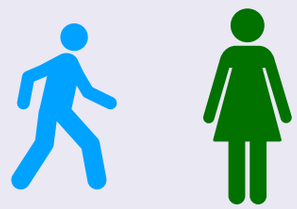
Registrar

Public bulletin board

Voting server



Decryption trustees



Voters

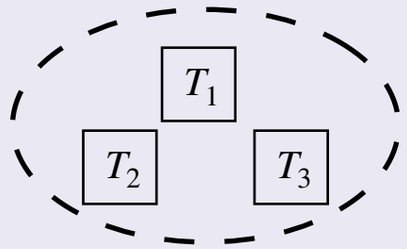
A complex environment…

A lot of participants… with different roles…

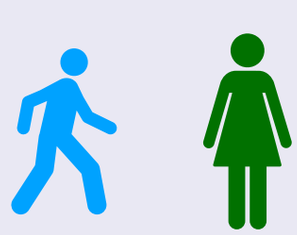
Registrar

Public bulletin board

Voting server



Decryption trustees

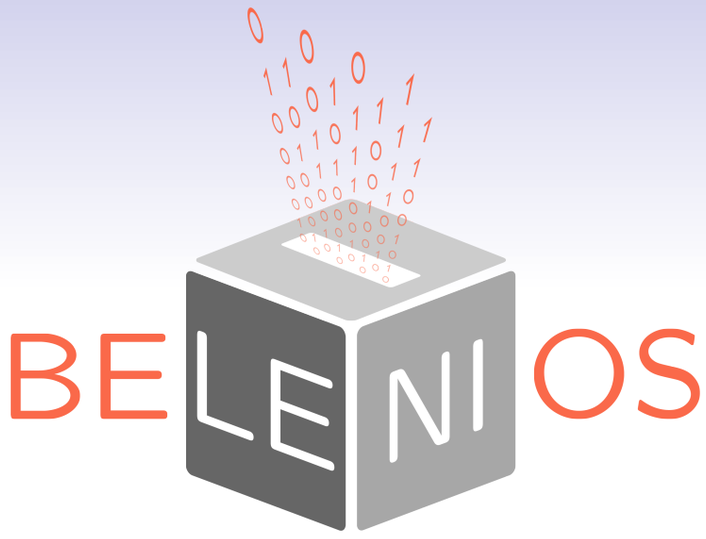


Voters

And complex scenarios…

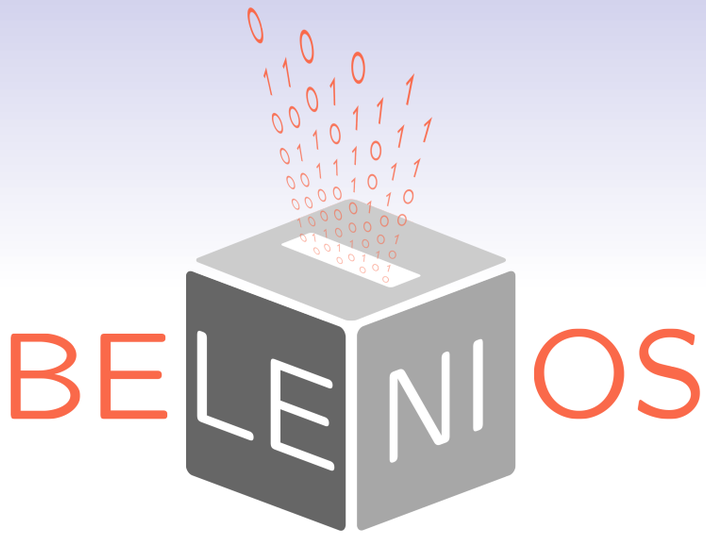
- Re-vote
- Two-round elections
- Multiple ballot-boxes (*e.g.*, one per village/city)

Security claims



- verifiability as soon as the registrar **or** the voting server is honest

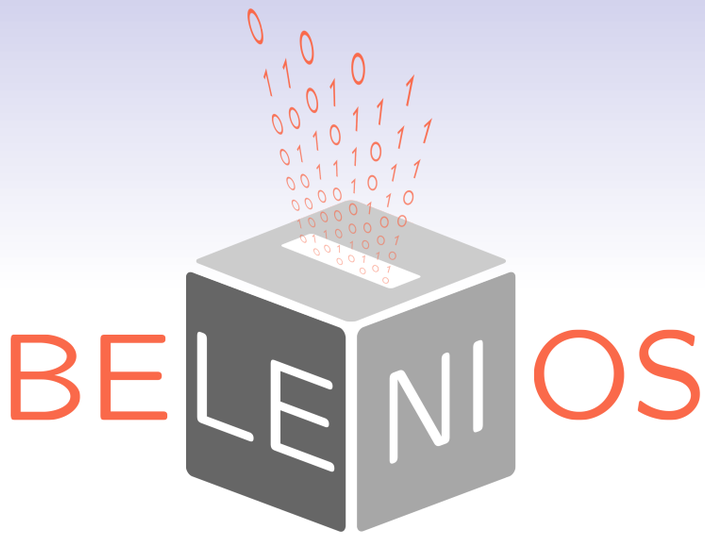
Security claims



- verifiability as soon as the registrar **or** the voting server is honest

- vote secrecy as soon as **k out of n** decryption trustees are honest

Security claims



- verifiability as soon as the registrar **or** the voting server is honest

ballot re-ordering attacks if re-vote is allowed

[Baloglu *et al.*- CSF'21]

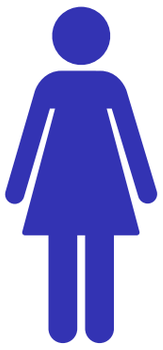
- vote secrecy as soon as **k out of n** decryption trustees are honest

Ballot re-ordering attack

[Baloglu *et. al.*- CSF'21]

Individual verifiability - if I see my last ballot on the bulletin board, it will be counted

Attack scenario

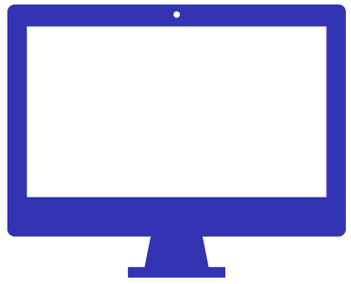


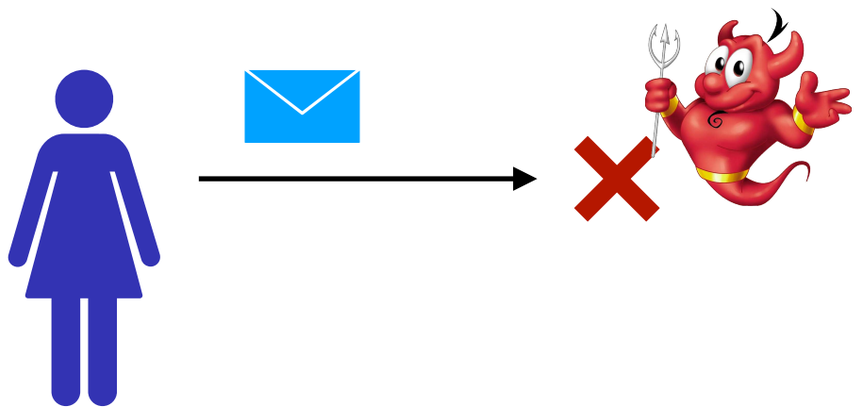
Ballot re-ordering attack

[Baloglu *et. al.*- CSF'21]

Individual verifiability - if I see my last ballot on the bulletin board, it will be counted

Attack scenario



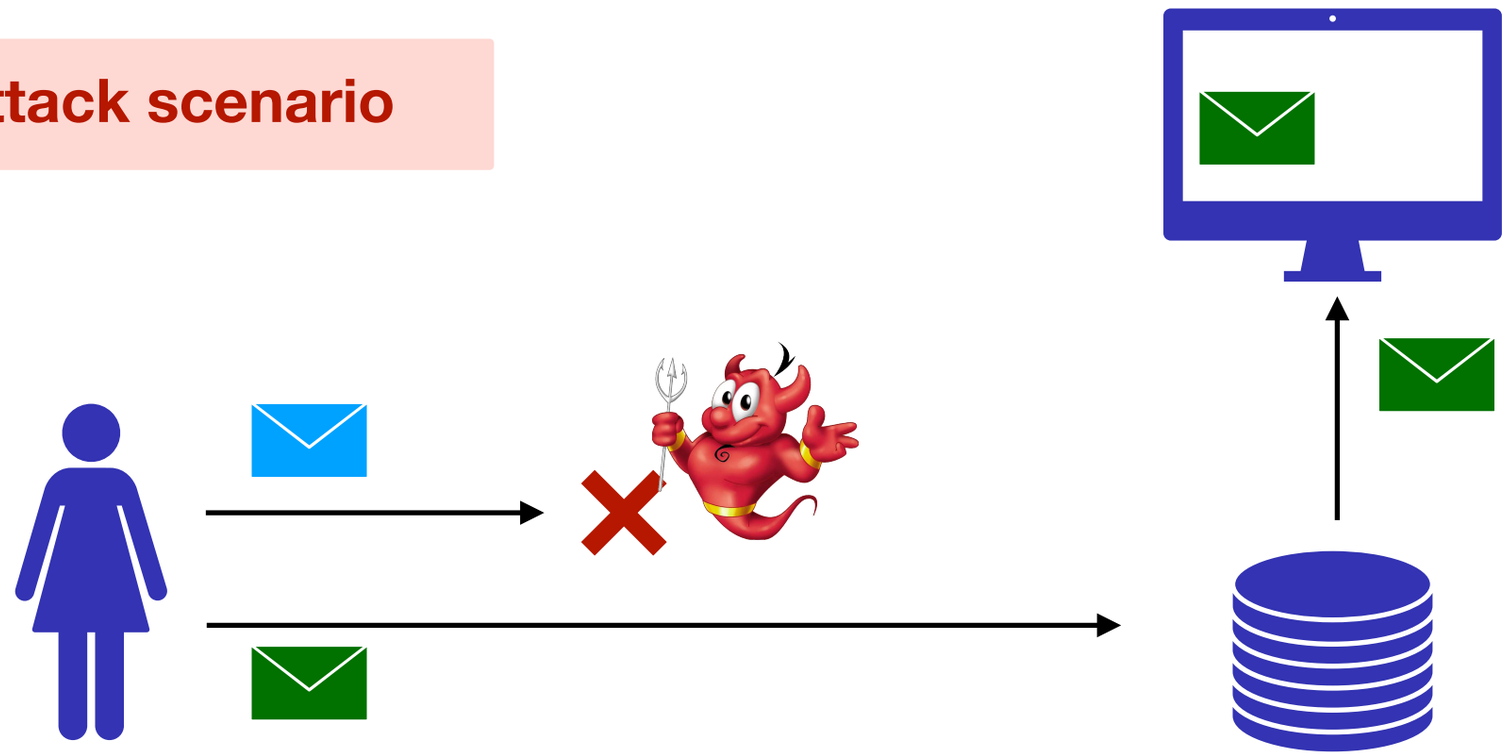


Ballot re-ordering attack

[Baloglu *et. al.*- CSF'21]

Individual verifiability - if I see my last ballot on the bulletin board, it will be counted

Attack scenario

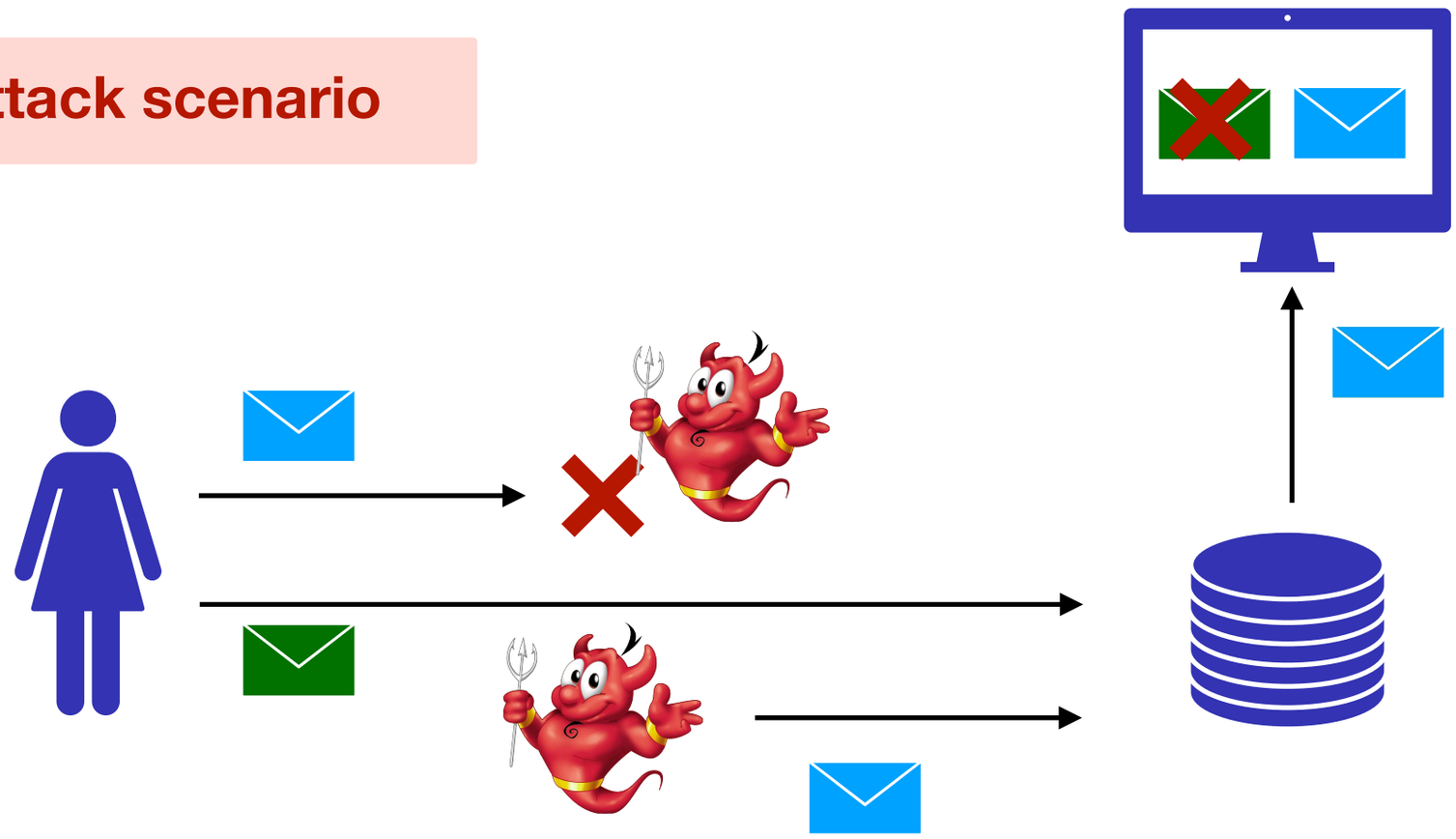


Ballot re-ordering attack

[Baloglu *et. al.*- CSF'21]

Individual verifiability - if I see my last ballot on the bulletin board, it will be counted

Attack scenario

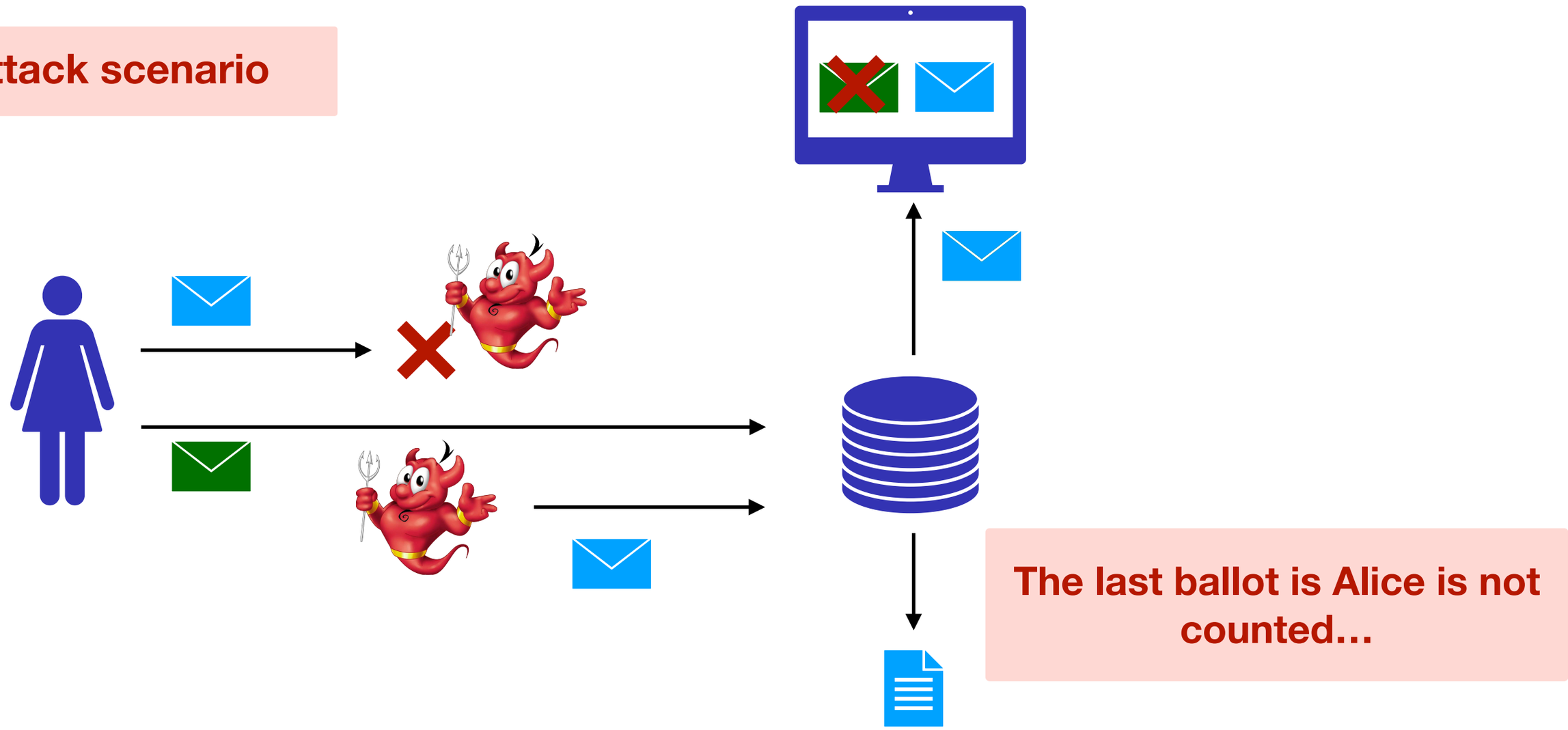


Ballot re-ordering attack

[Baloglu *et. al.*- CSF'21]

Individual verifiability - if I see my last ballot on the bulletin board, it will be counted

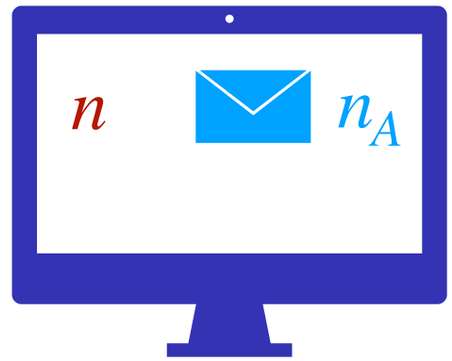
Attack scenario



The last ballot is Alice is not counted…

Fixing the attack with counters

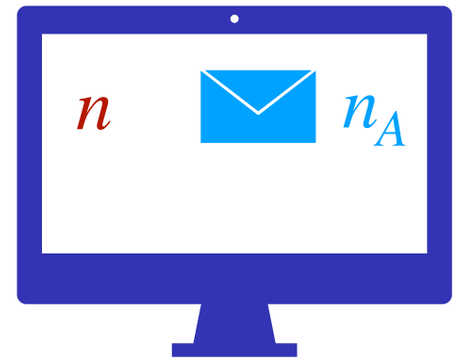
1. The bulletin board is initialized with a counter set to 0

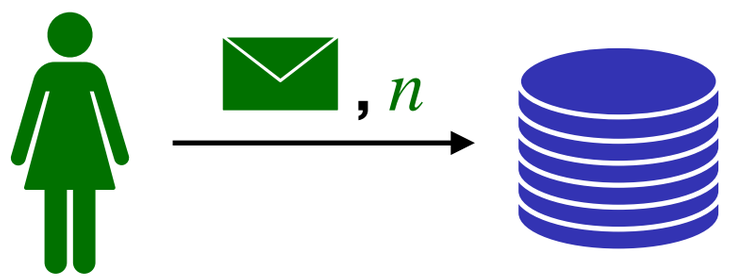


Fixing the attack with counters

1. The bulletin board is initialized with a counter set to 0

2. Alice adds the counter in her ballot



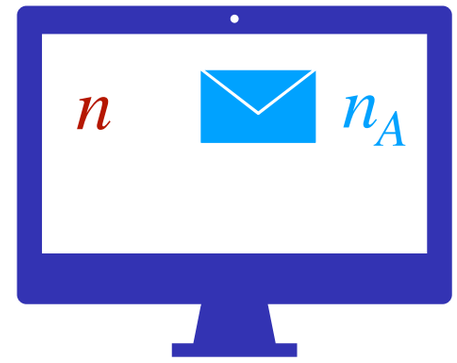


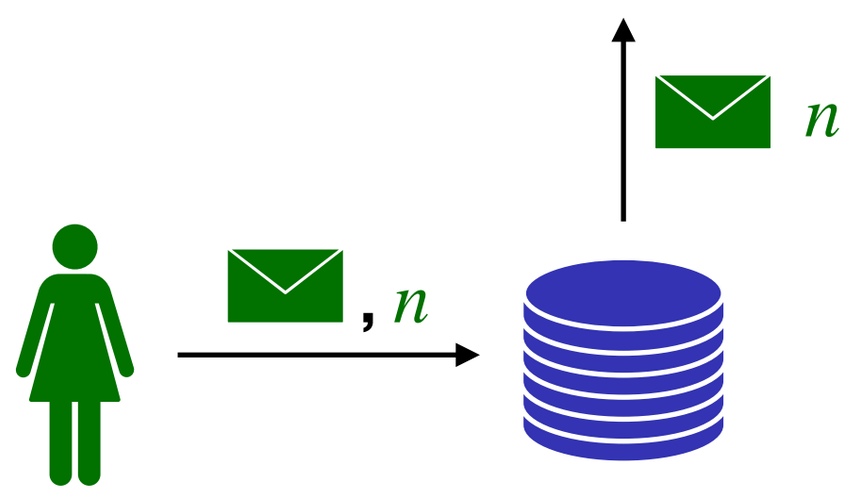
Fixing the attack with counters

1. The bulletin board is initialized with a counter set to 0

2. Alice adds the counter in her ballot

3. The server accepts ballots with a greater counter (compared to the last Alice's accepted ballot) and increments it by 1



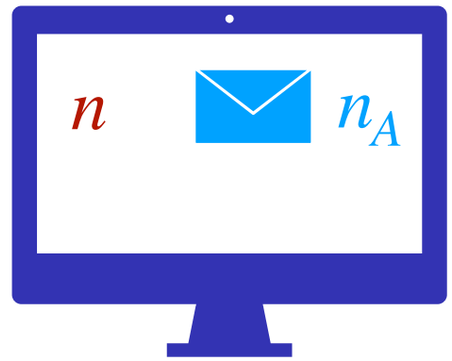


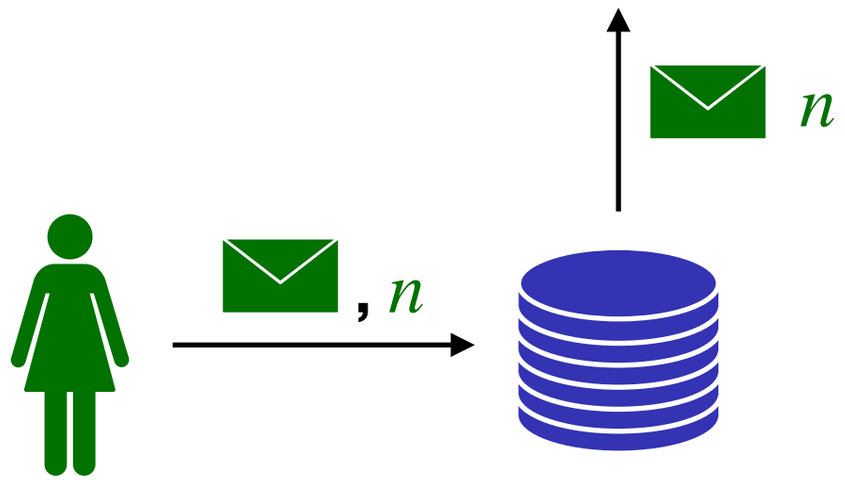
Fixing the attack with counters

1. The bulletin board is initialized with a counter set to 0

2. Alice adds the counter in her ballot

3. The server accepts ballots with a greater counter (compared to the last Alice's accepted ballot) and increments it by 1

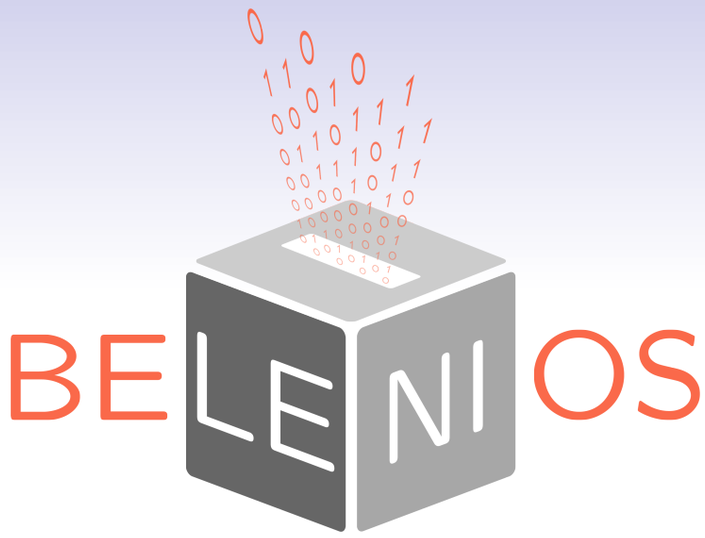




Security analysis in Proverif

- use of natural numbers natively supported by ProVerif since recently
- slightly simplify the use of counters in the model; prove the gap by hand
- rely on axioms/lemmas and `[precise]` to avoid inaccuracies of the tool

Security claims



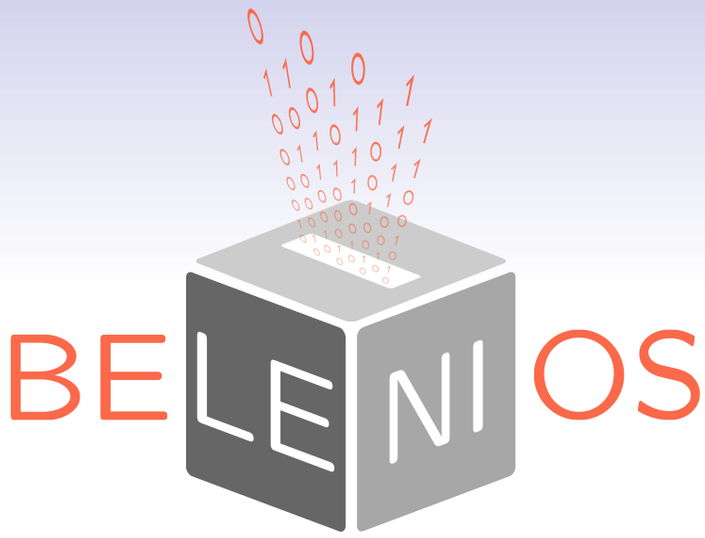
- verifiability as soon as the registrar **or** the voting server is honest

ballot re-ordering attacks if re-vote is allowed

[Baloglu *et al.*- CSF'21]

- vote secrecy as soon as **k out of n** decryption trustees are honest

Security claims



- verifiability as soon as the registrar **or** the voting server is honest

ballot re-ordering attacks if re-vote is allowed

[Baloglu *et al.*- CSF'21]

- vote secrecy as soon as **k out of n** decryption trustees are honest

multi-election attacks [NEW]

A privacy attack against Belenios

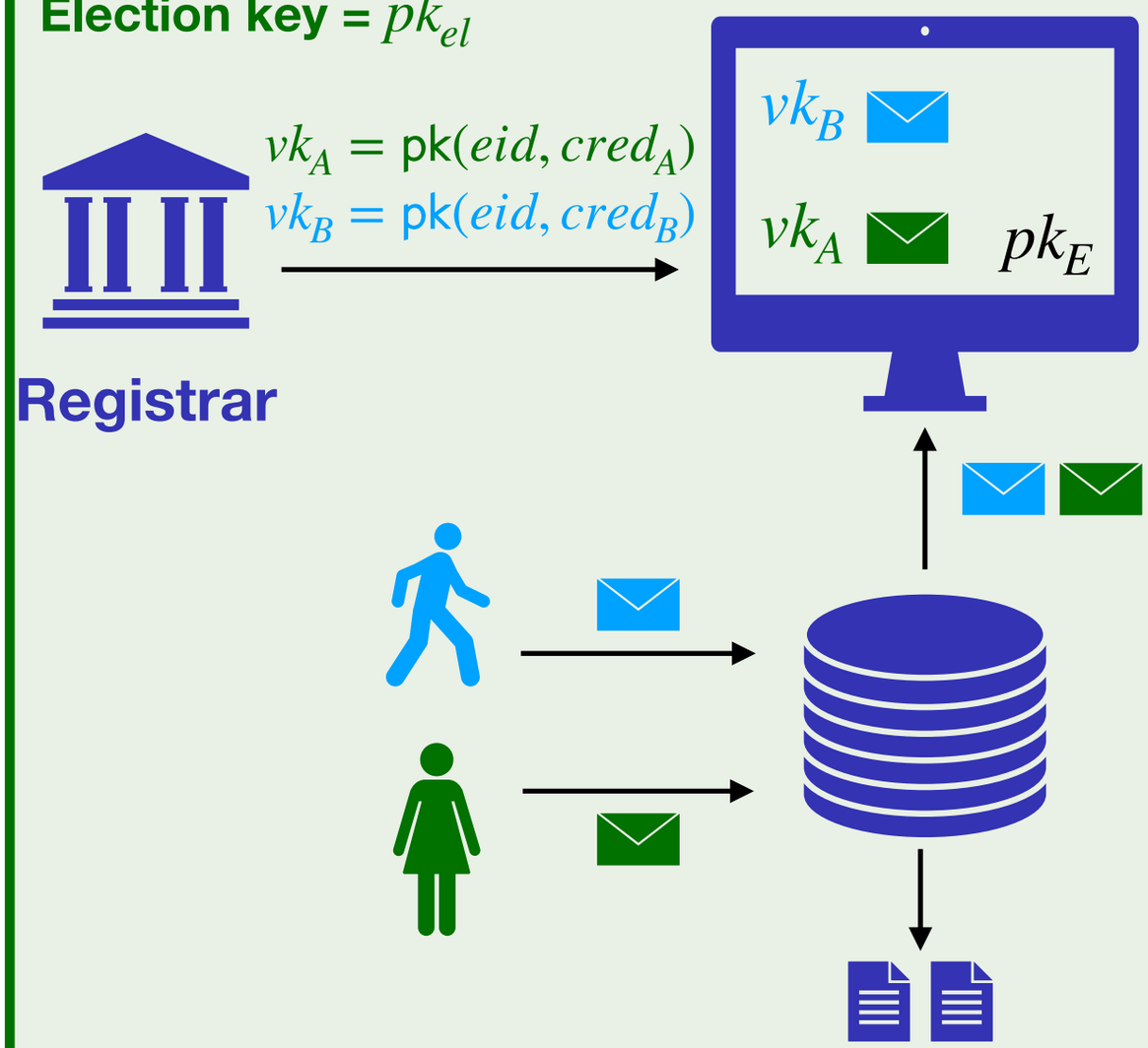
Election 1 (important election)

Election 2 (small/test election)

A privacy attack against Belenios

Election 1 (important election)

Election key = pk_{el}

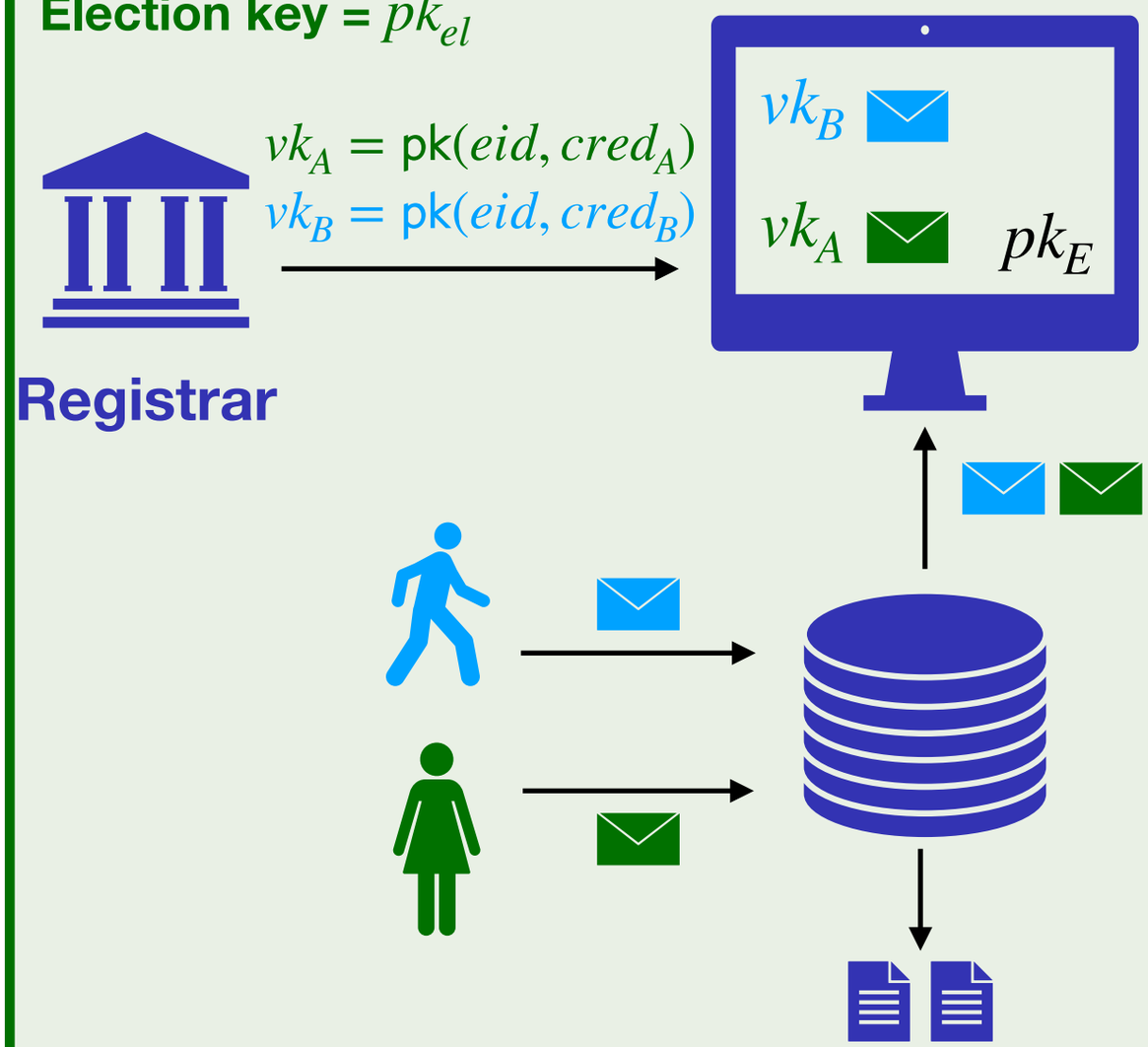


Election 2 (small/test election)

A privacy attack against Belenios

Election 1 (important election)

Election key = pk_{el}



Election 2 (small/test election)

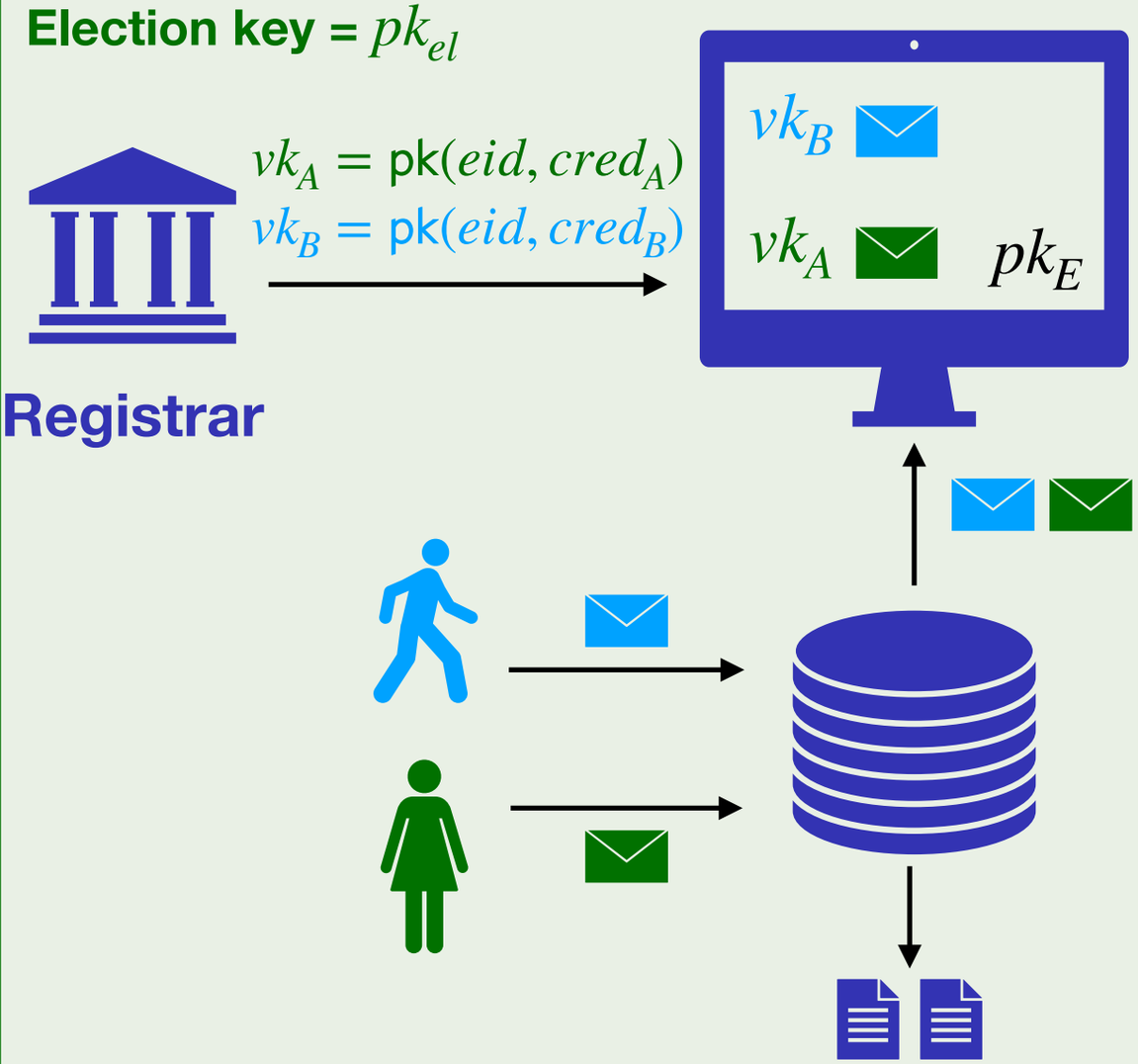
Election key = pk_{el}

The trustees can use the same keys

A privacy attack against Belenios

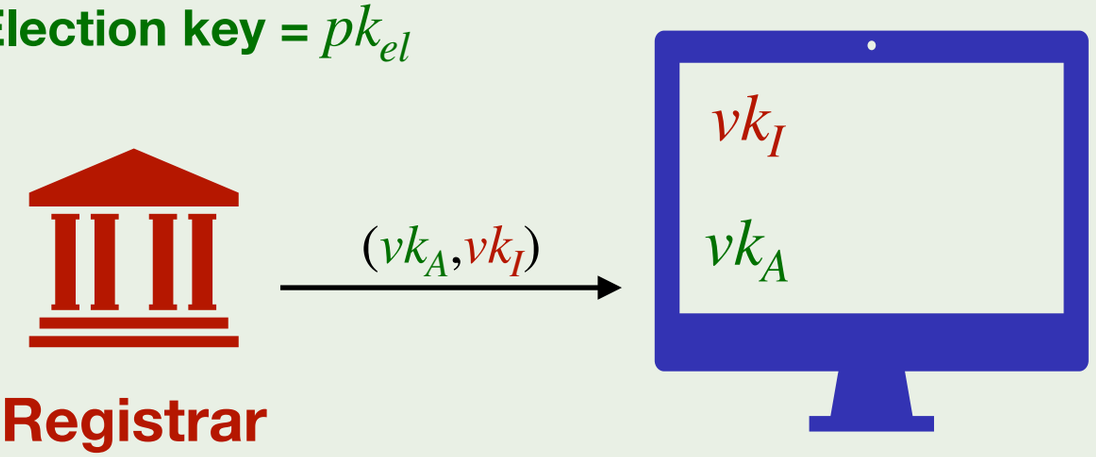
Election 1 (important election)

Election key = pk_{el}

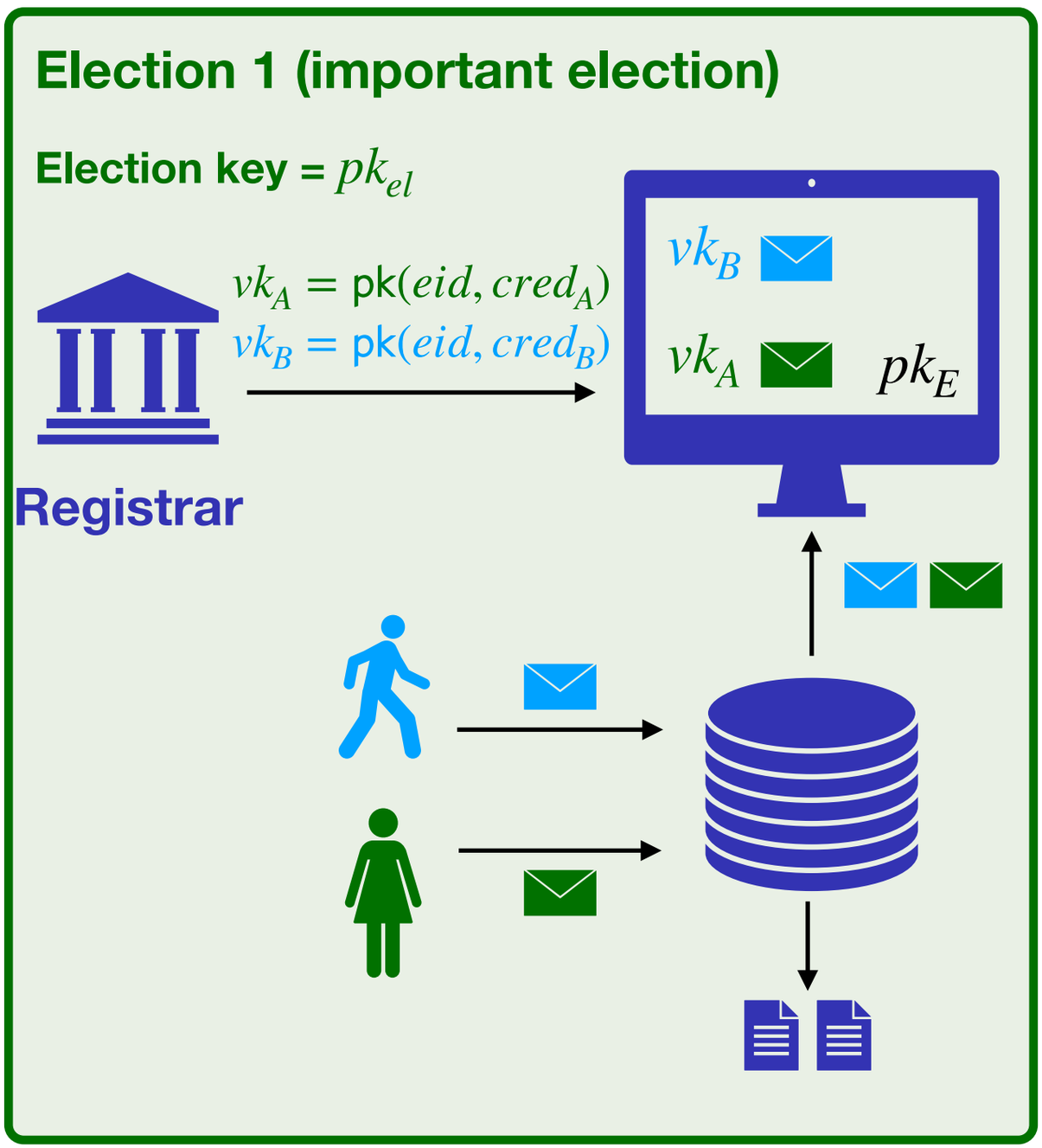


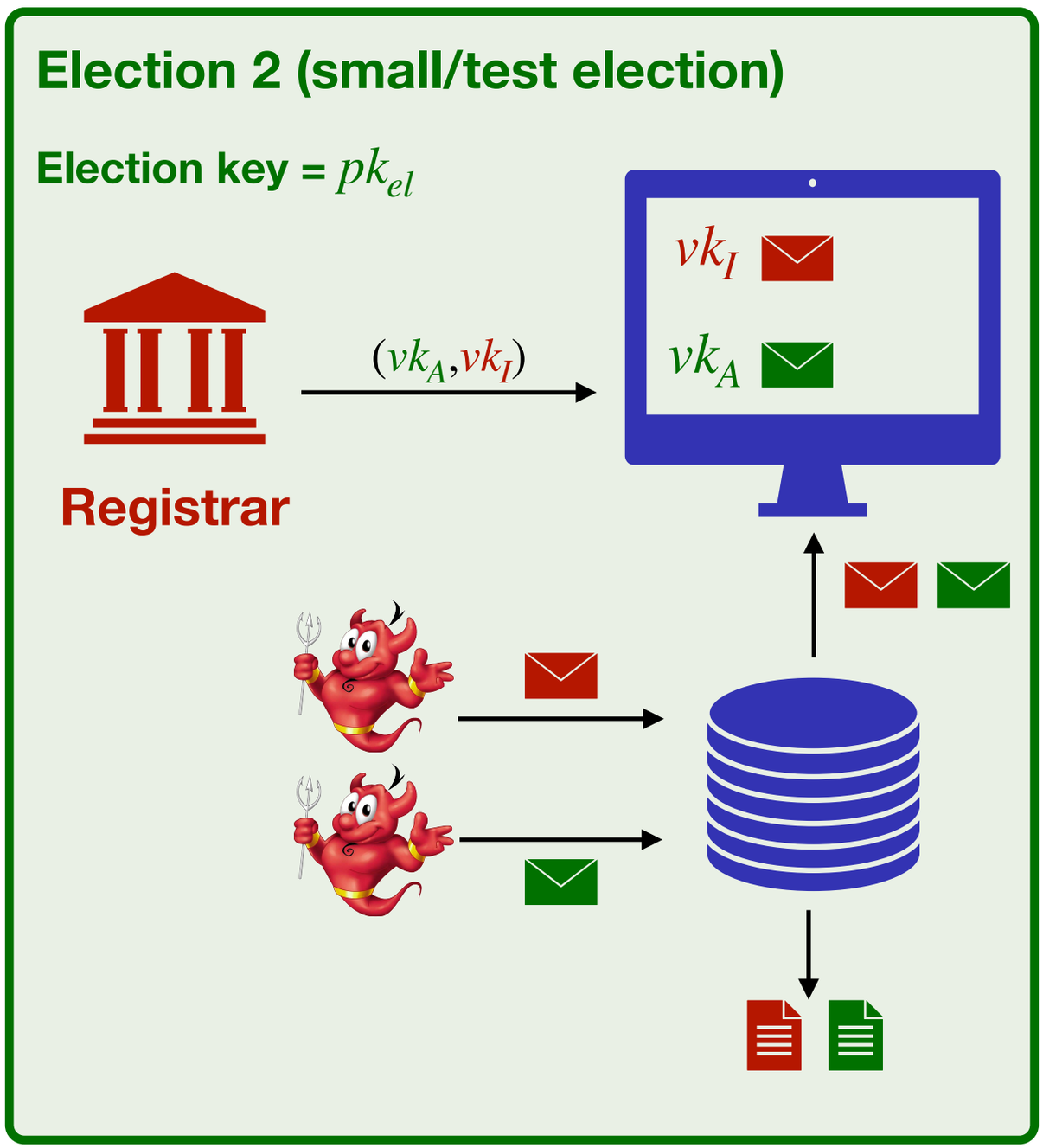
Election 2 (small/test election)

Election key = pk_{el}



A privacy attack against Belenios

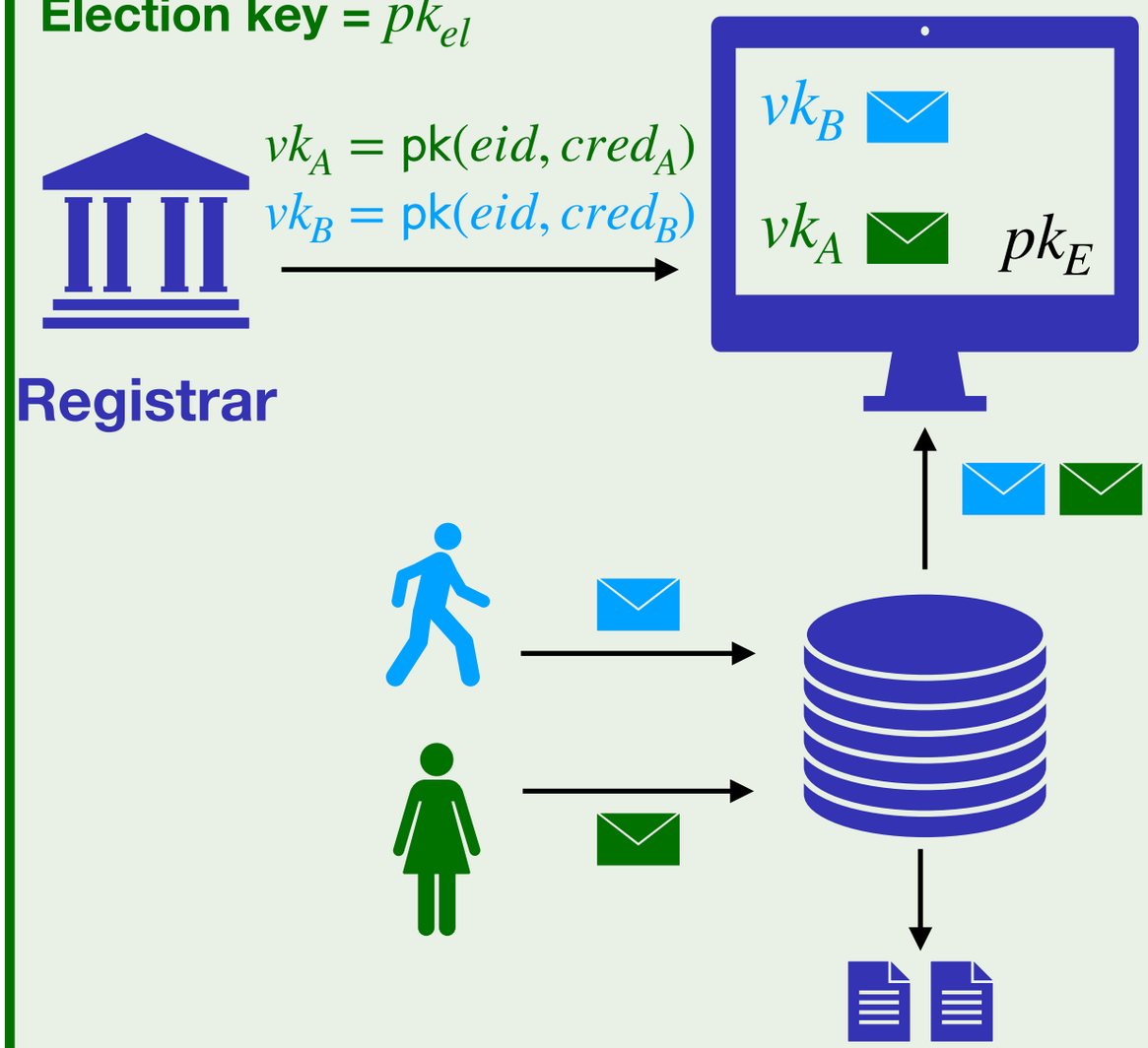




A privacy attack against Belenios

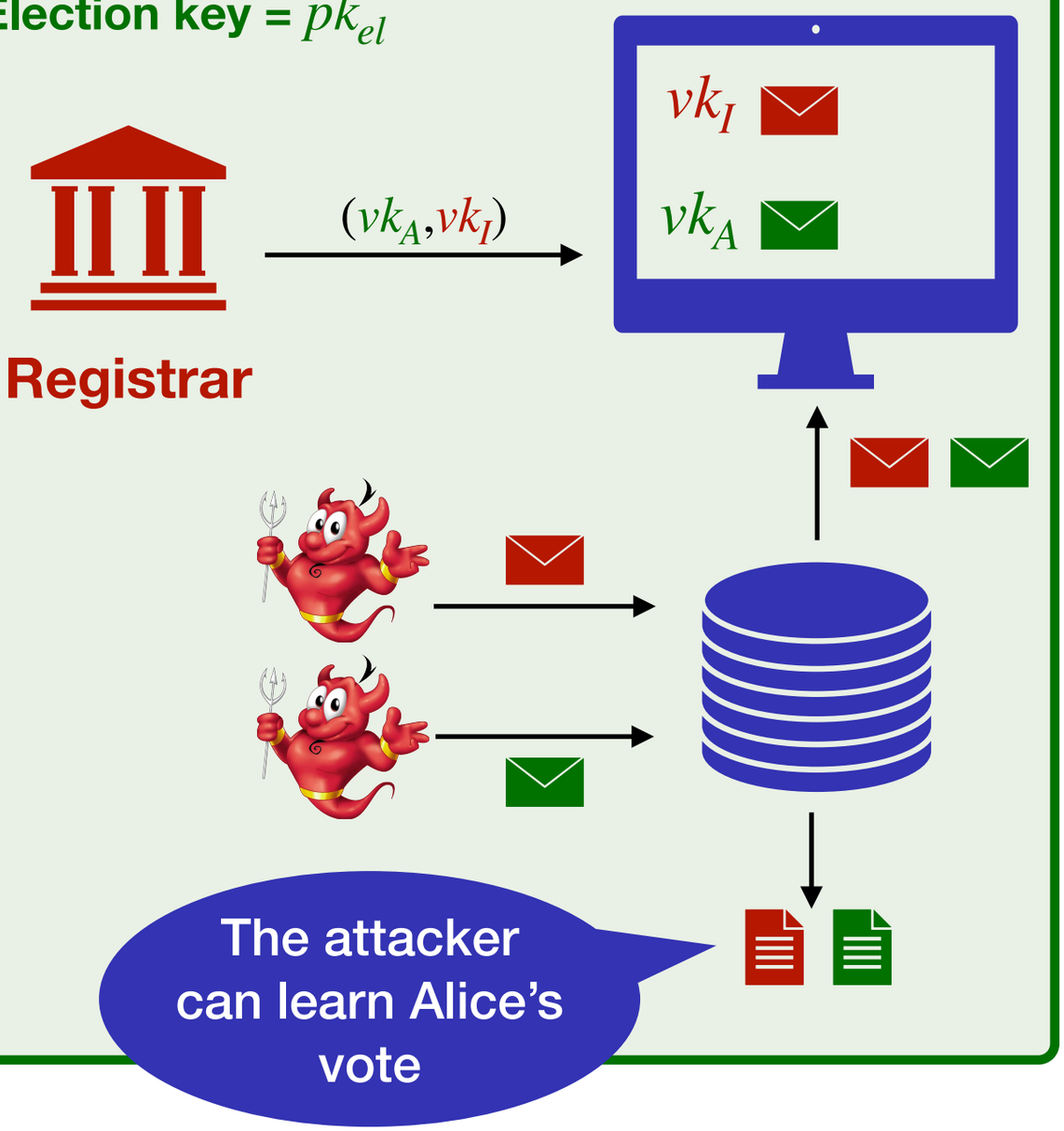
Election 1 (important election)

Election key = pk_{el}



Election 2 (small/test election)

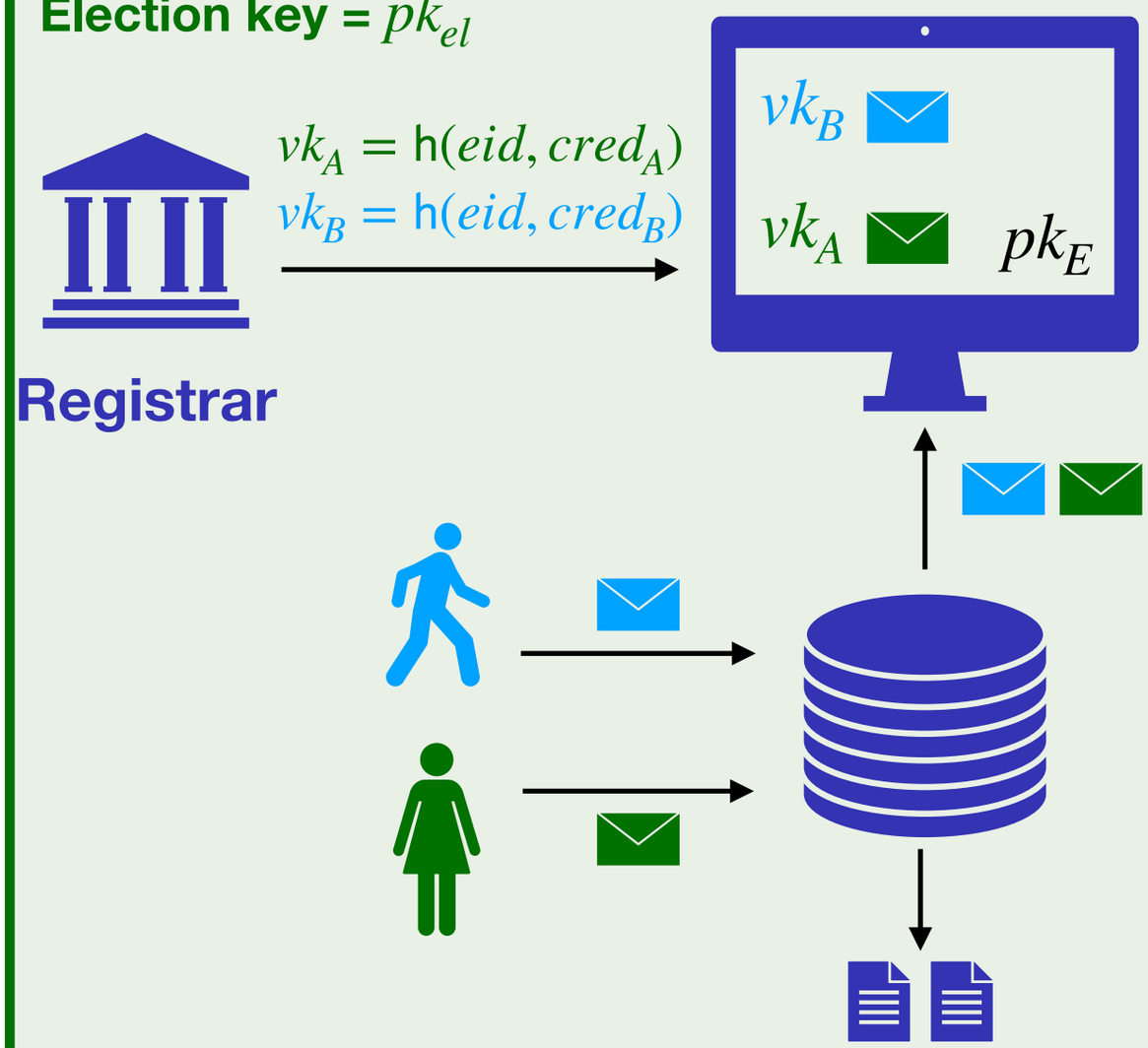
Election key = pk_{el}



A privacy attack against Belenios

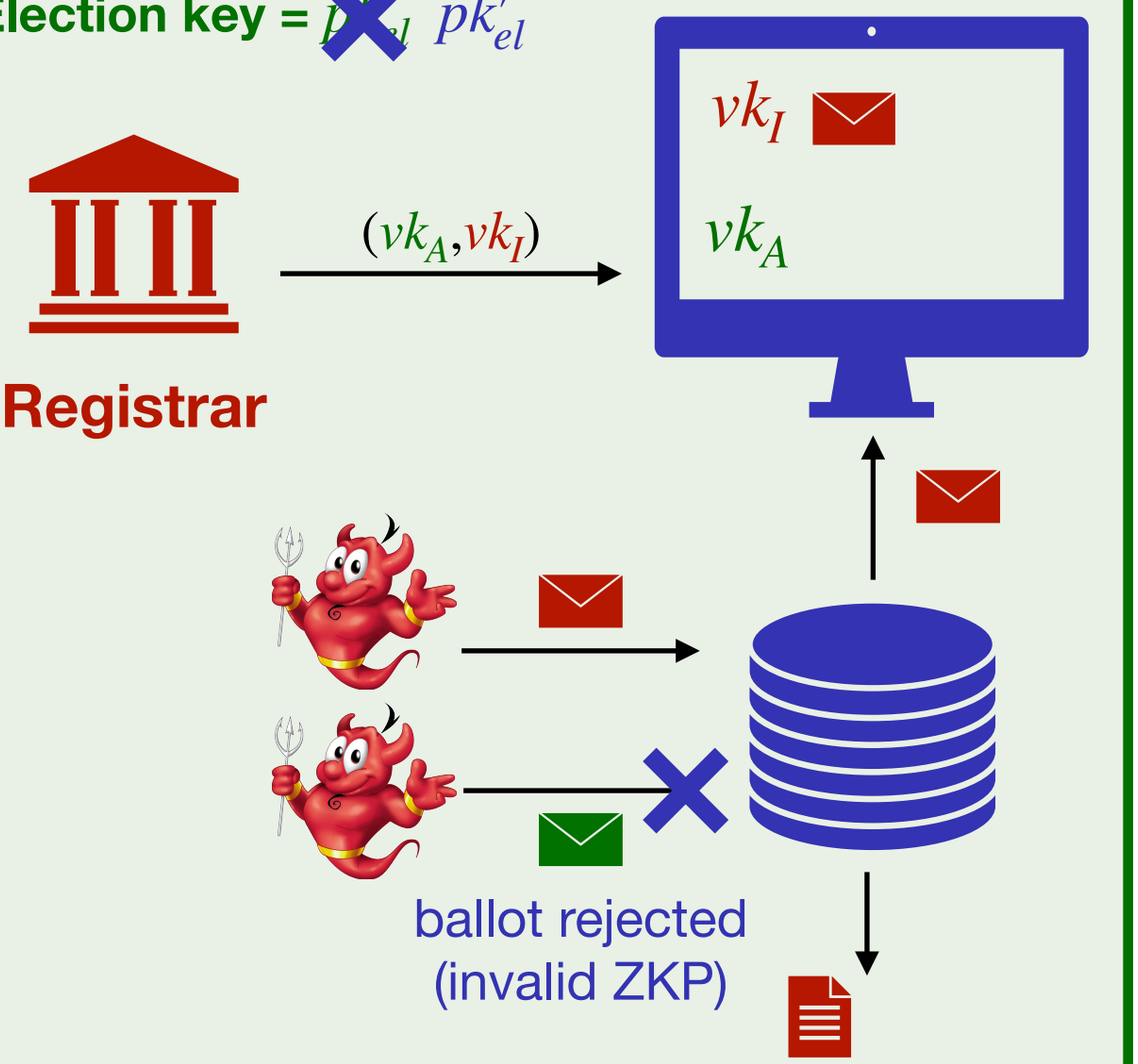
Election 1 (important election)

Election key = pk_{el}



Election 2 (small/test election)

Election key = ~~pk_{el}~~ pk'_{el}



After v1.13 with an honest server

Fix - the server acts as a decryption trustee and must refresh its key for each election

Belenios - summary

[submitted at ESORICS'22]

Contributions :

- A (partial) fix: the Voting Server acts as a Trustee for decryption!
- A comprehensive model of Belenios including multi-elections
- Security proofs in ProVerif
- Paper proofs justifying the approximations about counters

Belenios - summary

[submitted at ESORICS'22]

Contributions :

- A (partial) fix: the Voting Server acts as a Trustee for decryption!
- A comprehensive model of Belenios including multi-elections
- Security proofs in ProVerif
- Paper proofs justifying the approximations about counters

	Registrar	Server	Belenios <v1.13	Belenios + Server Trustee	Belenios + Server Trustee + counters/commit
Verifiability	Dis	Hon	❌ [CSF'21]	❌	✅
	Hon	Dis	❌ [CSF'21]	❌	✅
Privacy	Dis	Hon	❌	✅	✅
	Hon	Dis	❌	✅ / ❌	✅ / ❌

Belenios - summary

[submitted at ESORICS'22]

Contributions :

- A (partial) fix: the Voting Server acts as a Trustee for decryption!
- A comprehensive model of Belenios including multi-elections
- Security proofs in ProVerif
- Paper proofs justifying the approximations about counters

	Registrar	Server	Belenios <v1.13	Belenios + Server Trustee	Belenios + Server Trustee + counters/commit
Verifiability	Dis	Hon	❌ [CSF'21]	❌	✅
	Hon	Dis	❌ [CSF'21]	❌	✅
Privacy	Dis	Hon	❌	✅	✅
	Hon	Dis	❌	✅ / ❌	✅ / ❌

Only if all the elections are audited

Belenios - summary

[submitted at ESORICS'22]

Contributions :

- A (partial) fix: the Voting Server acts as a Trustee for decryption!
- A comprehensive model of Belenios including multi-elections
- Security proofs in ProVerif
- Paper proofs justifying the a… counters

Based on counters to avoid ballot-reordering attacks

	Registrar	Server	Belenios <v1.13	Belenios + Server Trustee	Belenios + …rver Trustee + c…nters/commit
Verifiability	Dis	Hon	❌ [CSF'21]	❌	✅
	Hon	Dis	❌ [CSF'21]	❌	✅
Privacy	Dis	Hon	❌	✅	✅
	Hon	Dis	❌	✅ / ❌	✅ / ❌

Only if all the elections are audited

Take home

Design and prove the security of an e-voting protocol is difficult…

- re-vote policies
- multiple elections
- …

Take home

Design and prove the security of an e-voting protocol is difficult…

- re-vote policies
- multiple elections
- …

The Swiss solution was also vulnerable to multi-elections attacks (will be presented at RWC'22)

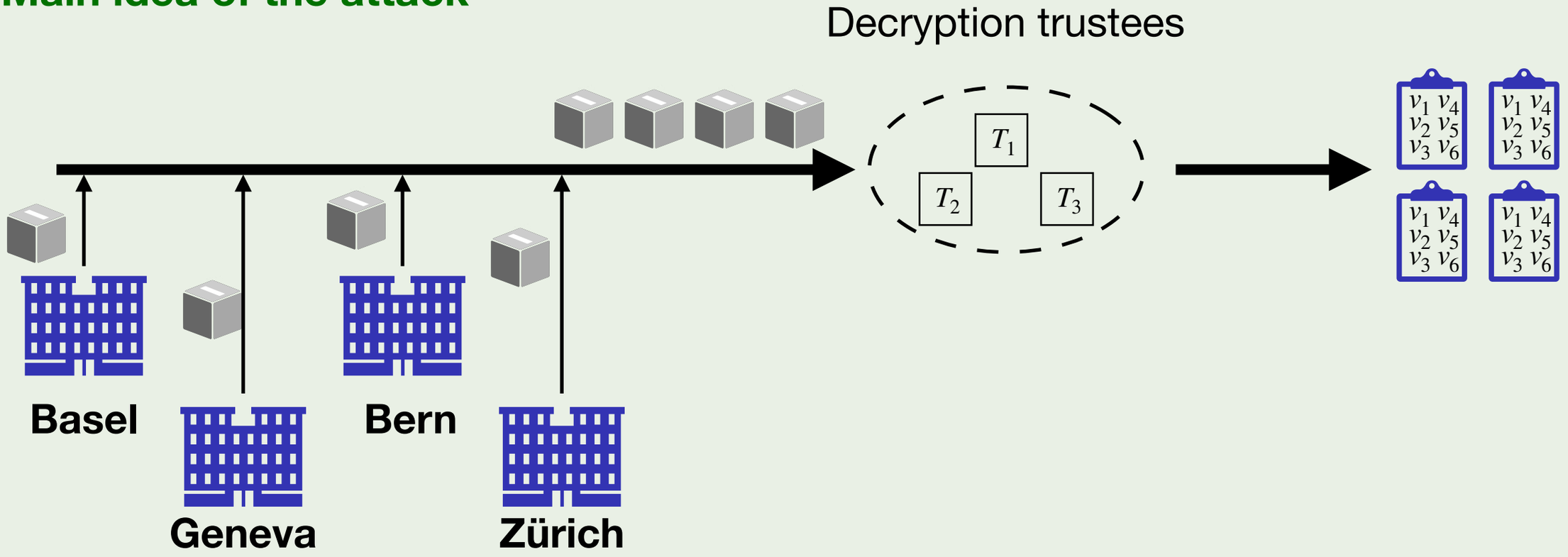
Take home

Design and prove the security of an e-voting protocol is difficult…

- re-vote policies
- multiple elections
- …

The Swiss solution was also vulnerable to multi-elections attacks (will be presented at RWC'22)

Main idea of the attack



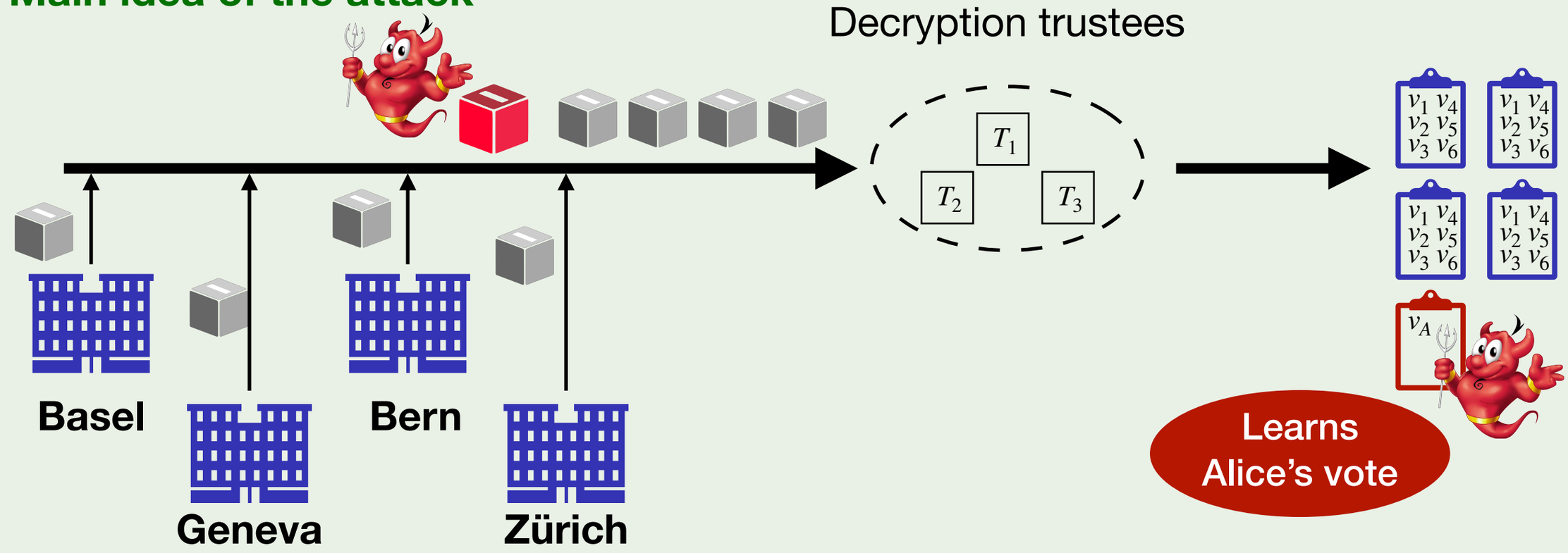
Take home

Design and prove the security of an e-voting protocol is difficult…

- re-vote policies
- multiple elections
- …

The Swiss solution was also vulnerable to multi-elections attacks (will be presented at RWC'22)

Main idea of the attack



Future works

- **Ensure cast-as-intended in Belenios**
  - ➡ model arithmetics
  - ➡ model probabilities (e.g., random audits)

- **Study accountability (what happens in case of failure?)**
  - ➡ design choice to improve Belenios
  - ➡ verify liveness properties