Some numbers...

- May 27th — June 1st: first round of the election
- June 10th — June 15th: second round of the election

- > 1.5 millions: number of eligible voters (French citizens abroad only)
- 11: number of deputies to elect, i.e. constituencies
- ~200: number of consulates
Some numbers…

May 27th — June 1st  first round of the election
June 10th — June 15th  second round of the election

> 1.5 millions  number of eligible voters (French citizens abroad only)

11  number of deputies to elect, i.e. constituencies

~200  number of consulates

⚠️ The results are published at the consulates level!
Some numbers...

May 27th — June 1st  
first round of the election
June 10th — June 15th  
second round of the election

> 1.5 millions  
number of eligible voters (French citizens abroad only)

11  
number of deputies to elect, i.e. constituencies

~200  
number of consulates

The results are published at the consulates level!

~524 000  
number of expressed votes (~251k first round and ~273k second round)

76,9%  
percentage of online voting (22,7% in person, 0,3% postal voting)
4 stakeholders

1. Organizer: the French Ministry of Europe and Foreign Affairs (the ministry)
4 stakeholders

1. Organizer: the French Ministry of Europe and Foreign Affairs (the ministry)

2. Institutional security advisor: the French National Cybersecurity Agency (ANSSI)
4 stakeholders

1. **Organizer:** the French Ministry of Europe and Foreign Affairs (the ministry)

2. **Institutional security advisor:** the French National Cybersecurity Agency (ANSSI)

3. **Vendor/service provider:** Voxaly Docaposte (Voxaly or the vendor)
4 stakeholders

1. Organizer: the French Ministry of Europe and Foreign Affairs (the ministry)

2. Institutional security advisor: the French National Cybersecurity Agency (ANSSI)

3. Vendor/service provider: Voxaly Docaposte (Voxaly or the vendor)

4. External third party: V. Cortier, P. Gaudry and S. Glondu (the Loria)
#Législatives2022 Être français, habiter à l'étranger & pouvoir vérifier son vote, ce sera possible ! Mandatés comme tiers de confiance par @francediplo & le @CNRS, des scientifiques @labo_Loria @Inria mettront en ligne un site sécurisé. 

+ d'infos : verifiabilite-legislatives2022.fr/informations.h...
“Be French, live abroad, and be able to verify your vote, it will be possible. […] Acting as trusted third parties, researchers will launch a secure website.”
Hum... is it true? Is it really secure?

“Be French, live abroad, and be able to verify your vote, it will be possible. [...] Acting as trusted third parties, researchers will launch a secure website.”
Outline

1. Reverse the threat model and the protocol

2. Vulnerabilities, attacks, and fixes
   ▶ how to defeat verifiability?
   ▶ how to defeat vote privacy?

3. Other concerns and take away
How to define the security targets?

1. The Code électoral (the French law)
How to define the security targets?

1. The Code électoral
   (the French law)

2. The CNIL recommendations
   (National Commission on Informatics and Liberty in English)
   ➔ level 3 is expected
How to define the security targets?

1. The Code électoral
   (the French law)

2. The CNIL recommendations
   (National Commission on Informatics and Liberty in English)

The CNIL recommendations are not legal requirements... but the protocol must meet them in practice any way!
Security properties

**Vote secrecy**

"Votes must remain confidential"
—Code électoral, Article R176-3-9

"[the system must] ensure the strict confidentiality of the ballots as soon as created."
—CNIL, Security objective n°1-04

"[The system must] ensure that the identity of the voter and the expression of his choice can not be linked during the whole process"
—CNIL, Security objective n°1-07

**Verifiability**

"When a voter's vote is registered, the voter is provided with a digital receipt allowing them to verify online that their vote has been taken into account."
—Code électoral, Article R176-3-9

"ensure the transparency of the ballot-box for all the voters [...] It must be possible for the voters to ensure that their ballot has been counted in the ballot-box."
—CNIL, Security objective n°2-07
Security properties

Vote secrecy

“Votes must remain confidential.”
—Code électoral, Article R176-3-9

“[the system must] ensure the strict confidentiality of the ballots as soon as created.”
—CNIL, Security objective n°1-04

 “[The system must] ensure that the identity of the voter and the expression of his choice can not be linked during the whole process”
—CNIL, Security objective n°1-07

An attacker cannot learn the choice of a target voter

Verifiability

“When a voter's vote is registered, the voter is provided with a digital receipt allowing them to verify online that their vote has been taken into account.”
—Code électoral, Article R176-3-9

"ensure the transparency of the ballot-box for all the voters [...] It must be possible for the voters to ensure that their ballot has been counted in the ballot-box."
—CNIL, Security objective n°2-07
Security properties

Vote secrecy

An attacker cannot learn the choice of a target voter

Verifyability

A voter must have the guarantee that their ballot appears in the ballot-box
Security properties

Vote secrecy

"Votes must remain confidential." — Code électoral, Article R176-3-9

"[the system must] ensure the strict confidentiality of the ballots as soon as created." — CNIL, Security objective n°1-04

"[The system must] ensure that the identity of the voter and the expression of his choice can not be linked during the whole process" — CNIL, Security objective n°1-07

An attacker cannot learn the choice of a target voter

Verifiability

"When a voter's vote is registered, the voter is provided with a digital receipt allowing them to verify online that their vote has been taken into account." — Code électoral, Article R176-3-9

"ensure the transparency of the ballot-box for all the voters [...] It must be possible for the voters to ensure that their ballot has been counted in the ballot-box." — CNIL, Security objective n°2-07

A voter must have the guarantee that their ballot appears in the ballot-box

Ballots must be sent by legitimate voters only

The result must corresponds to the content of the ballot-box

?
"Security level 3: The threat actors include the voters, the election operators, outsiders, insiders within the provider or internal staff. They can be resourceful or highly motivated."

—CNIL, Security level 3
"Security level 3: The threat actors include the voters, the election operators, outsiders, insiders within the provider or internal staff. They can be resourceful or highly motivated."

—CNIL, Security level 3

**Objective #3-02:** The system must allow transparency of the ballot-box for all voters from third-party tools.
Threat model

"Security level 3: The threat actors include the voters, the election operators, outsiders, insiders within the provider or internal staff. They can be resourceful or highly motivated."

—CNIL, Security level 3

Objective #3-02: The system must allow transparency of the ballot-box for all voters from third-party tools.

<table>
<thead>
<tr>
<th>Voter</th>
<th>Voting device</th>
<th>Com. channels</th>
<th>Voting server</th>
<th>Dec. auth.</th>
<th>3rd-party</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiability</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
</tr>
</tbody>
</table>

😊 = trustworthy
😊 = compromised
"Security level 3: The threat actors include the voters, the election operators, outsiders, insiders within the provider or internal staff. They can be resourceful or highly motivated."

—CNIL, Security level 3

Objective #3-02: The system must allow transparency of the ballot-box for all voters from third-party tools.

<table>
<thead>
<tr>
<th>Verifiability</th>
<th>Voter</th>
<th>Voting device</th>
<th>Com. channels</th>
<th>Voting server</th>
<th>Dec. auth.</th>
<th>3rd-party</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
<td>🌻</td>
</tr>
</tbody>
</table>

= trustworthy

= compromised
"Security level 3: The threat actors include the voters, the election operators, outsiders, insiders within the provider or internal staff. They can be resourceful or highly motivated."
—CNIL, Security level 3

Objective #3-02: The system must allow transparency of the ballot-box for all voters from third-party tools.

<table>
<thead>
<tr>
<th>Voter</th>
<th>Voting device</th>
<th>Com. channels</th>
<th>Voting server</th>
<th>Dec. auth.</th>
<th>3rd-party</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiability</td>
<td>✔️</td>
<td>✋</td>
<td>✔️</td>
<td>✋</td>
<td>✋</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>✔️</td>
<td>✋</td>
<td>✋</td>
<td>✋</td>
<td>✋</td>
</tr>
</tbody>
</table>

- ✔️ = trustworthy
- 😈 = compromised

TLS is broken (e.g., middle-box TLS, corrupted network administrator, …)
Threat model

"Security level 3: The threat actors include the voters, *the election operators*, outsiders, *insiders within the provider or internal staff*. They can be resourceful or highly motivated."

—CNIL, Security level 3

<table>
<thead>
<tr>
<th>Objective #3-02:</th>
<th>The system must allow transparency of the ballot-box for all voters from <em>third-party tools</em>.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Voter</th>
<th>Voting device</th>
<th>Com. channels</th>
<th>Voting server</th>
<th>Dec. auth.</th>
<th>3rd-party</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiability</td>
<td>☠️</td>
<td>☠️</td>
<td>☠️</td>
<td>☠️</td>
<td>☠️</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>☠️</td>
<td>☠️</td>
<td>☠️</td>
<td>☠️</td>
<td>☠️</td>
</tr>
</tbody>
</table>

- ☠️ = trustworthy
- 🙁 = compromised
- ☠️* = trustworthy (However, compromise decreases attacks complexity.)

*TLS is broken (e.g. middle-box TLS, corrupted network administrator, …)*
How to obtain a comprehensive description of the protocol?

A specification of the system

- published by Voxaly Docapost on April 21\textsuperscript{st} 2022
- allowing one to develop a third party verifier

⚠️ This specification is incomplete... it does not describe the protocol itself!
How to obtain a comprehensive description of the protocol?

A specification of the system

- published by Voxaly Docapost on April 21st 2022
- allowing one to develop a third party verifier

⚠️ This specification is incomplete... it does not describe the protocol itself!

Some reverse engineering

- based on the voter's journey (official tutorial and observation in-situ)
- based on HTML/JS/CSS data collected by different voters
- cross checking those data with data collected during a previous large-scale test
Security by obfuscation?

Standard obfuscation techniques:

- function and variable renaming
- control flow alteration (infinite for loop and breaks, switch case, nested functions, etc)
Security by obfuscation?

Standard obfuscation techniques:
- function and variable renaming
- control flow alteration (infinite for loop and breaks, switch case, nested functions, etc)

Few funny elements...
- it's mix of French and English: bulletin, codeActivation, erreurHashVerification, ...
correctLength, chosenCandidates, updateVoteStatus, ...

- obfuscation “by-design”, e.g, o.voteSignature is not a signature 🐞
A comprehensive description of the protocol
A comprehensive description of the protocol

1. Authentication: the voter sends their login/password to the server
A comprehensive description of the protocol

1. **Authentication**: the voter sends their login/password to the server

2. **Vote section and confirmation**
A comprehensive description of the protocol

1. **Authentication:** the voter sends their login/password to the server

2. **Vote section and confirmation**

3. **Code activation:** once confirmed, the voter initiates the sending of the activation code by email
A comprehensive description of the protocol

1. **Authentication**: the voter sends their login/password to the server

2. **Vote section and confirmation**

3. **Code activation**: once confirmed, the voter initiates the sending of the activation code by email

4. **Sending the ballot**: the voter sends their ballot together with the activation code
A comprehensive description of the protocol

1. Authentication: the voter sends their login/password to the server

2. Vote section and confirmation

3. Code activation: once confirmed, the voter initiates the sending of the activation code by email

4. Sending the ballot: the voter sends their ballot together with the activation code

🤔 Why is the ballot sent twice… ?
1. **Authentication**: the voter sends their login/password to the server

2. **Vote section and confirmation**

3. **Code activation**: once confirmed, the voter initiates the sending of the activation code by email

4. **Sending the ballot**: the voter sends their ballot together with the activation code

   🤔 Why is the ballot sent twice…?

5. **Receiving the receipt**: the server sends the PDF receipt to the voter
A comprehensive description of the protocol

1. **Authentication:** the voter sends their login/password to the server

2. **Vote section and confirmation**

3. **Code activation:** once confirmed, the voter initiates the sending of the activation code by email

4. **Sending the ballot:** the voter sends their ballot together with the activation code

   😐 Why is the ballot sent twice… ?

5. **Receiving the receipt:** the server sends the PDF receipt to the voter

This is the first public comprehensive description of the protocol.
Outline

1. Reverse the threat model and the protocol

2. Vulnerabilities, attacks, and fixes
   - how to defeat verifiability?
   - how to defeat vote privacy?

3. Other concerns and take away
More details about the receipt

1. Reference of the ballot: \( H = \text{hash}(\text{ballot} \& \text{context}) \)

2. Seal of the ballot: \( cSU = \text{sign}_{skS}(\text{ballot} \& \text{context}') \)

3. Ballot fingerprint: \( hb = \text{hash}(\text{ballot}) \)
More details about the receipt

1. **Reference of the ballot:** \( H = \text{hash}(\text{ballot} \& \text{context}) \)

2. **Seal of the ballot:** \( cSU = \text{sign}_{skS}(\text{ballot} \& \text{context'}) \)

3. **Ballot fingerprint:** \( hb = \text{hash}(\text{ballot}) \) 😢 This is useless...

---

**Elections législatives 2022 1er tour**

Preuve de dépôt du bulletin de vote dans l’urne

Votre bulletin de vote a bien été introduit dans l’urne électronique.

La référence ci-dessous vous permet de contrôler que votre bulletin est bien dans l’urne.

\[ \text{Valeur chiffrée de votre bulletin de vote : } 80011\&1\&3318f83ea80861c9d8f3d7d90f7y7966f87y598a9d76ef9689 \]

Pour contrôler la référence de votre bulletin : cliquez ici

https://votefae.diplomatie.gouv.fr/pages/verifierEmpreinte

---

Voici la preuve de dépôt de votre bulletin dans l’urne.

Votre bulletin de vote a bien été introduit dans l’urne électronique.

La référence ci-dessous vous permet de contrôler que votre bulletin est bien dans l’urne.

\[ \text{Valeur chiffrée de votre bulletin de vote : } 80011\&1\&3318f83ea80861c9d8f3d7d90f7y7966f87y598a9d76ef9689 \]

Pour contrôler la référence de votre bulletin : cliquez ici

https://votefae.diplomatie.gouv.fr/pages/verifierEmpreinte

---

Ce cachet électronique vous permet également de vérifier que votre preuve de vote a bien été produite par le système de vote homologué.

La valeur chiffrée de votre bulletin de vote ci-dessous vous permet de contrôler que le contenu de votre bulletin de vote est identique tout au long du scrutin. Cette valeur est à comparer avec celle obtenue en vérifiant la présence de votre bulletin dans l’urne.

\[ \text{Valeur chiffrée de votre bulletin de vote : } 80011\&1\&3318f83ea80861c9d8f3d7d90f7y7966f87y598a9d76ef9689 \]

Pour contrôler le cachet électronique, cliquez ici

https://votefae.diplomatie.gouv.fr/pages/verificationCachetServeur

---

Cette preuve vous permet également de vérifier que le dépouillement est conforme aux exigences de la CNIL, en matière de transparence de l’urne. Pour ce faire, vous devrez renseigner le cachet électronique ci-dessous.

Vous pouvez accéder à l’outil en cliquant ici.
1. Reference of the ballot:  \( H = \text{hash}(\text{ballot} \& \text{context}) \)

2. Seal of the ballot:  \( cSU = \text{sign}_{skS}(\text{ballot} \& \text{context'}) \)

3. Ballot fingerprint:  \( hb = \text{hash}(\text{ballot}) \)  

This is useless...

Vulnerability 1:

- The seal is not checked by the voting device
- \( H \) is computed by the voting device (\( H' \)) and received from the server 4 times (\( H_{S1}, H_{S2}, H_{S3}, H_{S4} \)).

  ➡️ the device ensures only: \( H' = H_{S1} = H_{S3} \)
  ➡️ the voter can only see \( H_{S2} \) and \( H_{S4} \)
More details about the receipt

1. **Reference of the ballot:**  \( H = \text{hash}(\text{ballot} \& \text{context}) \)

2. **Seal of the ballot:**  \( cSU = \text{sign}_{skS} (\text{ballot} \& \text{context'}) \)

3. **Ballot fingerprint:**  \( hb = \text{hash}(\text{ballot}) \)  😞 This is useless...

**Vulnerability 1:**
- The seal is not checked by the voting device
- \( H \) is computed by the voting device \((H^c)\) and received from the server 4 times \((H_{s1}, H_{s2}, H_{s3}, H_{s4})\).
  - \( H^c \) the device ensures only: \( H^c = H_{s1} = H_{s3} \)
  - the voter can only see \( H_{s2} \) and \( H_{s4} \)
More details about the receipt

1. Reference of the ballot: \( H = \text{hash}(\text{ballot} \& \text{context}) \)

2. Seal of the ballot: \( cSU = \text{sign}_{skS}(\text{ballot} \& \text{context}') \)

3. Ballot fingerprint: \( hb = \text{hash}(\text{ballot}) \) 😞 This is useless...

Vulnerability 1:
- The seal is not checked by the voting device
- \( H \) is computed by the voting device \( (H') \) and received from the server 4 times \( (H_{S1}, H_{S2}, H_{S3}, H_{S4}) \).
  ➡️ the device ensures only: \( H' = H_{S1} = H_{S3} \)
  ➡️ the voter can only see \( H_{S2} \) and \( H_{S4} \)

Vulnerability 2: The ballot \( b \) is not cryptographically bound to the consulate, i.e. \( \text{ballotBoxId} \)
Attack against verifiability

The references seen by the voter may not correspond to their ballot.
Attack against verifiability

The references seen by the voter may not correspond to their ballot.

Voting server
Attack against verifiability

The references seen by the voter may not correspond to their ballot.

Step 1: Alice votes as expected

\[ b_1, H_1^{s_1}, H_1^{s_2}, H_1^{s_3}, H_1^{s_4}, cSU_1 \]
The references seen by the voter may not correspond to their ballot.

**Step 1:** Alice votes as expected

**Step 2:** the attacker intercepts Bob’s request
- computes $H_{1}^{s_{1}}$ and $H_{1}^{s_{3}}$ as expected
- replays Alice’s data otherwise
The references seen by the voter may not correspond to their ballot.

Step 1: Alice votes as expected

Step 2: the attacker intercepts Bob’s request
- computes $H_2^{s_1}$ and $H_2^{s_3}$ as expected
- replays Alice’s data otherwise

Result: Bob’s ballot is dropped… but nothing went wrong in Bob’s process
Attack against verifiability

The references seen by the voter may not correspond to their ballot.

Step 1: Alice votes as expected

Step 2: the attacker intercepts Bob’s request
  ▶ computes $H_{2}^{s_1}$ and $H_{2}^{s_3}$ as expected
  ▶ replays Alice’s data otherwise

Result: Bob’s ballot is dropped… but nothing went wrong in Bob’s process

Improvement: the attacker can completely modify Bob’s ballot
An almost undetectable attack

1. No error detected during the voting process: \( H_2^c = H_2^{s_1} = H_2^{s_3} \neq H_1^{s_2} = H_1^{s_4} \)
   but this check is never done….

2. Bob receives a valid receipt: Bob’s receipt correspond to Alice’s ballot or the attacker’s ballot…
   both are included in the ballot-box \(\Rightarrow\) verifications succeed
An almost undetectable attack

1. No error detected during the voting process: \( H_2^C = H_2^{S_1} = H_2^{S_3} \neq H_1^{S_2} = H_1^{S_4} \)
   but this check is never done….

2. Bob receives a valid receipt: Bob’s receipt correspond to Alice’s ballot or the attacker’s ballot… both are included in the ballot-box \( \Rightarrow \) verifications succeed

The Loria’s verifier is useless to guarantee individual verifiability…
An almost undetectable attack

1. No error detected during the voting process: \( H_2^c = H_2^{s_1} = H_2^{s_3} \neq H_1^{s_2} = H_1^{s_4} \)
   but this check is never done….

2. Bob receives a valid receipt: Bob’s receipt correspond to Alice’s ballot or the attacker’s ballot… both are included in the ballot-box \( \Rightarrow \) verifications succeed

   The Loria’s verifier is useless to guarantee individual verifiability…

In rare cases, detection is possible…

- Attack 1 (drop only): Bob can see on the signing sheet that he is considered as absentee 
  \( \Rightarrow \) requires Bob goes to the polling station… it seems unlikely…
An almost undetectable attack

1. No error detected during the voting process: \( H_2^c = H_2^{s_1} = H_2^{s_3} \neq H_1^{s_2} = H_1^{s_4} \)
   but this check is never done….

2. Bob receives a valid receipt: Bob’s receipt correspond to Alice’s ballot or the attacker’s ballot…
   both are included in the ballot-box \( \Rightarrow \) verifications succeed

The Loria’s verifier is useless to guarantee individual verifiability…

In rare cases, detection is possible…

- **Attack 1 (drop only):** Bob can see on the signing sheet that he is considered as absentee
  - requires Bob goes to the polling station… it seems unlikely…

- **Attack 2 (drop and replace):** detectable if no-one else voted for Bob’s candidate
  - unlikely in large consulates…
Attack against vote secrecy

The ballot $b$ are not cryptographically bound to the consulate
Attack against vote secrecy

The ballot $b$ are not cryptographically bound to the consulate
Attack against vote secrecy

The ballot \( b \) are not cryptographically bound to the consulate

Consulate 1

E.g.
SIDNEY consulate

Consulate 2

E.g.
EKATERINBURG consulate

Compromised voting server
Attack against vote secrecy

The ballot $b$ are not cryptographically bound to the consulate

E.g SIDNEY consulate

Consulate 1

Consulate 2

E.g EKATERINBURG consulate
Attack against vote secrecy

The ballot \( b \) are not cryptographically bound to the consulate.

Consulate 1

Consulate 2

Compromised voting server

E.g. SIDNEY consulate

E.g. EKATERINBURG consulate
Attack against vote secrecy

The ballot $b$ are not cryptographically bound to the consulate

Consulate 1

Consulate 2

Compromised voting server

E.g EKATERINBURG consulate

E.g SIDNEY consulate
Attack against vote secrecy

The ballot $b$ are not cryptographically bound to the consulate

E.g. SIDNEY consulate

E.g. EKATERINBURG consulate

Consulate 1

Consulate 2

Compromised voting server
Attack against vote secrecy

The ballot $b$ are not cryptographically bound to the consulate

E.g. SIDNEY consulate

Consulate 1

Consulate 2

E.g. EKATERINBURG consulate

Compromised voting server
Attack against vote secrecy

The ballot are not cryptographically bound to the consulate

E.g. SIDNEY consulate

E.g. EKATERINBURG consulate

Consulate 1

Consulate 2

Compromised voting server

Tally of Consulate 2 reveals Alice’s choice
Impact of the attack

Assumptions to mount a completely undetectable attack:

- a channel attacker is enough
- at least as many corrupted voter as candidates
- at least as many expressed votes as candidates in the small consulate
- at least one vote per candidate in the large consulate
Impact of the attack

Assumptions to mount a completely undetectable attack:

- a channel attacker is enough
- at least as many corrupted voter as candidates
- at least as many expressed votes as candidates in the small consulate
- at least one vote per candidate in the large consulate

Impact

- can learn the choice or a bias on the choice of target voters: one per “small” consulate
- could contribute to remote coercion attacks: gather and isolate all coerced voters ballots in the same consulate
- is completely undetectable
Summary of attacks

1- Individual verifiability does not hold
Despite the use of a third-party verifier, an attacker who compromises the communication channels (or even worse the voting server) can significantly modify the outcome of the election by dropping and replacing ballots.

2- Vote secrecy does not hold
An attacker who compromises the communication channels (or even more so the voting server) can learn the plaintext vote of arbitrary target voters. The number of voters who can be targeted is immediately related to the number of consulates with a small number of votes cast.
Summary of attacks

1- Individual verifiability does not hold
Despite the use of a third-party verifier, an attacker who compromises the communication channels (or even worse the voting server) can significantly modify the outcome of the election by dropping and replacing ballots.

2- Vote secrecy does not hold
An attacker who compromises the communication channels (or even more so the voting server) can learn the plaintext vote of arbitrary target voters. The number of voters who can be targeted is immediately related to the number of consulates with a small number of votes cast.

Very easy fixes
- display locally created data to the voter only (i.e. create the PDF in local)
- add ballotBoxId in the context of the ZKPs
Summary of attacks

1- Individual verifiability does not hold
Despite the use of a third-party verifier, an attacker who compromises the communication channels (or even worse the voting server) can significantly modify the outcome of the election by dropping and replacing ballots.

2- Vote secrecy does not hold
An attacker who compromises the communication channels (or even more so the voting server) can learn the plaintext vote of arbitrary target voters. The number of voters who can be targeted is immediately related to the number of consulates with a small number of votes cast.

Very easy fixes
- display locally created data to the voter only (i.e. create the PDF in local)
- add ballotBoxId in the context of the ZKPs

We detail 6 different variants of these attacks and propose fixes in the full report!

[ePrint 2022/1653]
Outline

1. Reverse the threat model and the protocol

2. Vulnerabilities, attacks, and fixes
   ▶ how to defeat verifiability?
   ▶ how to defeat vote privacy?

3. Other concerns and take away
On the importance of…the voting device

Regarding security, the key element is the voting device…(not the voting server)
On the importance of... the voting device

Regarding security, the key element is the voting device... (not the voting server)

<table>
<thead>
<tr>
<th></th>
<th>Voter</th>
<th>Voting device</th>
<th>Com. channels</th>
<th>Voting server</th>
<th>Dec. auth.</th>
<th>3rd-party</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifiability</td>
<td>☀️</td>
<td>☐️</td>
<td>☩️</td>
<td>☩️</td>
<td>☩️</td>
<td>☐️</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>☀️</td>
<td>☐️</td>
<td>☩️</td>
<td>☩️</td>
<td>☩️</td>
<td>☐️</td>
</tr>
</tbody>
</table>

It's the unique trustworthy component

- ☀️ = trustworthy
- ☩️ = untrustworthy
On the importance of... the voting device

Regarding security, the key element is the voting device... (not the voting server)

It's the unique trustworthy component

Now, the voting client is a Javascript program provided by the server...

➡ need to find a solution to make it really trustworthy? (transparency, audibility...)
➡ ensure cast-as-intended?
On the importance of...
the eligibility

Today, authentication is ensured by an untrustworthy server and an (almost) inaccessible signing sheet....
On the importance of... the eligibility

Today, authentication is ensured by an untrustworthy server and an (almost) inaccessible signing sheet....

3 authentication element:
- a login sent by the service provider Orange by email
- a password sent by the service provider mTarget
- an activation code sent on-the-flight by Orange too
On the importance of…
the eligibility

Today, authentication is ensured by an untrustworthy server and an (almost) inaccessible signing sheet….

3 authentication element:
- a login sent by the service provider Orange by email
- a password sent by the service provider mTarget
- an activation code sent on-the-flight by Orange too

But a ballot contains none of them… the voting server can vote for absentees…
On the importance of…
the eligibility

Today, authentication is ensured by an untrustworthy server and an (almost) inaccessible signing sheet…

3 authentication element:
- a login sent by the service provider Orange by email
- a password sent by the service provider mTarget
- an activation code sent on-the-flight by Orange too

But a ballot contains none of them… the voting server can vote for absentees…

Can we improve the protocol to prevent such a weakness? Yes, we think so!
(but we have no solution to present for now…)
On the importance of...
the literature

the system suffers from well-known vulnerabilities...
On the importance of...
the literature

the system suffers from well-known vulnerabilities...

A lack of elements in the ZKPs contexts leads to attacks...

• our vote secrecy attacks
• Cortier and Smyth attack (2011) to break verifiability and vote secrecy
• (maybe) Cortier, Gaudry and Yang attack (2020) to break verifiability
On the importance of the literature

The system suffers from well-known vulnerabilities...

A lack of elements in the ZKPs contexts leads to attacks...

- our vote secrecy attacks
- Cortier and Smyth attack (2011) to break verifiability and vote secrecy
- (maybe) Cortier, Gaudry and Yang attack (2020) to break verifiability

Fixes are really easy to implement!
On the importance of... the literature

the system suffers from well-known vulnerabilities...

A lack of elements in the ZKPs contexts leads to attacks...
- our vote secrecy attacks
- Cortier and Smyth attack (2011) to break verifiability and vote secrecy
- (maybe) Cortier, Gaudry and Yang attack (2020) to break verifiability

No weeding makes ballot replay attacks possible...
- an attacker can replay Alice’s ballot to bias the result and learn Alice’s choice
- impact recently studied by Mestel et. al. (2022)

 Fixes are really easy to implement!
On the importance of... the literature

the system suffers from well-known vulnerabilities...

A lack of elements in the ZKPs contexts leads to attacks...

- our vote secrecy attacks
- Cortier and Smyth attack (2011) to break verifiability and vote secrecy
- (maybe) Cortier, Gaudry and Yang attack (2020) to break verifiability

No weeding makes ballot replay attacks possible...

- an attacker can replay Alice’s ballot to bias the result and learn Alice’s choice
- impact recently studied by Mestel et. al. (2022)
We provide the first public and comprehensive specification of the protocol.

We show that the system fails to ensure verifiability and vote secrecy under a reasonable threat model:
- assumes a channel attacker only
- 6 attacks, some of them being completely undetectable

We propose fixes for each attack and recall well-known vulnerability and fixes of the literature that the protocol should implement.

Some of our fixes is will be implemented for future elections, others will depend on the timeline...
Hope for the future

We hope our recommendations will be taken into account for the next public tender…

- define a clearer threat model
- pay attention to the threats and vulnerabilities we pointed out
- push for more transparency, in particular regarding the voting device
Hope for the future

We hope our recommendations will be taken into account for the next public tender...

- define a clearer threat model
- pay attention to the threats and vulnerabilities we pointed out
- push for more transparency, in particular regarding the voting device

Still open questions to improve the system:

- **Eligibility:** develop new techniques or convince authorities to use existing ones…?
- **Cast-as-intended:** random audits or plaintext verification don’t seem acceptable… what can we do?
- **Vote secrecy:** 4 out of 16 is not acceptable… can we do better?
Hope for the future

We hope our recommandations will be taken into account for the next public tender…

- define a clearer threat model
- pay attention to the threats and vulnerabilities we pointed out
- push for more transparency, in particular regarding the voting device

Still open questions to improve the system:

- **Eligibility:** develop new techniques or convince authorities to use existing ones…?
- **Cast-as-intended:** random audits or plaintext verification don’t seem acceptable… what can we do?
- **Vote secrecy:** 4 out of 16 is not acceptable… can we do better?

Thank you!