Œ: The Open Electronic Ballot Box

Laurent Fournier

June 18, 2011

Version: *0.1* State: *alpha* [SHA1 digest[1]: d3ae5ea0fa537d9dd6a03cee488dba3ab9e38225]

Abstract

Welcome to the Open Electronic Ballot Box Project ("OEU" or Œ)!

This is a citizen initiative to provide to everyone a simple tool to organize very secure[2] votes using the anyone computer ressources (no hardware/software dedicated machine) Using Œremove need of organizing vote offices, and all activites of traditional vote. Vote anywhere may increase participation. "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, power or knowledge 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 instances. Just try to imagine how hard would be for some malicious organization to infect at the same time all the opensource compilers, (`http://gcc.gnu.org` 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

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

(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

2 Crypto animals for Œ !

- Digest
- Signature
- Time Stamp

3 The process

Œtool is always improving but for using it, we must work on the same revision. So the first thing to do is to select the Œ, usually the last one and hoppely the more secure. For a large election, the digest of Œhas to be published on many place, and of course on the official web site of the organizer. This period of time has to be long enought to be sure that every one plan to use the same revision of Œ, can download it. The list (list.txt file) include an election identification, also published by the manager of the vote.

The manager has already or build a list of all participants of the vote, but this list is outside Œbecause it contents private informations; the full name, date of birth, address, phone nombers, e-mail, social security number....anything that can identify with high probability the participants. Means use for such list depend on the criticity of the vote and we can immagine that for non critical elections, a simple e-mail may be requested. Note that the security level of this task is the same for a traditional vote or for preparing an e-vote.

The manager then generate unique ids for each participants. It is better to generate a small hashed id instead of positive integers, because a malicious participant would have more difficulties to guess valid ids. Also manager is invited to use a secure canal for sending these ids. The security s not impacted, just the number of revocated ids to manage before closing the list. Each participant is invited to generate locally two keys RSA 2048 keys. Disconect the computer from the internet to be sure that nobody see what you are doing. Keep secret the private key (protected with at least with a pass phrase) and send the public key and and the signed id to the manager. There is multiple way to keep secure the private key, with strong authentification. Each citizen should be concerned with this constrainst. If you loose your private key, no one can help you to recover it. It is then better to generate another one and register again to the vote after requesting a revocation of the old public key to the manager. If the vote has started (registration closed), you cannot vote if you don't have private key copy ! If someone thief your private key and the passphrase you have written on a paper (don't do that), and has access to all authentifications devices you used during key generation, then he/she can vote at your place. More exactly, if you vote and the thief also vote, the result depend if its or not on the same box (before or after merging). Voting twice on the same box is simply rejected the second ballot, but if you use another box to vote (referencing the same election), then both votes are saved, but during the merging operation of the boxes, linked signature will raise a flag. If your two or several votes are for the same candidate, your ballot is saved, but any difference in candidate selection will produce an null ballot (do not confuse with white ballot, that is more a supplementary option in the selection list, if the election manager allows it).

After receiving all public ids, merging lists, the manager close the registration phase by signing a time stamped message on the list. All revocated ids (and public keys) are removed from the list. Then the list is fixed with the closure date. If anyone propose to new public id list to the manager, the former cannot use it and even if a sign it, only the first time stamping signature is valid. Time confidence depend on the selected time stamping organization on the Net. For critical vote, selected an old and secure time stamper is requested. When the list is closed and published on the Net, it is easy for every one to check that he/she bellons to that list. There is not your full name, but just your public id. Save your public key secret if you don't whant anybody except the manager to know that you are participating to this vote. This is a poor anonymity that you cannot achieve with classical vote...every body in the vote office can see you.

Let the fun start! Citizen can vote just after receiving the closed public id list. Empty ballot box referencing the list can be fill in. The header of the list reference all candidates for the election, with some id (small number). This id is well visible on the ballot box after people vote. It is even easy to count temporary ballot results for each box. That is why we suggest to send the box file directly to some merging server. From this server you can dowload to latest merged box with all the ballots mixed with yours. To some extend the merging server could find your ip

address and correlate with your identity if it has access to an ip geographical adress dictionnary. It also has to have the box file just before you vote to compare them and found for who you votes. To protect from you from this, you can use any other computer not at your home, you can use dynamic ip adresses or anonymous ip adress. Also it is recommended to add several votes (several individuals) on the same box before sending to the merge server. If you don't mind having the risk to show your vote to your related or small group, then share the same box. Again, if the box file on a computer if not sniffed (you can install a fresh linux install from scratch and share it with a group you trust), then the vote order is not revealed. When voting Œinsert the current vote at a random position. If you use an empty ballot box and send it to an individual, not a merging server, you cannot hide to him for who you vote. We can imagine that some people would prefer to go to some vote office providing computer ressources. There is less insurance that these computers are observers safe than that for your computer. The issue here is maintaining anonymity, but at large scale anonymous ballot are very probable. The result of the vote is not changed by the type of strategy (merging a lot of small boxes or less big boxes). Note that with traditional vote, the same problem occurs, because on some very small vote office, knowing the results gives some assumption of who votes for who, people knowing each other. With a difference is that the citizen can chose to participate to a small or a large group with e-vote when he/she is forced to move to a designated vote office in traditional vote.

4 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 ? This is the kernel of the system. linkable ring signature allows any member of a group of public keys to sign a message (here, the message contains the selection of the candidat) without saying anythin on her identity. Even on worse case any juge cannot prove that such a given signature comes from that individual. Everyone knows and can check that this is a member of the group who signed it. The linkable feature is some information that prove that the same individual signed two messages, still anonymouslly. This feature is used to refuse several vote of the same person.

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. someone pick my registration id, what can I do ? A rule is that every participants to a vote has to sign the id. If you did not received your id, contact the vote manager so see what is wrong, he will give you the same id again. If you received your id, but you think that someone else had rip it off. Try to sign first and your safe. If you try to sign after the thief, the urn will refuse registration. Contact the manager, prove your identity and then he will revocate your last id and give you a new one. A revocated id cannot participate to the vote.

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 ?).

12. What about hitorical/economical/social/phicological studies of e-vote ? We are reading carefully these studies and we welcome all suggestions and contribution in the group.

5 The little story

I started the project the day I register for voting one the preliminary election of the green party in France (EELV June 2011). This election was managed on Internet by a private company (Extelia). Opacity of the process and poor security level makes me furious, knowing that the company claims for very high security and certification level. I also found on Internet citizen associations against e-vote. I share their worries for the present (2011) but not for the future. I also found very interresting papers on 'group/ring signature' and how anonymity can be achieved. This is not my research field, so i have hard time to understand the details. It seems that starting 2008, some serious papers describes realistic e-vote shechmas, but I did not found implementations on the Net. I decided to start this project hopping to have a simple but complete implementation of e-vote.

```
voters
# PublicKeys [d3ae5ea0fa537d9dd6a03cee488dba3ab9e38225]# PublicKeys [d3ae5ea0fa537d9dd6a03cee488dba3ab9e38
```