Œ: The Open Electronic Ballot Box

Laurent Fournier

June 18, 2011

Version: *0.1* State: *alpha* [SHA1 digest[1]: 805438170dbd886f6829d850b196e1c7fcfe3649]

Abstract

Welcome to the Open Electronic Ballot Box Project (OEU or Œ)! This is a citizen initiative to provide to everyone all a simple tool to organize very secure[2] votes using the anyone computer ressources (No hardware/software dedicated machine). Removing need of voting office.

Open means that all tools and data are opensource but also that the system is fully distributed and replicated. There is no need of some autority that would keep more secret or more power than the basic citizen. The security model is based on state of the art cryptography research results but also on the principle that Internet is large enought to maintain billion of copies of non infected software. Just try to imagine how hard would be for some hacking organization to infect at the same time all the opensource compilers, (GCC for instance) in the world to make the operation '1+1' returning '3' on one very specific source code and at the same time keep result equal 2 for all other source code in the world!. You might think that only high skill math and software research people can understand all this. Well we will try to keep as much simple as possible, using many tutorials, examples, faq, tests. It is amazing how you can find today on the Net very well written RSA Tutorials. We would like to reach the same result with only the e-vote problem in mind. The main goal is that a 18 years old (age allowing vote for citizens in many countries) everage educated person could understand every details, after spending a minimum time study on it. We even think that teaching all the e-voting scheme could be integrated in the general education process. This can be support for Math, Computer Science and Democratic studies.

Hello world!

1 The three Œ Roles

Every one can have one or several roles, as Designer, Manager or Citizen. There is no rules and no constraint to candidate for these roles. It is only a matter of interest and skills. Nobody (individual or organization) shall discourage anyone to play these roles if he is interrested in.

1. Designer A designer is a person like me who design, improve, translate, fix the program given here (the same software generate the text you are currently reading). His name is listed in the text header. He or She me follow simple rules (Licence propagation, digest publishing,...)

2. Manager A Manager is a person or a group of persons (organization) that initiates a vote. The Manager does not modify the program. Usually, he is using the program of a designer who is not himself. A manager is in charge of initiating a vote, sending a unique signed ID to each citizen he would like to participate to a vote, let those citizen add their public key to the ring, revoque some IDs after a conflict and time stamp the urn to close the registration phase. He also close the vote by adding a final signed time stamp.

3. Citizen A citizen is an individual (can be a group if it make sense to give one ballot for a group) having generated locally a couple of public key, private key. He or she can register to a vote organized by a manager using the urn of a designer. A citizen willing to vote had to sign the unique id given by the manager. When the time stanp is set by the manager, he can vote. The voting task is technically adding a Linkable Ring Signature to his choice message. Any citizen can run validity checking tools on the urn, get voting results (temporary since not every registered citizen has not voted). Because the program and the data are all full text, every one, in particular citizer are invited to check validity and count with any external tools or even manually. Registered Citizen receved an e-mail from the namager before closing the vote

[1] You have to check first that this digest is the same given by running an external utility like *sha1sum* on the text source file and second, that this digest is the one given on the author Web page on Internet `https://github.com/oeu`.

[2] Secirity is a matter of probability, but in the range of existing ballot systems our system maybe reach current highest security level

2 Crypto animals for Œ !

- Digest
- Signature
- Time Stamp

3 FAQ

1. How do insure that the text I am reading is the right one ? This is simple, if your reading the text source file (oeu.py). Find a computer with Python interpreter, Run the script, it should not complain if this is an original. Use a LaTeX suite (pdflatex utility) on oeu.tex generated file to build a PDF formated document. If you are reading a Postscript, PDF or HTML file, you do have a full insurance that this is the right document. We strongly recommend you to download to source file. In both cases, to check the document origin and version, just check that the digest (in header PDF document or in last line of text file) is the one given on the WWWW `https://github.com/oeu/OpenElectronicUrn`).
2. What happen If someone change the content of this text ? Well the modified file has a different digest, so any list or ballots referencing this digest will be invalid.
3. How do you preserve anonymous vote ?
4. Putting several ballot in the ballot box is really not possible ? No, linkable ring signatures have a tag that is always the same if signed by the same individual. However, knowing the tag does not reveal the identity of the signer. One of the basic check on the ballot box is to find signatures with the same tags and qualify the signer public signature as invalid (for the current ballot).
5. If someone thiev my registration id, what can I do ?
6. If some smart guy crack the algorithm ? I am sure he or she will join us to fix the failure or improve the security of the system and will be qualified as the top designer. Until organization gives rewards, we will give him or her credit for nice contribution to democracy progress.
7. How do you merge ballot boxes ? This is explain in details somewehere else, but the idea is to merge only compatibles ballot boxes (referencing the same list, that reference the same oeu.py file) and to remove the identical parts.
8. Can you merge vote list ? With some constrainsts. During registration, list of public keys can be merged, but when the vote start, the list of public keys shall be fixed. Such list (digest) is referenced by the ballot box.
9. Why do you put every thing in text files? Because text file can be read with many and any text editor and it would be impossible for any cracker to infect all text editors on the planet at the same time without being discovered. Of course text files are not very compact, but it is not a big issue for voting. You can compress the text file (zip, tgz, bz2,...) before sending it on networks. Text files can be used for building documents (LaTex suite), write source code and store data...all we need for e-vote.
10. Why Python code ? It could have been any other language, but this one is used worldwide (impact on security), compact (text file size), modern (buildin high level idioms), easy to learn (very important for the project philosophy) and has nice libraries. In particular, the pycropto module provides tools for cryptographic algorithm.
11. Your system is very old fashon, we have now nice graphical interfaces ? Your are right, but we do not want to sacrify the security in making beautiful user interfaces. Graphical rendering is not requested for the moment (We did not currectly investigate the problem of showing candidate face pictures instead of their names or ids ...shall democratic countries assume that voters can read ?).

4 The little story

I started the project because I wanted to vote the preliminary election of the green party in France (EELV June 2011). This election was contucted on Internet by a private company (DEXUUUUUU). I found the process not secure at all. blabla...

`voters`
`ballots`