An eVoting System Based on Ring Signatures

J. L. Tornos, J. L. Salazar, J. J. Piles, J. Saldana, L. Casadesus, J. Ruiz-Mas y J. Fernández-Navajas

Dept. of Communications and Electronic Engineering, University of Zaragoza
Ada Byron Building, 50018, Zaragoza (Spain)

Tel: +34 976 76 2698 E-mail: {jltornos, jsalazar, jpiles, jsaldana, luis.casadesus, jruiz, navajas}@unizar.es

Received: April 28, 2014 Accepted: June 15, 2014 Published: July 30, 2014

DOI: 10.5296/npa.v6i2.5390 URL: http://dx.doi.org/10.5296/npa. v6i2.5390

Abstract

The increase of Internet penetration in the last years is boosting the popularity of eVoting systems. However, in order to have a security level similar to that of an in-person voting, a series of security requisites have to be accomplished. This article describes a secure eVoting protocol based on ring signatures. The implementation details and the different modules of a voting platform including this signature protocol are presented. In addition, a special characteristic has been included in the voting protocol: during the signature process a parameter called "linking tag" is generated, able to identify the different votes sent by a single voter during a voting process. This characteristic makes it interesting in e-Cognocracy and Quality of Experience evaluation scenarios.

Keywords: eVoting, Ring Signatures, QoE

1. Introduction

Since the middle of the last Century, different mechanisms have been incorporated to improve voting processes, and the subsequent counting of votes. The first mechanisms introduced were based on mechanical methods, as e.g. punched cards, designed with the main aim of making it easier the counting process. The same happened with the methods for optical reading of the votes. Some years later, Direct Recording Electronic (DRE) systems [1] were introduced in order to permit the direct registration of the vote, thus avoiding the need for counting the votes. These mechanisms were located between polling stations and the voters, who identified themselves before the staff in charge of the voting, and used them afterwards.

With the rise and the popularization of the Internet, different methods are being employed in order to get the opinion of the users. Surveys are nowadays popular in online newspapers, forums and social networks, but they are only used with informative purposes, or for polling the opinion of the readers about a topic. These methods for gathering information lack the security mechanisms which are required in a real voting process, be it online or in-person. They usually do not perform a control of the users who have already participated (so a single user is able to participate a number of times), or sometimes they include very basic methods for avoiding this, as e.g. using an e-mail account as the user identity when participating.

A number of requirements have to be accomplished in order to transform a polling system into a real voting system, with the objective of making its security level comparable, or even higher, to those based on in-person voting [2]. For this aim, different eVoting systems have been developed, based on one of these four protocols: mix-nets [3 - 5], blind signatures [6, 7], homomorphic encryption [8, 9] and ring signatures [10]. Each of them uses a different cryptographic protocol or a number of secure servers, in order to secure the vote and to warrant the anonymity of the voter at the same time.

The first advantage [11] of an eVoting system is that it allows the citizens to vote from home or their office; furthermore, if the system is adapted for smartphones and tablets, it will even be possible to vote from any other place with Internet connectivity. Some other advantages can be found: people who are outside their municipality may also participate; young people will become more interested on participating, since they are used to these technologies; the time required for vote count gets significantly reduced; and the temporal restrictions that appear in a normal voting process can be relaxed.

This article presents the design, implementation and tests of a secure eVoting system using ring signatures [10] so as to grant the anonymity of the voters, based on the protocol described in [12]. The whole system has been implemented with free and open-source software (FOSS). The system consists of five interconnected blocks:

- The *user administration* module is in charge of users' management: user registration and users' data modification; loading of certificates and coordination with accepted Certification Authorities (CAs).
- The *voting administration* module manages the different voting processes stored in the

server. This module is the responsible of the census of users which can participate in the voting, the number of rounds, the duration of the voting process, the ballot box, etc.

- The *connection and verification* module connects the voter and the *administration* module, selecting the voting to be accessed by the user, and verifying the conditions, so as to permit or deny the access.
- The *client* module is to be utilized at the user's device. It has been implemented both in a web page and in a smartphone-specific app, thus allowing a higher mobility. It is in charge of the cryptographic calculus required for the emission of the vote, and for its sending, through the *connection* module, to the "electronic ballot box".
- The *key generation* module performs the key and certificate calculations required to identify the user.

The remainder of the article is organized as follows: In Section we present the related work. Section III describes the different protocols in which secure eVoting systems are based. In Section IV we explain in detail the proposed protocol, describing each of its blocks. In Section V the implementation of the protocol and the developed proof of concept are explained. The paper ends with the conclusions and the description of future work lines.

2. Related Work

In the last years, electronic and secure voting systems have increased their importance. They are no longer considered as simple methods for doing surveys and polls in social networks, but they are becoming a solid alternative to in-person voting systems. Some examples of this tendency can be found in European countries as Estonia, Switzerland and Norway. In the first case, an eVoting system has been employed in Estonia since 2005, for both local and national elections. The use of this system has increased his penetration [13], which supposed 24 percent of the total votes cast in the elections in 2011. In Switzerland [14], eVoting has been used for more than ten years. At the beginning, the number of citizens able to use the system was limited to 20 percent in a canton, and to 10 percent in the whole country. Nowadays, the penetration rate is 30 percent in a single canton. The experience with eVoting in Norway [15] started during the local government elections in 2011 and 2013, and it was also used in the parliamentary elections [16].

In [17] a general framework for the design and implementation of eVoting systems is defined. They also present an implementation which enables the participation by means of a web browser or SMS, using a login/password user identification. In [18] an eVoting system was developed, based in mobile devices, allowing both offline and online identification of the voter, using the credentials associated to the SIM of the used phone. The participation of the user is enabled by means of symmetric keys and an SMS-based system.

In [19] cloud computing is integrated into the eVoting system so as to increase the cooperation between different institutions and the participation of citizens. The cloud-based system allows cost reduction and increases flexibility at the same time. However, the authors remark that the users must feel the security, so a really trustworthy method has to be designed. The article presents a framework including the desired characteristics and the theoretical challenges to be issued, but it does not develop an implementation.

3. Secure eVoting systems

In order to be considered as secure, an eVoting protocol must accomplish a set of basic requirements [2]:

- **Privacy:** votes must be kept secret.
- **Completeness:** all the valid votes should be counted.
- **Soundness:** non-valid votes should not account in the final result.
- **Unreusability:** a voter can only vote one time.
- **Elegibility:** only people with the right to vote should be able to vote.
- **Fairness:** No external factors can affect the voting process.

In addition, a series of extended requirements can be considered in order to increase the attractiveness of an eVoting system:

- **Robustness:** The voting should be carried out in spite of partial failures of the system.
- **Universal verifiability:** Once the result of the election has been published, anyone should be able to verify that the tally is correct and the votes were correctly cast.
- **Receipt-freeness:** a voter should not be able to receive or construct a receipt or token revealing the content of his/her vote.
- **Incoercibility:** A voter should not be coerced at the moment of voting.

The different eVoting systems are built with the aim of at least accomplishing the basic requisites described above. The strategy for achieving this objective may vary between them. Therefore according to the different strategies they employ, a secure eVoting system can be classified in one of the next groups:

- *Mix-nets* [3]: these eVoting systems use secure servers (known as *mixes),* that receive as an input a set of votes, and generate as an output the same set of votes, but disordered. Two different methods for cyphering the voting information can be employed: the information can go cyphered through the whole set of *mixes* to traverse (known as *cascade* or *series of mixes),* or a re-encryption system can be used, as proposed in [20]. Three conditions have to be accomplished in order to grant a correct *mix-net*-based system [4]:
    - Operate correctly: the output should correspond to a permutation of the input.

  - Privacy: none of the inputs can be related with an output of the *mix*.

  - Robustness: every *mix* should provide a "proof or strong evidence" that the output corresponds to the input.

- Blind signatures [6]: The clearest example when explaining this method is carbon copy. We can imagine the voter inserting his/her vote in a carbon envelope. He gives the envelope to the certification authority who, once recognized him/her as a valid participant, signs the envelope. Next, the voter extracts the vote, which has been signed by the authority thanks to the carbon envelope. So the voter can deposit or send the vote to the correspondent ballot box. If we translate the concept to the digital world, the voter sends an obfuscated vote to the certification authority, who signs it after verifying that it is valid for that voting process. Once the voter has received the signed vote, he undoes the obfuscation and sends the signed vote, with the signature attached, to the ballot box. An electronic voting system employing this method is the one presented in [21].

- Homomorphic encryption [8]: It is used for voting based on counting, as e.g. referendums. It consists of operating with the cyphered votes, in order to obtain a cyphered final result of the election, which is then decrypted and made public. This allows the identification of the voter when he sends his cyphered vote, taking into account that it will never be deciphered. Homomorphic cyphering is limited to basic operations as addition and subtraction, since the cyphering function has to be a homomorphism, accomplishing $E(x + y) = E(x) + E(y)$. This is why it is especially adequate for referendums.

- Ring signatures [10]: They are used to identify a user as a member of a group, without the need for revealing his identity. In addition, the obtained anonymity when using these signatures is irrevocable; and these signatures are also *spontaneous*, meaning that a manager is not required for coordinating a group of voters and the keys they use, as it happens in group signatures [22]. A further step was taken [23] when ring signatures could link a number of different votes of a single voter without revealing his identity. Thanks to this feature, a voter can vote a number of times during the valid period, and the only valid vote will be the one established in the voting conditions (usually the last vote sent will be the valid one).

Other important characteristic of an eVoting Systems is the number of Trusted Third Parties (TTP) it requires. First of all, the presence of a CA is always required, in order to provide the users with the keys necessary to vote. However, additional requisites can demand a higher number of TTPs. For example, in the systems using blind signatures, at least one more TTP is necessary, since the authority signing the vote should not be the same one that collects them. Or in an ideal mix-net, each of the mixes should be independent, and should therefore be considered as an independent TTP. In the protocols implementing ring signatures, only one TTP is required, since the users are identified as belonging to a group and not individually.

Fig. 1 shows a voting protocol using blind signatures with two TTPs. Two different communication phases can be appreciated:

- Identification and registry with the Registration TTP (RTTP). The user downloads the information from the server of the voting provider. Once he has decided the content of his vote, he identifies himself against the RTTP, and sends his obfuscated vote. The RTTP verifies his identity, and that he has not already participated; if everything is correct, it signs the obfuscated vote and sends it back to the user.

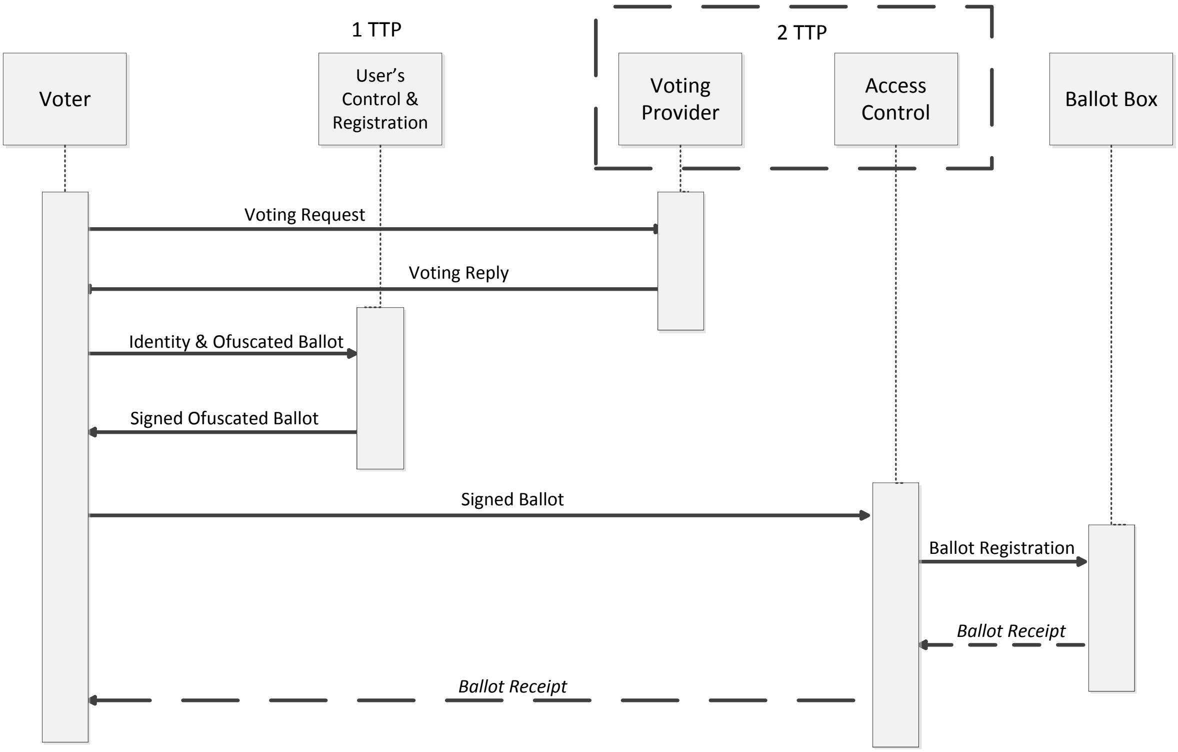


Figure 1. eVoting protocol using two TTPs

- Communication with the second TTP. In this case, it is the Authenticated TTP, (ATTP). The voter undoes the obfuscation of his vote, and sends it to the ATTP, with the RTTP signature attached. The ATTP verifies that the attached signature corresponds with the RTTP signature. If everything is correct, the vote is sent to the ballot box. The ballot may send a receipt to the user, or the ATTP may be in charge of sending an identifier to the voter.

Some eVoting systems (e.g. those using ring signatures), do not require the registration of the voter, so the second TTP it is not necessary (Fig. 2). Another advantage, in addition to the reduction of the number of TTPs involved, is that all the parties (voting provider, access control and ballot box) can be placed in a single server, thus reducing the hardware requirements of the voting.

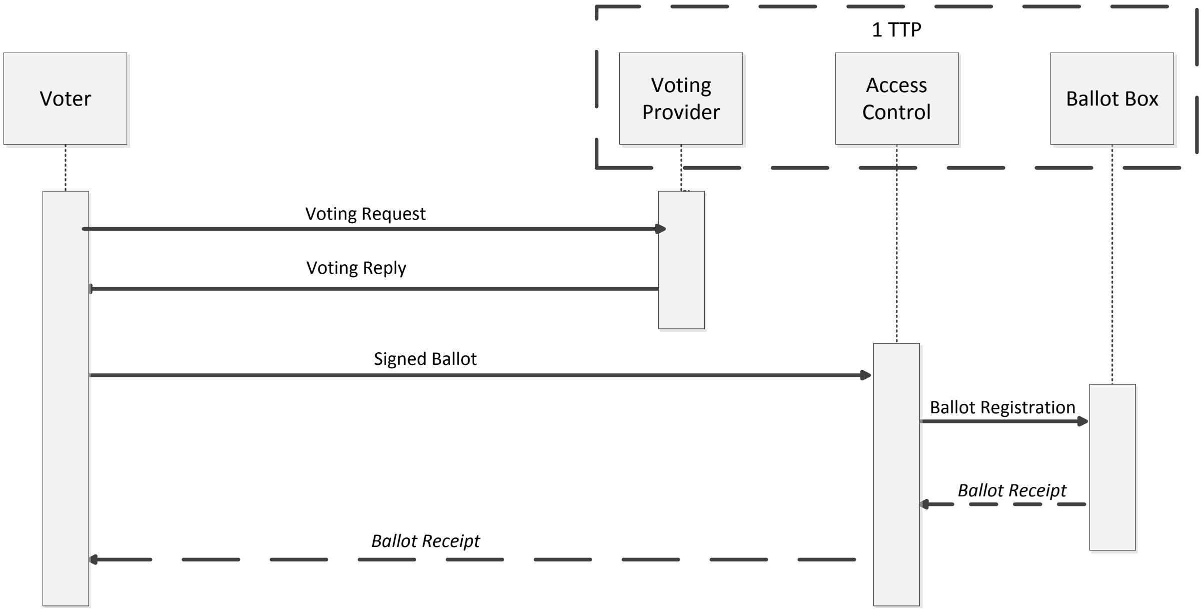


Figure 2. eVoting protocol using one TTP

4. Design of an eVoting system based on ring signatures

The eVoting system we have developed in this paper is based on ring signatures protocol, and in the security descriptions presented in [12]. It has to be modular, in order to make it more flexible and easily adaptable to different modifications on the security requirements, the administration process, or the users' needs. As a result, the system is composed of five different blocks. In addition, the user should incorporate the signature module into the device he will use on the voting process, since this module will be in charge of the cryptographic operations associated to the signature of the vote.

The *key generation* module and the *client* module are at the user's side. The first one develops the tasks related to user's voting setup (calculate the private key, create the Certificate Signing Request and the PKCS12 [24], which will be used later by the client module). In the server we can find three modules: *user administration* (in charge if registrations, unregistrations, certificate update and users management in general); *voting administration* module (enunciate the voting, parameters and the census); *connection and verification* module, in charge of connecting the user's device with the administrative module; it also performs some middle verifications in order to grant a correct performance of the system.

4.1. Client modules

The client has to install two different modules in his device: *key generation* and *client* module.

4.1.1. Key generation module

One of the basic principles of asymmetric cryptography is that the private key is only known by its owner. In order to accomplish this rule, people participating in voting processes must create their own keys, to be signed by a CA, thus obtaining the certificate to be used when voting. The keys used in the proposed eVoting system should accomplish these characteristics:

- All the math operations are modulo n, where $n = pq = (2p' + 1)(2q' + 1)$ of λ bits, and with *p, q, p', q'* prime numbers.
- The private keys (e_1, e_2) are two different prime numbers belonging to the interval $(2^l - 2^\mu, 2^l + 2^\mu)$, where l and μ are security parameters of the protocol.
- The public key is$(2e_1e_2 + 1)$, which is a prime number.

Due to the specific characteristics of these keys, and taking into account that a standard certificate for storing them is not defined, we have used a standard RSA [25] certificate to store the public/private key pair. The public key will be stored in the same way than a normal RSA key, being n = pq the modulo, and storing the value $2e_1e_2 + 1$ as the public key.

For the storage of the private key, four parameters are required (one more parameter with respect to standard RSA keys): in the modulo field we store $n = pq$; in the public exponent field, the value of $2e_1e_2 + 1$ is stored; the value of e1 is stored in the private exponent field, and the field prime1 is used to store e2.

Once the user obtains the public key certificate, correctly signed by the AC, he sends it to the *user administration* module, to be stored as the user identifier. The user stores the certificate containing his public key in the file PKCS12, which also stores his private key, protected by a password.

4.1.2. Client module

This module performs the cryptographic calculations required by the voting protocol. Once the voting parameters and the public keys of all the voters have been received, it signs the votes with its private key. Public keys values will be used for the calculation of one of the required parameters for carrying out the voting. The user's private key is protected in the PKCS12 container, so each time a signature is to be performed, the password will be required to the user.

4.2. Server modules

Three different modules are required at the server side. One is in charge of user management, other controls the voting processes, and the last one performs the connections and verifications required between the server and the client.

4.2.1. User administration module

The first module to be installed in the voting server is in charge of the management of the users. It maintains the list of valid users, who should register in the system uploading the digital certificate including their public key signed by a trusted CA. In addition, it verifies that the certificates are valid, that the period of validity includes the moment of the voting, and it removes the certificates that may have been compromised. The census to be used on each voting will be obtained from this list. Finally, the module also controls the permissions assigned to the users that create or modify voting processes.

4.2.2. Voting administration module

The *voting administration* module is composed of different parts, interfacing in different ways: first, it stores the census including the users allowed to participate in the voting. These users should have previously registered in the *user administration* module. In addition, the voting administrator defines the time period when the voting will be open, and it has to get the parameters required for the cryptographic signature protocol. This module also permits to classify the users into different groups, and each group may have a different weight, which is useful for performing weighted voting. In addition, the administrator is allowed to create a number of voting rounds, which can be required in certain voting processes where a minimum consensus has to be reached.

Thanks to the characteristics of the voting system described in [12], the signature of the user’s vote is accompanied by a value called *linking tag*, useful for linking together the votes from the same user. This permits a user control avoiding that a single user can vote more than a single time, or allowing a number of subsequent votes of a user, being the last vote the valid one.

4.2.3. Connection and verification module

It receives the participation requests from the client, and transfers the information related to the selected voting (question, parameters, configuration, participants, etc.). In addition, it will also receive the vote and process it. Thus, if the signature attached to the vote is correct, it is sent to the ballot box.

5. Implementation of the eVoting platform

As said in the introduction, the eVoting platform has been implemented with Free Open Source Software (FOSS) and in a modular way. Different programming languages and tools as MySQL, Java, JavaScript, JSP, Apache Tomcat, Android and Firefox have been employed. Two different versions of the user’s module have been implemented: one for Firefox browser, to be used in netbooks, laptops and desktops, and a specific application for Android smartphones and tablets.

5.1. Server

Two different profiles have been created for accessing the server, one related to user management and the other for voting management. The user management creates and manages the users’ profiles. Two different groups of users are defined: first, the users able to administrate and to maintain each voting developed in the platform. In addition, they will manage the keys employed when signing the votes to be sent to the ballot box. Thus, even if a single eVoting server is shared, and the ballot boxes corresponding to different voting processes are stored in it, only the administrator of each voting will be able to know the final result. The second group to be managed corresponds to the potential voters included in the platform, i.e. a record including all the eligible users, according to the criteria defined by each voting administrator.

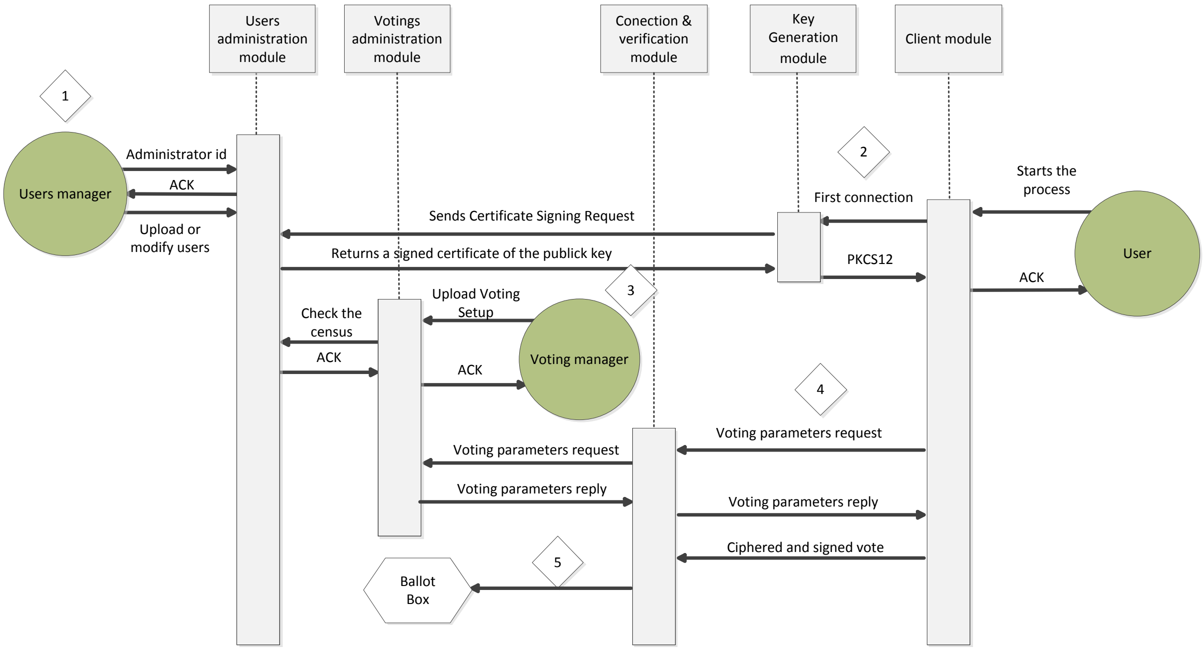


Figure 3. Interaction diagram between the different modules

The voting module has been designed with the aim of making the process of defining a voting easy. The administrator should introduce the next parameters: name of the voting or query, questions to answer, security parameters, number of rounds and the census of the users who may participate. The administrator will also be able to define different groups, each of them with a different weight if necessary. The question will be edited in HTML, and the only requirement from the platform is that a *getBallot()* function exists, able to take the vote in a string. This string will be signed by the user so as to grant the security of the process.

The *connection and verification* module has been implemented to work in a transparent way for both voting administrators and users. Since different voting processes from different administrators will be simultaneously stored in the platform, the parameters of each voting may be different, so the connection module is the one in charge of transmitting the parameters required for the correct performance of the voting. In addition, this module will verify the votes of the users and their signatures, in order to send to the ballot box only those whose signature is correct. By means of the *linking tag* described in [12], which is also a part of the signature, the module performs the required operations in order to know if the vote belongs to a person who has already voted. Depending on the policy established by the administrator, a person may vote more than a single time in a round. If this is not allowed, this module will reject the vote, and it will send a message reporting that he has already voted and that he cannot vote again. Other policies may allow a number of votes from a single voter, as explained before.

The interaction protocol between the different modules is illustrated in Fig. 3. Number 1: The user information (name, identification …) is first uploaded. Number 2: the user key is generated and some communications are needed to create the PKCS12 file containing the public certificate of the user. Number 3: when the administrator of a voting uploads the parameters to the platform, he also uploads the census with the participants. Then, the voting administrator has to connect to the *user administration* module in order to verify that all the users included in the census are registered in the platform. Thus, a user not registered in the platform will not be able to be in the census of any voting. Number 4: Once the voting has been correctly loaded, it will be visible to the users during the valid voting period. The participants will connect to the server and select the voting in which they want to participate. They will then download the parameters, perform the voting, the survey or the evaluation, sign the vote with the public key of the ballot box, and then they will send this vote to the server, also attaching its signature. Once the server has received the vote and its signature, it verifies the signature and, if correct, sends it to the ballot box, number 5.

The ballot box has been included in the same server. It is implemented as a database table, only accessible to the *connection and verification* module, where the already verified votes and their attached signatures, will be stored. Once the voting has finished, the voting administrators may (and should) verify the signature again, decipher the votes and proceed with the final accounting. The table structure of the database is shown in Fig. 4.

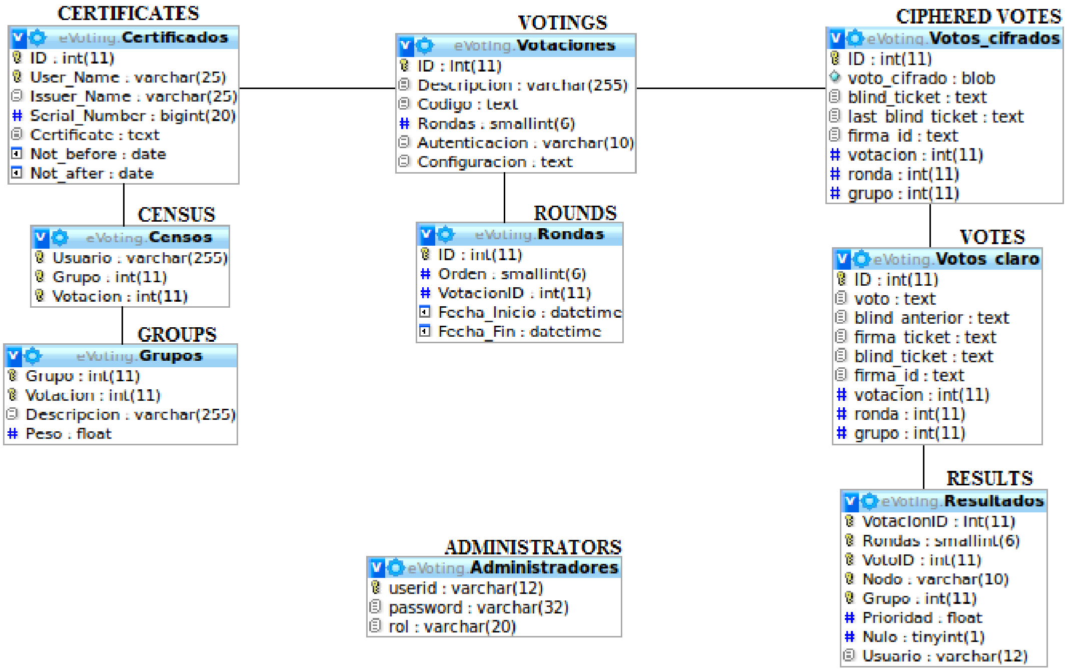


Figure 4. Database tables

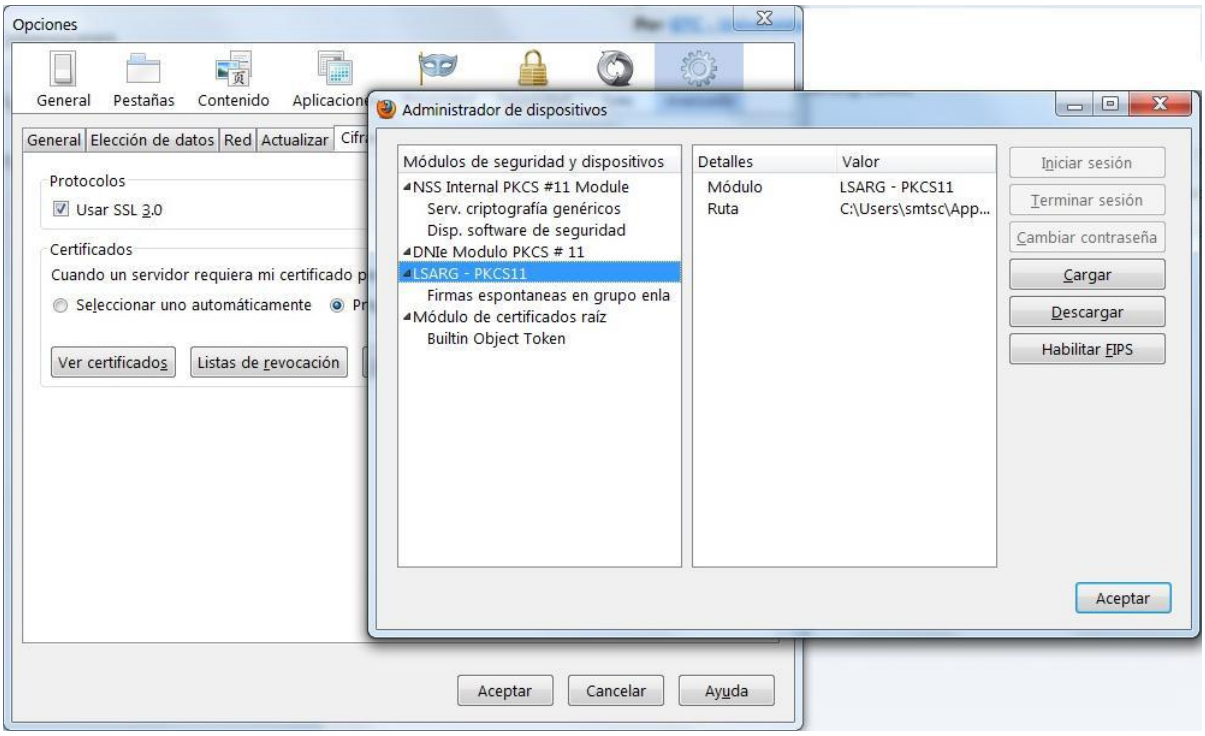
5.2. Client

As explained in 3.1., two different options are available for the client: one for PCs and laptops, based on Firefox browser, and another for smartphones and tablets using Android.

5.2.1. Firefox client

The client part has been developed as an extension to the browser. The decision of doing it this way was made in order to making its use easier for the users. This extension performs all the cryptographic operations providing security to the platform.

Once the browser extension has been installed, the user will have to report the location of his certificate and his public key, stored in a PKCS12. This extension will be selected as the one in charge of managing the interaction of the browser with the digital signature systems based on ring signatures (Fig.5). Each time the user participates in a survey, and evaluation or a voting process, he will have to introduce the password that protects his private key. The browser extension will take his vote, sign it with the public key of the ballot box and send it to the eVoting server. The extension runs correctly both in Windows (XP or latter) and Linux operating systems.



Opciones

Protocolos

☑ Usar SSL 3.0

Certificados

Cuando un servidor requiera mi certificado p

○ Seleccionar uno automáticamente ● Pr

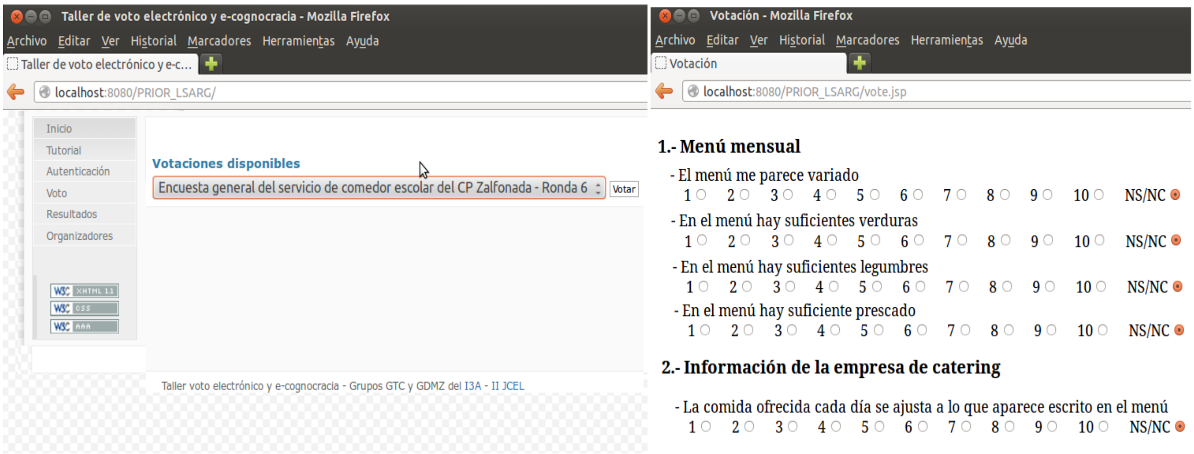
Administrador de dispositivos

Módulos de seguridad y dispositivos
- NSS Internal PKCS #11 Module
  - Serv. criptografía genéricos
  - Disp. software de seguridad
- DNIe Modulo PKCS # 11
- LSARG - PKCS11
  - Firmas espontaneas en grupo enla
- Módulo de certificados raíz
  - Builtin Object Token

Detalles	Valor
Módulo	LSARG - PKCS11
Ruta	C:\Users\smtsc\App...

Figure 5. Firefox extension installed to managed ring signatures

The user interface has two stages: first, a list of the available voting processes is presented, and the user can select one (Fig. 6a). Once selected, the voting page (Fig. 6b) is presented. This page can be adapted, according to the specific requirements of the voting process to be performed: it can be a simple form, or it can even include multimedia content when required, e.g., a video which QoE is to be rated by the user.


Taller de voto electrónico y e-cognocracia - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
Taller de voto electrónico y e-c...
localhost:8080/PRIOR_LSARG/
Inicio
Tutorial
Autenticación
Voto
Resultados
Organizadores
W3C XHTML 1.1
W3C CSS
W3C AAA
Votación - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
Votación
localhost:8080/PRIOR_LSARG/vote.jsp

Votaciones disponibles

Encuesta general del servicio de comedor escolar del CP Zalfonada - Ronda 6 [Votar]

Taller voto electrónico y e-cognocracia - Grupos GTC y GDMZ del I3A - II JCEL

6a

1.- Menú mensual

- El menú me parece variado
1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ○ NS/NC ●
- En el menú hay suficientes verduras
1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ○ NS/NC ●
- En el menú hay suficientes legumbres
1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ○ NS/NC ●
- En el menú hay suficiente prescado
1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ○ NS/NC ●

2.- Información de la empresa de catering

- La comida ofrecida cada día se ajusta a lo que aparece escrito en el menú
1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ○ NS/NC ●

6b

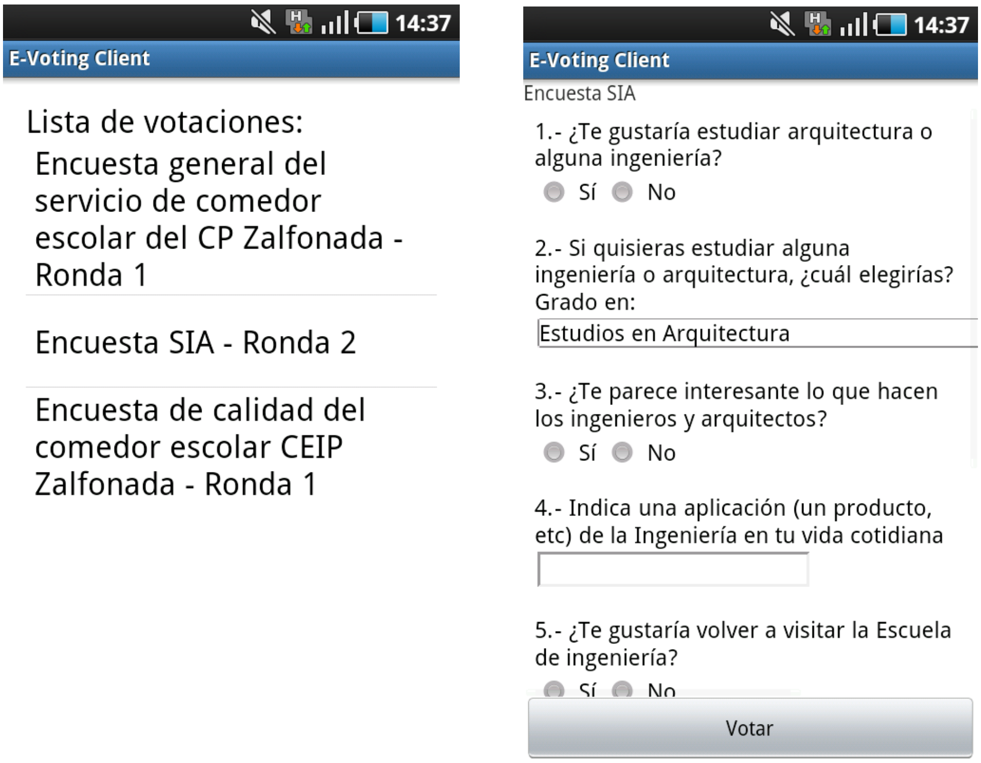
Figure 6. Firefox extension installed to managed ring signatures

5.2.2. Android client

On behalf of a greater versatility, the client system has also been developed for Android. An app oriented to smartphones and tablets has been built, providing the same functionalities than the Firefox extension, but with some adaptations for improving its usability. When the app is started, the available voting processes are shown, without the need of introducing the address of the server, Fig. 7a. The user then selects one of the options and proceeds with the voting. The methods for introducing data have been adapted to its use in mobile platforms, using radio buttons, check lists and lists in order to make it easier for the user to introduce the data.

Taking into account that a terminal can be used by a number of people, an option for selecting the user has been added. The app will display the username, and it will ask if that is the correct one or if another username is going to be employed.

Once the user has introduced his vote, he will have to introduce the password that protects his private key. If correct, the vote will first be cipher with the public key of the ballot box, and then signed with the private key of the user.



Lista de votaciones:

Encuesta general del servicio de comedor escolar del CP Zalfonada - Ronda 1

Encuesta SIA - Ronda 2

Encuesta de calidad del comedor escolar CEIP Zalfonada - Ronda 1

Encuesta SIA

1.- ¿Te gustaría estudiar arquitectura o alguna ingeniería?
○ Sí ○ No

2.- Si quisieras estudiar alguna ingeniería o arquitectura, ¿cuál elegirías? Grado en:
Estudios en Arquitectura

3.- ¿Te parece interesante lo que hacen los ingenieros y arquitectos?
○ Sí ○ No

4.- Indica una aplicación (un producto, etc) de la Ingeniería en tu vida cotidiana

5.- ¿Te gustaría volver a visitar la Escuela de ingeniería?
○ Sí ○ No

a b

Figure 7. Android app screen captures

5.3. *Proof of concept*

The eVoting platform has been tested in different moments: in the first experience [26], a proof of concept was presented using the Firefox client version. It was developed in a lab environment, and it consisted of a single round voting. The question was related with the web page of our Department, and it allowed different options. In order to increase the variability of the users, they were asked to employ different Operating Systems. The browser extension worked properly in Windows XP, Windows 7 and Ubuntu 10.04 LTS. Some users voted a number of times in different moments while the voting was open, thus checking the correct behavior of the *linking tag* associated to the vote.

Once the tests in a controlled environment had been deployed, new tests were developed with a higher number of participants and rounds in a more open environment. For that aim, the participation of users attending a conference at the University was requested. This time the Android app was used, since it made it easier the participation, since people carried their smartphones. The app was uploaded to Google Play app repository [27]. A tablet with the app was available for those not having a smartphone. The questions were adapted to the mobile platform, Fig. 7b, reducing the open fields and increasing the usability. A second round was also enabled in order to get the opinion of the users after the conference, although the number of participants was smaller.

5.4. Aplications

It can be said that the eVoting platform is currently usable. Thanks to the added functionality using the *linking tag*, a series of interesting applications are enabled. As described in [26], it can also be used for QoE evaluation purposes, when users have to rate the quality experienced when watching a video, running an application, etc. The user's opinion can be gathered, also knowing its evolution according to different changes as e.g. the hour of the day, or the day of the week. This is granted without compromising the anonymity of the user and without reducing the security.

Other potential application of this platform is e-cognocracy [28]. This kind of participative governance relies on ICT so as to achieve an active participation of the citizens on decision-making processes. This system requires to follow the opinions and to create weighted groups.

6. Conclusions and future work

eVoting systems are getting an increasing popularity, and their use is getting more importance, even in parliament elections of certain countries. It is important to distinguish between secure and non-secure methods for gathering information. Secure ones are comparable to those performed in-person. eVoting systems have a series of characteristics making the participation easier, since they provide more flexibility in the participation requisites, e.g. to be able to vote from a connected device, the enlargement of the time the voting is available (without incurring in additional costs), and to make the final count of votes easier.

This paper has summarized the requirements that an eVoting protocol must accomplish in order to be considered secure. An implementation based on ring signatures has been developed, showing its ability to grant the anonymity of the users by means of an identification indicating if a user is a member of a group, without revealing his identity. The platform has been totally implemented using FOSS, with two options, in order to allow an easier participation: web browser and smartphone.

As a special feature, the eVoting system includes a parameter called linking tag, which remains constant during the whole voting process for a user. Thus, the evolution of the opinion of a user can be followed, since his different votes can be linked together. This makes this platform suitable for e-Cognocracy and for QoE evaluations.

As future lines of research, more options for the end user will be added, and the mobile app will also be implemented in iOS. In addition, new tests including a higher number of users will be considered in order to check the scalability of the platform.

Acknowledgement

This work has been partially financed by CPUFLIPI Project (MICINN TIN2010-17298) of Spanish Government and also by Cátedra Telefónica-Universidad de Zaragoza.

References

[1] Kohno, T., Stubblefield, A., Rubin, A.D. and Wallach, D.S. "Analysis of an electronic voting system". In: IEEE Symposium on Security and Privacy. IEEE Computer Society Press, Los Alamitos (2004). http://dx.doi.org/10.1109/SECPRI.2004.1301313.

[2] Lee, B and Kim, K. "Receipt-free electronic voting scheme with a tamper resistant randomizer". In Proceedings of the 5th international conference on Information security and cryptology (ICISC'02), pp. 389 - 406. http://dx.doi.org/10.1007/3-540-36552-4_27

[3] Chaum, D.L. "Untraceable electronic mail, return addresses, and digital pseudonyms". Commun. ACM,24(2):84–90, 1981 http://dx.doi.org/10.1145/358549.358563

[4] Jakobsson, M., Juels,A. and Rivest, R. L. "Making mix nets robust for electronic voting by randomized partial checking". In Proceedings of the 11th USENIX Security Symposium (USENIX '02), 2002, pp 339 – 353

[5] Michels, M. and Horster, P. "Some remarks on a receipt-free and universally verifiable mix-type voting scheme". In ASIACRYPT '94, Pp. 125–132. Springer-Verlag, LNCS 1163, 1996.

[6] Chaum, D. "Blind signatures for untraceable payments". In Advances in Cryptology – Crypto '82, pages 199–203. Plenum Press, 1983. http://dx.doi.org/10.1007/978-1-4757-0602-4_18

[7] Xia, Z. and Schneider, S.: "A New Receipt-Free E-Voting Scheme Based on Blind Signature" In: WOTE: Workshop on Trustworthy Elections, pp. 14 – 28 (2006)

[8] Johnson, R., Molnar, D., Song, D., Wagner, D. "Homomorphic signature schemes". In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 244–262. Springer, Heidelberg (2002). http://dx.doi.org/10.1007/3-540-45760-7_17

[9] Acquisti, A: "Receipt-free homomorphic elections and write-in ballots." Cryptology ePrint Archive, Report 2004/105 (2004).

[10] Rivest, R.L., Shamir, A. and Tauman, Y. "How to leak a secret. "in Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01) , pp. 552-565 (2001).

[11] Schaupp, L.C. and Carter, L. "E-voting: from apathy to adoption" Journal of Enterprise Information Management, 18 (5/6) (2005), pp. 586–601

[12] Salazar, J.L., Piles, J., Ruiz, J., and Moreno-Jiménez, J.M. "Security approaches in e-cognocracy". Computer Standards and Interfaces, 32 (5–6), 2010, pp. 256–265. http://dx.doi.org/10.1016/j.csi.2010.01.004

[13] Madise, U. and Vinkel, P. "Constitutionality of Remote Internet Voting: The Estonian Perspective". Juridica Internationa XVIII. pp. 4-16 (2011) http://www.juridicainternational.eu/public/pdf/ji_2011_1_4.pdf (accessed 18.06.2014).

[14] Gerlach, J. and Gasser, U. (2009): Three Case Studies from Switzerland: E-Voting. March 2009, Berkman Center Research Publication No. 2009-03.1 http://cyber.law.harvard.edu/sites/cyber.law.harvard.edu/files/Gerlach-Gasser_SwissCases_Evoting.pdf (accessed 18.06.2014).

[15] Stenerud, I.S.G, and Christian B. "When Reality Comes Knocking: Norwegian Experiences with Verifiable Electronic Voting." 5th International Workshop on Electronic Voting, EVOTE 2012, Bregenz, Austria, pp. 21–33 (2012)

[16] 2013 elections http://www.regjeringen.no/en/dep/kmd/prosjekter/e-vote-trial/about-the-e-vote-project.html?id=597724 (accessed 18.06.2014).

[17] Qadah, G.Z. and Taha,R. "Electronic voting systems: requirements, design, and implementation" Computer Standards & Interfaces, 29 (3) (2007), pp. 376–386

[18] Ullah, M., Umar, A.I., Amin, N.; Nizamuddin, "An efficient and secure mobile phone voting system," Digital Information Management (ICDIM), 2013 Eighth International Conference on , vol., no., pp.332,336, 10-12 Sept. 2013 http://dx.doi.org/10.1109/ICDIM.2013.6693989

[19] Zissis, D., Lekkas, D. "Securing e-Government and e-Voting with an open cloud computing architecture", Government Information Quarterly, Volume 28, Issue 2, April 2011, Pages 239-251, ISSN 0740-624X, http://dx.doi.org/10.1016/j.giq.2010.05.010.

[20] Boneh, D. and Golle, P. "Almost entirely correct mixing with applications to voting" In ACM Conference on Computer and Communications Security , 2002 , pp. 68-77 .

[21] Ray, I., and Narasimhamurthi, N. (2001)." An anonymous electronic voting protocol for voting over the internet". Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, WECWIS 2001,. pp. 188-190. (2001) http://dx.doi.org/10.1109/WECWIS.2001.933922

[22] Chaum, D. and Van Heyst, E. "Group signatures". In Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques (EUROCRYPT'91), 1991, pp. 257 - 265. (1991) http://dx.doi.org/10.1007/3-540-46416-6_22

[23] Tsang, P.P. and Wei, V. K.: "Short linkable ring signatures for e-voting, e-cash and attestation". In Proceedings of the First international conference on Information Security Practice and Experience (ISPEC'05), 2005, pp. 48 - 60. http://dx.doi.org/10.1007/978-3-540-31979-5_5

[24] PKCS #12: Personal Information Exchange Syntax Standard. http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs12-personal-information-exchange-syntax-standard.htm (accessed 18.06.2014).

[25] Rivest R. L., Shamir, A. and Adleman, L. "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, (v.21 n.2), 1978, pp.120-126. http://dx.doi.org/10.1145/359340.359342

[26] Tornos, J.L, Salazar, J.L. and Piles, J. J. "An eVoting platform for QoE evaluation", IEEE International Symposium on Integrated Network Management (IM 2013). Ghent, Belgium, May 2013.

[27] Google Play https://play.google.com (accessed 218.06.2014).

[28] Salazar, J.L. , Piles, J. J., Ruiz, J. and Moreno-Jiménez J.M. "E-cognocracy and its voting process", Computer Standards and Interfaces, 2008, pp 124–131. http://dx.doi.org/10.1016/j.csi.2007.08.017

Copyright Disclaimer