An Analysis of the Security and Privacy Issues of the Neovote Online Voting System

Enka Blanchard

CNRS (French National Centre for Scientific Research),
Laboratory of industrial and human automation, mechanics and computer science (UPHF),
Centre for Internet and Society

www.koliaza.com

Joint work with **Antoine Gallais, Emmanuel Leblond, Djohar Le Clec'h-Sidhoum** and **Juliette Walter**

7th International Joint Conference on Electronic Voting, EVOTE-ID 2022 October 7th, 2022

French elections

French presidential elections in 2022 :

- 2-round paper election, no absentee but proxy voting
- 250M€ for campaigns and organisation
- 48.7 million voters, 5 major parties with primaries

Few regulations for primaries but 4 main rules for e-voting:

- Ensure privacy
- A posteriori verification of tallying
- Ensure transparency of ballot box
- Follow ANSSI rules

Previous and parallel work

Little analysis despite Neovote being one of the biggest French actors: 10k votes per year, competitive public markets and three primaries (EELV, LR, PP).

Criticism of the Primaire Populaire in general :

- Multiple registrations (also for LR primary)
- Tallying method (majority judgement)

Only one public analysis of Neovote, by *de Barros, Gergouil, Grelard and Thibault* :

- Internal elections in Bordeaux University
- Issues with ESMTPS, weak registration, low privacy

Whistle-blower for EELV primary election

Methodology

Methodological and ethical questions :

- Critical election, should we do anything that threatens its legitimacy ?
- How to analyse without interfering

Decisions:

- Register as usual voters without trying to influence the process unduly
- Record everything, download and analyse code (available upon request)
- Compare code with previous elections

Warn institutions and Neovote itself

Neovote's claims and transparency

From Neovote's website :

- 10k votes with no technical issues
- Homologated by top institutions (Senate, National Assembly...)
- Deployed on SecNumCloud but uses no cloud
- Non-standard vocabulary (avoiding mix-nets, random ballot boxes, geometric models)
- No external code, including modified Debian and full cryptographic stack
- No transparency "as they handle top-secret information", no idea of the models, algorithms or code used internally

Code comparisons

First problem: hard to download the scripts and refusal to interact with archiving websites

Main issue: obfuscation

- Regenerating the page changes all variable and function names
- There is a naming scheme
- Structure is maintained, strings too
- Scope sometimes prevents obfuscation

We compare structures and strings

AES_encrypt

```
AES_Encrypt_process
(data: Uint8Array):Uint8Array
{if (!is_bytes(data)) throw new
  TypeError("data isn't of expected type");

  let asm = this.asm;
  let heap = this.heap;
  let amode = AES_asm.ENC[this.mode];
  let hpos = AES_asm.HEAP_DATA;
  let pos = this.pos;
  let len = this.len;
  let dpos = 0;
  let dlen = data.length  0;
  let rpos = 0;
  let rlen = (len + dlen) & -16;
  let wlen = 0;

  let result = new Uint8Array(rlen);

      ⋮
```

```
.xTpmDHxL=function
($xTpmppgF)
{if(!xTpmppDL($xTpmppgF)){throw new
TypeError("data isn't of expected type");}

var $xTpmpDmV=this.$xTpmpDmV;
var $xTpmppgm=this.$xTpmppgm;
var $xTpmpDLT=xTpmpDNN.xTpmDHNL[this.$xTpmpDYs];
var $xTpmppgg=xTpmpDNN.xTpmDHxH;
var $xTpmpDYV=this.$xTpmpDYV;
var $xTpmprNV=this.$xTpmprNV;
var $xTpmppHx=0;
var $xTpmppHs=$xTpmppgF.length0;
var $xTpmpDLY=0;
var $xTpmpDLL=($xTpmprNV+$xTpmppHs)&-16;
var $xTpmppHg=0;

var $xTpmpDLD=new Uint8Array($xTpmpDLL);

      ⋮
```

RSAES-PKCS

```
export function getNonZeroRandomValues
(buf: Uint8Array)
{getRandomValues(buf);
  for (let  i = 0;
  i < buf.length;  i++) {
    let byte = buf[i];
    while (!byte) {
      const octet = new Uint8Array(1);
      getRandomValues(octet);
      byte = octet[0];
    }
    buf[i] = byte;}}
```

```
var xTpmpDDx=function
(xTpmpDpH)
{xTpmpDpD(xTpmpDpH);
for(var $xTpmprNs=0;
$xTpmprNs<xTpmpDpH.length;$xTpmprNs++){
var $xTpmpDDW=xTpmpDpH[$xTpmprNs];
while(!$xTpmpDDW){
var $xTpmpDDV=new Uint8Array(1);
xTpmpDpD($xTpmpDDV);
$xTpmpDDW=$xTpmpDDV[0];
}
xTpmpDpH[$xTpmprNs]=$xTpmpDDW;}
```

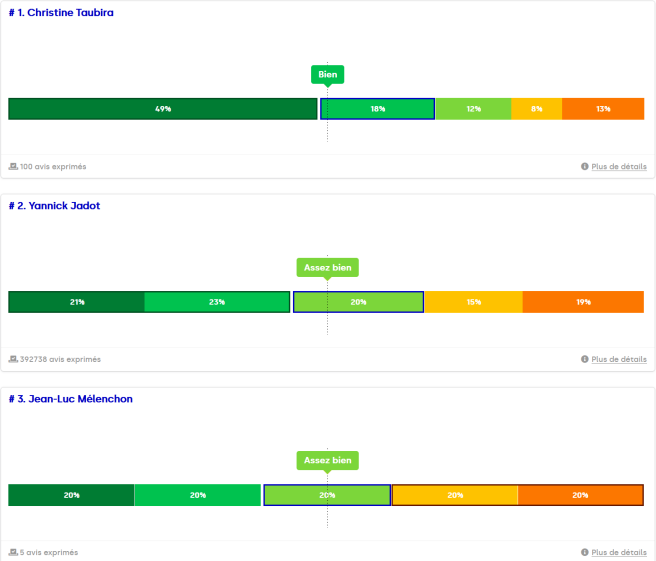
Publishing the tally

From exchanges with PP, their protocol :

- Raw results sent by Neovote to 3 parties : election administration, CNIL expert, and independent legal officer
- Independent confirmation that all went well
- Raw results not made public before control
- Final results computed and published online

Weaknesses because no E2E element and little transparency

Primaire Populaire vote



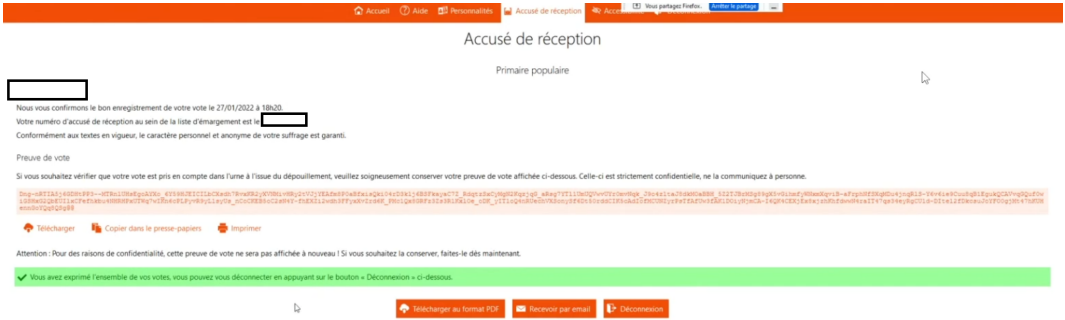
Verification availability and usability

Verification

- "You can check your vote later"
- Proof of vote (long string in browser)
- Forbidden to share proof of vote
- Easy to skip the proof of vote

For EELV it worked. For PP : no information on how to verify.

Verification availability and usability



Checking the verification

No publicly available code this time :

- Analyse EELV code
- Authenticate it by structure and function re-use
- Compatible with other analyses

Verification structure

Structure :

- Each receipt is composed of 5 hashes (computed on the client's side, encrypted using a constant public key)
- Ballot box is list of (ballot, hash)
- Extra (void) hashes in a different file

Verification protocol :

- Remove hashes present in *extra*
- Check that the remaining are in the ballot box
- Ask the server to decrypt all ballots (RSA using its public key)
- Tally the votes

Attacking the ballot box

Ballot box vulnerabilities

- The ballot box is not digitally signed
- We can create a fake ballot box with arbitrary data and make it indistinguishable from the real one

Receipt vulnerabilities

- The receipts are not digitally signed
- We can create fake-receipts attacks and denounce the election

Deanonymising voters

Hard to be sure, but current hash structure indicates that :

- If no extra hash is in ballot box (all in extra hashes) then voter can prove how they voted
- If some hashes only are present, voter can prove how they did not vote

We assume that hashes are generally not included in *extra hashes*.

- If voters don't share receipts, organisers can easily re-use hashes and modify the ballot box.
- If receipts are often made public, we can create sequences of receipts and prove how one voted (or did not vote) or have high probability that they voted a certain way.

Legal decisions

French legal decisions

- Appellate Court : cancelled a vote by Neovote in 2019 as not sufficiently guaranteeing privacy
- Supreme Court : expertise *in abstracto* is enough. Checking the system each time is not necessary
- Recommended privacy requirements are too weak (even with 2-factor, because of public data).

Conclusion

We see a multiplication of commercial e-voting systems.

The legal frameworks do not appear solid enough to impose good security.

How do we shift from technical expertise to mandated regulations, and make sure those are applied ?