Reversing, Breaking, and Fixing the French Legislative Election E-Voting Protocol

Alexandre Debant and Lucca Hirschi

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

Pesto team seminar
November 18th 2022
Some numbers…

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

> 1.1 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

> 1.1 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

> 1.1 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

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

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

2. Institutional security advisor: the French National Cybersecurity Agency

3. Vendor/service provider: Voxaly Docaposte
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)
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)

Responsible disclosure: all the elements presented in this talk have been firstly reported and discussed with those entities.
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
(not exhaustive)

“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

Vote secrecy
Security properties
(not exhaustive)

“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

Vote secrecy

An attacker cannot learn the choice of a target voter
**Security properties**

*not exhaustive*

**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

**Individual 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

- An attacker cannot learn the choice of a target voter
Security properties
(not exhaustive)

**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

**Individual 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

**An attacker cannot learn the choice of a target voter**

**A voter must have the guarantee that their ballot appears in the ballot-box**
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
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></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>

= trustworthy

= compromised
**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></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><strong>Verifiability</strong></td>
<td>🌻</td>
<td>🌻</td>
<td>🐶</td>
<td>🐶</td>
<td>🐶</td>
<td>🌻</td>
</tr>
<tr>
<td><strong>Confidentiality</strong></td>
<td>🌻</td>
<td>🌻</td>
<td>🐶</td>
<td>🐶</td>
<td>🐶</td>
<td>🌻</td>
</tr>
</tbody>
</table>

- 🌻 = trustworthy
- 🐶 = compromised

Cast-as-intended is acknowledge as not satisfied
"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>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>Confidentiality</td>
<td>🌟</td>
<td>🌟</td>
<td>🚨</td>
<td>🚨</td>
<td>🚨</td>
<td>🌟</td>
</tr>
</tbody>
</table>

🌟 = trustworthy
🚨 = compromised

Cast-as-intended is acknowledge as not satisfied

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></th>
<th>Voter</th>
<th>Voting device</th>
<th>Com. channels</th>
<th>Voting server</th>
<th>Dec. auth.</th>
<th>3\textsuperscript{rd}-party</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Verifiability</strong></td>
<td>☺</td>
<td>☺</td>
<td>☻</td>
<td>☻*</td>
<td>☻*</td>
<td>☻</td>
</tr>
<tr>
<td><strong>Confidentiality</strong></td>
<td>☺</td>
<td>☺</td>
<td>☻</td>
<td>☻*</td>
<td>☻*</td>
<td>☻</td>
</tr>
</tbody>
</table>

- ☺ = trustworthy
- ☻ = compromised
- ☻* = trustworthy (However, compromise decreases attacks complexity.)

Cast-as-intended is acknowledge as not satisfied

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 21st 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
Reverse in practice

Standard obfuscation techniques:

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

- use of the tool js - beautify to de-minimize the code
  - ~16k LoC in 4 interesting Javascript files
  - the file app.bundle.js contains the core logic of the protocol and loria.bundle.js the crypto primitives
- object attributes and HTML request are not obfuscated
- remains close to the full-scale test code which is much less obfuscated: side-by-side comparison possible
Concrete example:
core logic to forge the reference

```javascript
function(e, t, i, n, a) { // core logic computing HashClient starts after
e.vote = function() {
  if (data.param.signatureEnabled && t.e.Avote) {
    e.eVote = !0, e.errorHashVerification = !1;
    var i = forge.md.sha256.create()
    i.update(e.bulletinCrypte + data.election.ordre + data.param.
      electeurOrdre);
    var n = i.digest().toHex(),
    a = function(e) {
      [...]
    })(n),
    o = data.election.ordre + "&" + n + a;
    sessionStorage.setItem("HashClient", o);
}
```

Test phase

```javascript
function(e, t, n) {
  function ot(e) {
    [...]
    function v() {
      return (v = Pe())(Re.a.mark((function t() {
        var n, r, a, l, u, c, s;
        return Re.a.wrap((function(t) {
          for (;;) switch (t.prev = t.next) {
            case 0:
              [...]
            case 3: // core logic computing HashClient starts here
              return (n = new jsSHA("SHA-256", "TEXT")).update(o.
                bulletinCrypte + f.idTour + d.ordre + f.electeurOrdre),
              r = n.getHash("HEX"),
              (a = new jsSHA("SHA-256", "TEXT")).update(o.bulletinCrypte +
                o.voteSignature),
              l = a.getHash("HEX"),
              u = f.idTour + "&" + d.ordre + "&" + r + y(r),
              sessionStorage.setItem("HashClient", u),
```

Production phase
Concrete example:
core logic to forge the reference

```javascript
1 navclientApp.controller("PageVoteController", ['$scope', 'httpClient',
  'location', '$timeout', 'breadcrumbService',
  function(e, t, i, n, a) { // core logic computing HashClient starts after
e.vote = function() {
  if (data.param.signatureEnabled && !a) {
    e.evote = !0; e.erreurHashVerification = !1;
    var i = forge.md.sha256.create();
    i.update(e.bulletinCrypte + data.election.ordre + data.param.
      electeurEtOrdre);
    var a = i.digest().toHex(),
    a = function(e) {
      [...] }
  }
  o = data.election.ordre + "\&" + n + a;
  sessionStorage.setItem("HashClient", o);
}
```

Test phase

Production phase

```javascript
1 function(e, t, n) {
2   function ot(e) {
3     [...] }  
4   function v(t) {
5     return (v = Pe() && a.mark((function t() {
6       var n, r, a, l, u, c, s;
7       return Re.a.wrap((function(t) {
8         for (;) switch (t.prev = t.next) {
9           case 0: [...]
10           case 3: // core logic computing HashClient starts here
11           return (n = new jsSHA("SHA-256", "TEXT"}).update(o.
12             bulletinCrypte + t.idTour + d.ordre + t.electeurEtOrdre),
13             r = n.getHash("HEX"),
14             (a = new jsSHA("SHA-256", "TEXT")).update(o.bulletinCrypte +
15             o.voteSignature),
16             t = a.getHash("HEX"),
17             u = t.idTour + "\&" + d.ordre + "\&" + r + y(r),
18             sessionStorage.setItem("HashClient", u),
```

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…?
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

   �ум 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
Elections législatives 2022 1er tour

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 de vote est bien dans l'urne.

Pour contrôler la référence de votre bulletin : cliquez ici
https://votefae.diplomatie.gouv.fr/pages/verifierEmpreinte

Une fois le dépouillement effectué, vous pouvez vérifier que votre bulletin de vote a bien été prise en compte dans le calcul des résultats, à l'aide d'un outil tiers développé par le CNRS, conformément aux exigences de la CNIL en matière de transparence de l'urne. Pour ce faire, vous devrez renseigner le cachet électronique ci-dessous.

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

Pour contrôler le cachet électronique, cliquez ici
https://votefae.diplomatie.gouv.fr/pages/verificationCachetServeur

Une fois le dépouillement effectué, vous pouvez vérifier que votre bulletin a bien été pris en compte dans le calcul des résultats, à l'aide d'un outil tiers développé par le CNRS, conformément 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.
More details about the receipt

1. Reference of the ballot:

\[ H = \text{roundId}||\text{electionId}||\text{hash(b||roundId||electionId||ballotBoxId)} \]
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.

8001141331893939ea80861c89ddad73g90f77b969d97979689598aad76899689 654a76ad587a6

Pour contrôler la référence de votre bulletin : cliquez ici
https://votefae.diplomatie.gouv.fr/pages/verifierEmpreinte

Une fois le dépouillement effectué, vous pouvez vérifier que votre bulletin a bien été pris en compte dans le calcul des résultats, à l'aide d'un outil tiers développé par le CNRS, conformément 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.

Reference of the ballot:

\[ H = \text{roundId}||\text{electionId}||\text{hash}(b||\text{roundId}||\text{electionId}||\text{ballotBoxId}) \]

\[ H \] is computed by the voting device \((H^c)\) and received from the server 4 times \((H^1, H^2, H^3, H^4)\).
More details about the receipt

1. Reference of the ballot:

\[ H = \text{roundId}||\text{electionId}||\text{hash}(b||\text{roundId}||\text{electionId}||\text{ballotBoxId}) \]

\( H \) is computed by the voting device (\( H^s \)) and received from the server 4 times (\( H^s_1, H^s_2, H^s_3, H^s_4 \)).
More details about the receipt

1. Reference of the ballot:

\[ H = \text{roundId}||\text{electionId}||\text{hash(b||roundId||electionId||ballotBoxId)} \]

\( H \) is computed by the voting device (\( H^c \)) and received from the server 4 times (\( H^s_1, H^s_2, H^s_3, H^s_4 \)).

- the device ensures only: \( H^c = H^s_1 = H^s_3 \)
- the voter can only see \( H^s_2 \) and \( H^s_4 \)
More details about the receipt

1. Reference of the ballot:

\[ H = \text{roundId}||\text{electionId}||\text{hash}(b)||\text{roundId}||\text{electionId}||\text{ballotBoxId} \]

\( H \) is computed by the voting device \((H^c)\) and received from the server 4 times \((H^s_1, H^s_2, H^s_3, H^s_4)\).

\( \rightarrow \) the device ensures only: \(H^c = H^s_1 = H^s_3\)

\( \rightarrow \) the voter can only see \(H^s_2\) and \(H^s_4\)

2. Seal of the ballot:

\[ cSU = \text{infoSU}||\text{sign}_{sk_5}(\text{hash}(\text{infoSU})) \]

\( \text{infoSU} = \text{roundId}||\text{electionId}||\text{electionName}||\text{ballotBoxId}||\text{hash}(b) \)
More details about the receipt

1. Reference of the ballot:

$$H = roundId || electionId || \text{hash}(b || roundId || electionId || ballotBoxId)$$

$H$ is computed by the voting device ($H^C$) and received from the server 4 times ($H^s_1, H^s_2, H^s_3, H^s_4$).

- the device ensures only: $H^c = H^s_1 = H^s_3$
- the voter can only see $H^s_2$ and $H^s_4$

2. Seal of the ballot:

$$cSU = infoSU || \text{sign}_{skS}(\text{hash}(infoSU)),$$

$infoSU = roundId || electionId || electionName || ballotBoxId || \text{hash}(b)$

The ballot $b$ is not cryptographically bound to the consulate, i.e. $ballotBoxId$
Elections législatives 2022 1er tour

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.

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

VoterBoxId

La valeur chiffrée de votre bulletin de vote ci-dessous vous permet de vérifier 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.

Pour contrôler la valeur chiffrée, cliquez ici

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

Une fois le dépouillement effectué, vous pouvez vérifier que votre bulletin a bien été pris en compte dans le calcul des résultats, à l’aide d’un outil tiers développé par le CNRS, conformément 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:

\[ H = \text{roundId}||\text{electionId}||\text{hash}(b||\text{roundId}||\text{electionId}||\text{ballotBoxId}) \]

\( H \) is computed by the voting device \((H^c)\) and received from the server 4 times \((H^s_1, H^s_2, H^s_3, H^s_4)\).

⇒ the device ensures only: \( H^c = H^s_1 = H^s_3 \)

⇒ the voter can only see \( H^s_2 \) and \( H^s_4 \)

2. Seal of the ballot:

\[ cSU = \text{infoSU}||\text{sign}_{sk_5}(\text{hash} (\text{infoSU})) \]

\( \text{infoSU} = \text{roundId}||\text{electionId}||\text{electionName}||\text{ballotBoxId}||\text{hash}(b) \)

The ballot \( b \) is not cryptographically bound to the consulate, i.e. \( \text{ballotBoxId} \)

3. Ballot fingerprint:

\[ hb = \text{hash}(b) \]
Attack against verifiability

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

Step 1: Alice votes as expected
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
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
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
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....
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 ⇒ 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…
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…

- **Attack 2 (drop and replace):** detectable if no-one else voted for Bob’s candidate
  \( \Rightarrow \) unlikely in large consulates…
The seal $cSU$ and the ballot $b$ are not cryptographically bound to the consulate.
Attack against vote secrecy

The seal $cSU$ and the ballot $b$ are not cryptographically bound to the consulate.

Consulate 1

Consulate 2

Compromised voting server
Attack against vote secrecy

The seal $cSU$ and 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
The seal $cSU$ and the ballot $b$ are not cryptographically bound to the consulate.
Attack against vote secrecy

The seal $cSU$ and the ballot $b$ are not cryptographically bound to the consulate.

E.g. EKATERINBURG consulate

E.g. SIDNEY consulate
Attack against vote secrecy

The seal $cSU$ and the ballot $b$ are not cryptographically bound to the consulate.

Consulate 1

E.g. EKATERINBURG consulate

Consulate 2

E.g. SIDNEY consulate
The seal $cSU$ and the ballot $b$ are not cryptographically bound to the consulate.
Attack against vote secrecy

The seal \(cSU\) and 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 seal $cSU$ and the ballot $b$ are not cryptographically bound to the 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 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 worse 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.

We detail 6 different variants of these attacks and propose fixes in the full report!
[ePrint 2022/1653]

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
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 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)
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!

Everything is in place to make ballot weeding... but they weren’t aware of...
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

<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>

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

need to find a solution to make it really trustworthy
On the importance of... the voting device

Possible solutions to improve integrity of the voting device:
  ▶ use a standalone application; or
  ▶ use an easily auditable client (e.g., non obfuscated Single-page Application); or
  ▶ use other browser integrity services...
On the importance of...
the voting device

Possible solutions to improve integrity of the voting device:
- use a standalone application; or
- use an easily auditable client (e.g., non obfuscated Single-page Application); or
- use other browser integrity services...

Currently, the voting device is not auditable:
- the javascript code is provided after authentication
  - attacker can decide if he cheats depending of the voter
- the javascript is not static, it contains voter dependent data (e.g., consulate identifier)
  - audits can not guarantee that voters receive the correct data
On the importance of...
the voting device

Possible solutions to improve integrity of the voting device:

‣ use a standalone application; or
‣ use an easily auditable client (e.g., non obfuscated Single-page Application); or
‣ use other browser integrity services…

Currently, the voting device is not auditable:

‣ the javascript code is provided after authentication
  ➔ attacker can decide if he cheats depending of the voter
‣ the javascript is not static, it contains voter dependent data (e.g., consulate identifier)
  ➔ audits can not guarantee that voters receive the correct data

Vote privacy attack: attacker can exploit this weakness to mount our vote privacy attack
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…)
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.

One of our fixes is already implemented, others will depend on the timeline...
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.

**One of our fixes is already implemented**, others will depend on the timeline...

Details are in the full report on HAL (soon...)
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

We hope academic community will address open questions of practitioners

- can we improve eligibility in practice? Existing solutions does not seem appealing for practitioners…
- can we reflect more security features in the cryptography (e.g., the quorum for decryption)?
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

We hope academic community will address open questions of practitioners

- can we improve eligibility in practice? Existing solutions does not seem appealing for practitioners...
- can we reflect more security features in the cryptography (e.g., the quorum for decryption)?

Thank you!