An eVoting platform for QoE evaluation

José Luis Tornos, José Luis Salazar and Joan Josep Piles
Communication Technologies Group
University of Zaragoza
{jltornos, jsalazar, jpiles}@unizar.es

Abstract— **Electronic voting systems have long been used as a way of collecting information. In many of them security is not taken into account; however, it is usually a must in a voting process. In this paper we describe an implementation of a secure eVoting system, based on ring signatures providing multiple features such as linkability or anonymity. This makes our proposal very attractive as a tool for information gathering in QoE evaluation. The system also allows linking together all the ballots of each user, without loss of anonymity, and watching the different trends in the users' opinions. Users carry out the voting with a web browser, which enables the addition of multimedia content to the poll, thus helping the information gathering process in QoE evaluation.**

Keywords—QoE, eVoting Platform, Ring Signatures

I. INTRODUCTION

Mechanisms for eVoting have been a constant since last mid-century. The first systems used punch cards or optical readers to register the ballot. This kind of voting just employed electronic devices to tally the ballots, even if the ballot itself was not registered in an electronic way.

Later, devices with different kinds of interfaces were employed as electronic ballot boxes. These Direct Recording Electronic (DRE) systems [1] let the ballots be automatically registered after being issued, and impose no further requirement to make the final count. Ballot boxes of this sort were placed in the polling stations and it was voters who had to go there to cast their vote. However, the process is not exactly the same when using DRE or traditional systems [2].

With the advancements of ICT, and specially the Internet, different kinds of eVoting were developed without the voter being required at the polling station. Voters then only needed a terminal connected to the voting net and some security mechanisms. Nowadays, there is a great number of voting possibilities via the Internet even in absence of a secure network.

Polls found in websites are one example of Internet eVoting. The questions posed can deal with any topic and the process is usually as simple as picking one of the proposed options. The questions are usually closed–ended because the final objective is to achieve quantitative results. Most often, the partial results of the voting are shown to the user immediately after he casts his vote.

There are also voting processes that go one step further in security, requiring every user to register in the system before being allowed to vote. Social networks and forums are the typical sites where these kinds of polls are deployed. In these votings, a user first identifies himself in the platform and only then does he get access to the voting area. There, he chooses one of the proposed options, if the question is closed–ended, or answers the question in a textbox if the question is open-ended.

The voting systems hitherto explained do not satisfy the full requirements of a secure voting [3]. For this reason, there were developed several secure eVoting systems based on one of the following protocols which are indeed suitable: blind signatures [4], mix-nets [5], homomorphic encryption [6, 7] and ring signatures [8]. These schemas allow the implementation of voting systems that provides users with anonymity either by means of specific cryptographic primitives or through a net of servers which prevent making a link between each voter and his ballot.

However, it is possible to go one step further and look for trends in the ballots of the users, thus tracking the changes in their state of mind. When this feature is needed it is necessary to find a way to link together the ballots of a user. Furthermore, the actual identity of the voter must not be revealed, since the anonymity of the system would be otherwise broken. The system described in [9] fulfills this requirement. Through the use of linkable spontaneous ring (LSR) signatures [10] the system is able to link all the ballots of each user, without losing anonymity, with a parameter named "linking tag". This parameter allows linking all the ballots issued by a user throughout the different rounds of a voting. It is not possible, however, to link them across different votings.

This mechanism is one of the tenants of an emergent democratic system, e-cognocracy [11], which employs ICT seeking to achieve an active participation of citizens in the decision-making process of the government. e-Cognocracy also requires tracing the users' opinions and the creation of different groups of voters carrying different weights. As the system is based on LSR signatures, it can establish a relation among voters and their ballots. This is a very powerful mechanism to perform secure polls with high quality information sources within a marketing environment.

It is in this last field where this tool is very useful to make external quality controls. Users have a system where they can

978-3-901882-50-0

contribute their ideas or suggestions in a secure way. They do not revoke their anonymity at any moment and this allows an in-depth study of the different suggestions. All these characteristics are most suitable to implement a tool to collect (in a provably secure way) information about the QoE in product, system or protocol. By using an eVoting system which guarantees anonymity the users feel reassured and they can then express their opinions in a free manner. Also, thanks to linkability the results can be filtered in different ways showing the evolution in time of the suggestions issued by a user.

In order to implement a system compliant to the specific requirements of the protocol described in [9], it is necessary to develop both the administrator of the system software (server) and the user's program (client). In this paper we detail an implementation of such a complete eVoting system. It presents all the characteristics described above and involves two different tools working together. In the server software several tools to operate the system are implemented, while client-side a JavaScript program is installed in the user's web browser. This program allows the user to access a secure voting platform and it performs all the voting process.

We chose a web browser as the tool to perform the voting based on the usability of the system. Users regularly employ web browsers and the only uniqueness about the eVoting process will be when the system asks the user for the password protecting his private key. Besides, using web browser as the way to collect QoE information is very useful as it allows introducing multimedia content within the poll and provides an integrated eVoting environment.

In section II we briefly present secure eVoting systems. We present our implementation of a secure eVoting system in section III. In section IV we analyse the election process, signature and information management. Finally we offer our conclusions in section V.

II. SECURE EVOTING SYSTEM

Information gathering and polling on the Internet is usually done through surveys and forms, as can be seen on many websites or forums. The different voting systems differentiate themselves by the means used to register the voters or surveyed users. There is a first group of systems where the registration is anonymous. When a user signs up, he creates a profile and is henceforth identified by a pseudonym or nickname. In this kind of systems the real identity of a user cannot be traced back just from his alias.

The other group consists of those platforms where a user must prove his real life identity to the administrators of the forum or social network before being allowed to register, although this authentication process can be delegated to a Trusted Third Party (TTP) if the system lacks the means to do it itself. In this kind of sites each user is provided with a profile once his identity established, and gets a user name. This identifier can be a nickname, the user's real name, his position, etc.

Nowadays, both systems are widespread throughout the Internet. Systems where user authentication is not needed are used in many forums and social networks. Authenticated ones are used in such institutions as universities or private

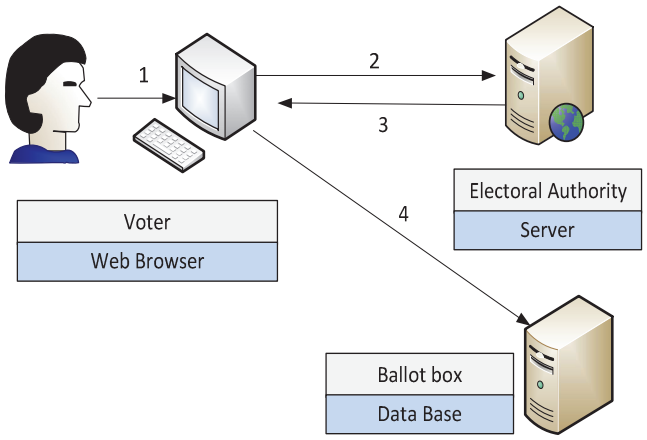

Fig. 1. Scheme of a traditional eVoting system

enterprises. However, both systems described above have several shortcomings that impede their use as a real voting system. First, when there is no authentication users can vote as many times as they want. Secondly, in both cases the system administrator knows the content of each user's ballot. Since the identification of the user is made in a straightforward manner, the identity of the user can be linked to his ballot.

To avoid these security concerns, several secure eVoting systems have been developed. These systems are based on a TTP that provides the users with their keys or certificates. Then, each voter uses them to carry out the identification in the system, thus avoiding the problems described above.

eVoting systems can be classified in four groups attending to the cryptographic primitive they use to satisfy security requirements:

- Blind signatures [4]: In a blind signature process, each voter sends to an authority its ballot obfuscated in such a way that its content cannot be known. This authority then verifies that the voter is eligible to vote and signs the obfuscated ballot. When the voter gets back his ballot signed by the authority, undoes the obfuscation and sends it to the ballot box together with the authority's signature. Finally, the ballot box checks the ballot and the signature and, if everything is correct, adds the vote to the tally.
- Mix-nets [5]: Mix-nets to anonymize the ballots by using a net of servers. These servers, called mixers, receive a batch of votes and perform a permutation of them in such a way that an observer cannot link the inputs with their outputs.
- Homomorphic encryption [6, 7]: A cryptosystem can be said to be homomorphic if there exists an operation $\oplus$ that can be applied to the encrypted messages M_1' and M_2' that corresponds to another operation $\otimes$ applied to the clear messages M_1 and M_2. Thus: $dec(enc(M_1) \oplus enc(M_2)) = M_1 \otimes M_2$. This property has been used to develop eVoting systems that tally votes as aggregates, without decrypting individual votes.

- Ring signatures [8]: The kind of ring signatures used in eVoting systems is spontaneous, short and linkable. Using these signatures a user is able to identify himself as a member of a group without revealing his identity. It is possible to avoid duplicate votes, since the linkability feature allows the detection of ballots with the same linking tag. Finally, as they are also spontaneous, they do not need key management and thus the anonymity is unconditional.

The main scheme of a voting process is shown in Fig. 1. There are a number of analogies between traditional voting and secure eVoting. In both systems the voter is assumed to have a valid way of identification, either Passport or ID-card in traditional elections and a valid certificate in eVoting systems:

- Step 1: In a remote voting process, the voter uses a terminal with an Internet connection. In a classic voting, the voter goes to the polling station.
- Steps 2 and 3: The voter contacts the electoral authority and requests information about open polls (2). The electoral authority, an Apache server in our implementation, answers with the available polls (3). In traditional voting systems the voter would look at the different ballots and the questions they posed.
- Step 4: The voter picks his preferred option, signs the vote and sends it to the ballot box. The ballot box verifies the vote using the attached signature. In classic voting, some authority at the polling location checks the identity of the user, and only then the voter is allowed to cast his vote.

Depending on the eVoting system, the electoral authority must be independent of the ballot box. Such is the case when the eVoting protocol is based on blind signatures. Some other systems, (e.g. those based on ring signatures) do not have this requirement and thus only one server is needed to host the ballot box. In this case, the steps involved in the eVoting process are shown in Fig. 2.

Ring signatures [8] are an evolution of group signatures [12]. Unlike them, ring signatures do not need a manager to set up the group and keys. This manager is able to revoke the anonymity, which is a drawback in our scenario. Ring signatures can thus be spontaneous (i.e. no previous preparation is required) and their anonymity is unconditional. Later, ring signatures which allow linking together different votes of the same user were developed [13]. The weak point of these signatures was its length, which increased linearly with the number of members of the ring. In [10] this problem is overcome and a constant length signature can be obtained with any number of members of the ring.

III. Implementation of a Secure eVoting System

The secure eVoting system implemented [9] uses short linkable ring signatures. For the actual implementation of the system several different modules have been developed that, once combined, offer a correct implementation of the proposed secure eVoting system. First, there must be a Public Key Infrastructure (PKI) [14] in charge of managing the digital

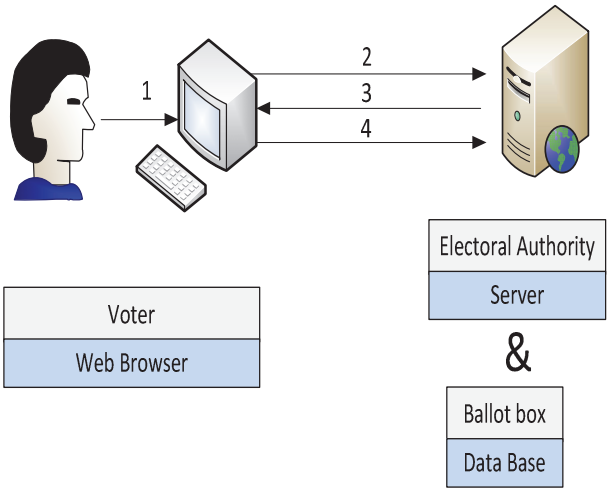

Fig. 2. eVoting process with one TTP

certificates [15] of all the parties involved in the eVoting process (both voters and authorities).

A PKI is based on the trust users put in Certification Authorities. This allows them to work as Trusted Third Parties and sign identity certificates. When a user wants to prove his identity to another party (who must also trust the same CA the user does), he will use his certificate. This way, identification tasks are eased since most of the work is delegated to the Certification Authority (CA) and to the Registration Authority (RA). CAs use asymmetric encryption (e.g. RSA [16]) to secure the certificates. They are signed with the private key of the CA so that everybody else can easily verify the signature. Anybody trusting the CA will also validate the content of the certificate and thus the identity of the user.

Additionally, some kind of voting manager is needed in the server and the means for users to access the open polls. Voters will use some client-side software within their browser without the need for any extra step of software installation. As transparency is a concern in this kind of systems, all these modules have been developed using free software: Firefox for the client; Apache/Tomcat for the server; and MySQL to manage the database.

A. PKI deployment

In addition to the several requirements of a standard eVoting system, our proposal has some specific needs that go beyond what a usual PKI offers. Hence, we have developed an ad-hoc PKI to cover our specific environment.

The platform we have decided to implement is described in [9], and it uses keys having the following characteristics:

- Operations are made modulo n, where $n = pq = (2p' + 1)(2q' + 1)$ of λ bits, being p, q, p', q' prime numbers.
- The private keys (e_1, e_2) are distinct prime numbers in the interval $(2^l - 2^\mu, 2^l + 2^\mu)$, where l and μ are security parameters of the protocol.

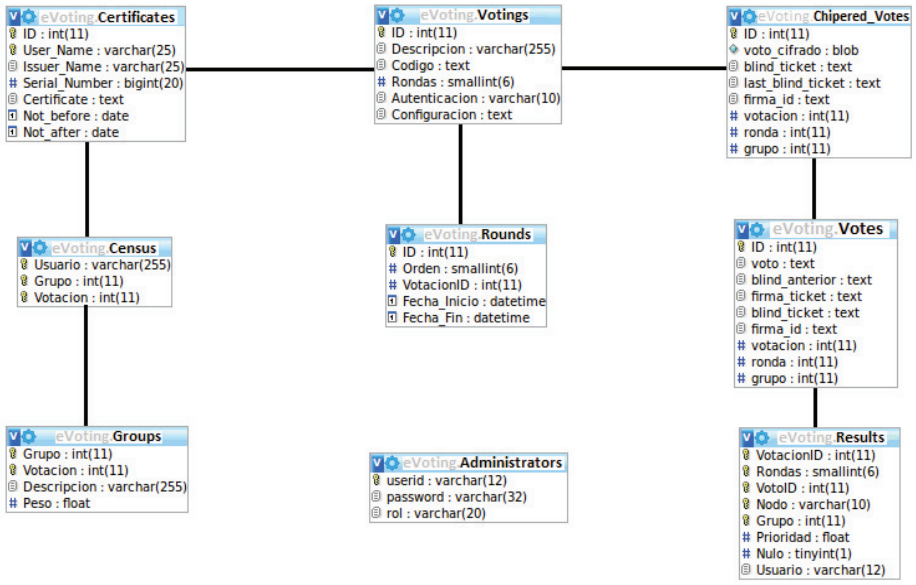

Fig. 3. Relationships between tables of the database

- The public key is $(2e_1e_2 + 1)$, which must be also a prime number.

There are no standardized certificates for this kind of keys and, at the same time, any standard web browser must be able to manage the certificates where they are stored. To overcome this problem, we map the relevant parameters in a standard RSA certificate. The public key is mapped in a straightforward manner: the modulus $n=pq$ is stored in the modulus field and the public key $2e_1e_2 + 1$ in the exponent field. As for the private key, we have three values to store: n, e_1 and e_2, that will be kept in the following fields: n is stored in the modulus field; $2e_1e_2 + 1$ in the public exponent field; e_1 in private exponent field; and e_2 in the prime1 field.

We chose standard certificates instead of developing our own ones because the former provides a clearer integration with the web browser, while using the latter would be more cumbersome for the user, as he would be required to install a modified version of the browser.

A PKCS12 certificate [17] store is needed for each user, containing both his private key and the public certificate. This way we can use standard components that can be understood by the web browser.

There must be an adequate management and control of the certificates for the PKI to be trustworthy. Proper methods of user identification and authentication must be put in place before giving their certificates to the users. Appropriate policies also need to be defined in case a non-trusted certificate should be revoked.

B. *Voting server/Electoral Authority*

In our system, the voting server is in charge of the management, administration and elaboration of the polls. It also manages the voters and census lists. It uses Java as the programming language, and the data is stored in a MySQL database. Relations among the different tables of the system are shown in Fig. 3.

Its management function can be split in two main areas: user management and voting management. User management is done in several steps. First, the user must present a valid PKI certificate to the system. The system administrator then manually uploads it to the server, where it is verified. Once stored, the server application extracts the relevant fields from the certificate and stores them. The full certificate is also kept since it will be used by the rest of the voters to calculate some mathematical parameter used in the eVoting protocol and to expose the census. Certification expiry is managed automatically and no certificate can be explored used out of their validity period.

Voting management is fully done by the administrator, who uploads the necessary parameters for the voting: variables, census, number of rounds, and the question. He then creates the different user groups and assigns a weight to each of them (there can also be only one group where there is no weighting).

The fields corresponding to the voting parameters are fixed (though they can be of variable width). Also, the census is just a list of the users allowed to take part in the poll. However, there is a wider range of possibilities when it comes to the question posed.

The administrator is allowed to create a personalized environment for each scenario. It can be any valid HTML / JavaScript without any restriction. It just has to provide a JavaScript function named *voteBallot()* that will be called by the voting framework. This function shall return a string representing the content of the ballot. This way, the ballot can be fully customizable with either open or closed-ended question, using the standard user interface voters are familiar with (radio buttons, text boxes, images, etc.).

C. Client/JavaScript

To improve the usability and the test of the eVoting, the client program has been developed using JavaScript. The user's browser automatically downloads the JavaScript hosted in the voting page and executes it. The client program is designed to be compatible with the web browser Firefox running under any supported PC platform (Windows, MacOS, or Linux among others).

This JavaScript script is responsible of managing PKCS12 certificates used in the eVoting system. It also carries out the information interchange necessary to obtain the voting parameters and signs the ballot. This whole process has been designed from the ground up to be transparent to the user, who will only be asked the password for the certificate store where his private key is kept. The voting concludes with the system showing the user some key information about the process and the final value of the signature.

IV. Election process, signature and information management for QoE evaluation

The interaction of the user with the eVoting system is shown in Fig. 2. The user begins the process with the voting system and chooses the voting in which he wants to take part. Then, the server shows the voting screen and the voter answers to the question, either by picking his preferred option or giving an opinion if it is an open-ended question. The signing process starts after the selection is finished. The script in the client will ask the voter both the location of the PKCS12 certificate where the private key is stored and the password to unlock it. Once the access has been granted the signing process starts, and the parameters needed to carry out the signature are requested from the server. If needed, the answer is hashed before its signature. Once the ballot is signed, it is encrypted using the public key of the ballot box and it is sent to the server together with its signature.

During all the process the user is shown several information windows, including the selected option, the user who is going to do the signature, and the numeric value of the signature.

The tally starts once the poll is closed. The ballot box is in charge of decrypting the votes using its private key. In this moment the ballot box verifies the signatures attached to the ballots. Those that are not valid will be discarded together with their votes. When all the valid ballots have been decrypted, the votes from the same voter are detected, using the linking tag, and all but the last one are excluded from the recount. This is so because the system allows voters to change their mind and their vote while the poll is still active. The only valid vote is the last one cast within the valid period, though this policy can be changed by the voting administrator.

As the proposed eVoting system is based on ring signatures, a list with the identities of the users who have participated cannot be published, but a list with the linking tags of the users who have voted can indeed be made publicly available. If there are voters with different weights, extra recounts can be done later weighting the partial tallies to obtain the final result.

Votings can have various rounds, each one within a fixed time period. Therefore, an analysis of the evolution in the voters' opinions can be done because the ballots of the same voter will be linked via their linking tag. Thanks to this feature, this eVoting system is of great value to poll users about QoE. It is not only a secure mechanism but it is also able to track the evolution of their opinions.

QoE management has three basic steps [18]: understanding and modeling QoE; monitoring and estimating QoE and adapting and controlling QoE. In the first one it is necessary to perform an evaluation of the QoE perceived by the users, usually asking for their mean opinion scores (MOS). Once those are obtained, a value which represents the mean opinion of all the users is calculated. Since QoE and QoS are related [19, 20], with our proposed platform we can give the same model (with predefined QoS parameters) to all the users and ask for their QoE evaluation. Thanks to the linkability feature of the system, it can be seen that some users respond in the right and expected way. In future votings, these users will have a higher weight and so it will be possible to obtain a better QoE estimation when an absolute control of the evaluated object cannot be achieved. Besides, through the analysis of the results, it is possible to make observations, both individual and collective, of users' QoE variations along time and with different QoS parameters. This allows the poll administrator to perform a more detailed analysis of the voters' opinions and to feedback the polls with more specific information for each of the groups with similar features. All this is achieved without ever losing user's anonymity.

To test the system, a laboratory experiment has been performed. The client program was installed on several computers of the laboratory network with different O.S., and an easy voting with three options was carried out. The obtained results allowed us to check the correct performance of the system both in the client and in the server side: the creation of a voting and a census, the user management, the voting selection, the signature of the chosen option, the development of the voting, and the final tally were correctly performed. We also did a PKI experience: we created certificates for each user, then verified each user's identification before giving him his PKCS12 certificate and finally proceeded to their secure distribution.

V. Conclusions

A secure eVoting system allows the collection of quality and reliable information. Our proposal meets the requirements to be used in several fields. Its immediate application is the substitution, or complementation, of present voting systems, new management systems of democracy (e.g. eCognocracy), or to carry out the information gathering in a secure and reliable way in marketing polls. Finally, it can also be a tool in systems to evaluate QoE of a protocol or algorithm.

The implementation of a secure eVoting system requires satisfying a set of features in such a way that their implementation does not introduce any flaws. In this paper a real implementation of a secure eVoting platform using ring signatures is shown. To develop the system, free and verifiable software has been employed.

The system presents several features that make it suitable to make QoE measures in secure environments. Our protocol guarantees anonymity of the voters and also allows linking the issued votes by a user and analyzing the QoE evolution during the process.

As a future line of work, we are trying to perform a wider study about the usability of the system through its use in bigger and more complex polls than has been possible until now in the laboratory testing environment.

ACKNOWLEDGEMENTS

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] T. Kohno, A . Stubblefield, A.D. Rubin and D.S. Wallach: "Analysis of an electronic voting system". In: IEEE Symposium on Security and Privacy. IEEE Computer Society Press, Los Alamitos (2004)

[2] S. P. Everett, K. K. Greene, M.D. Byrne, D.S. Wallach, K. Derr, D. Sandler and T. Torous: "Electronic voting machines versus traditional methods: improved preference, similar performance". In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08). 2008, pp. 883 - 892. DOI=10.1145/1357054.1357195

[3] B. Lee and K. Kim: "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.

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

[5] M. Jakobsson, A. Juels, and R. L. Rivest: "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.

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

[7] I. Damgård, M. Jurik: "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System". In: PKC 2001. LNCS, (vol. 1992), 2001, pp. 119–136.

[8] R.L. Rivest, A. Shamir and Y. Tauman: "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), 2001 , pp. 552-565.

[9] J.L. Salazar, J. Piles, J. Ruiz and J.M. Moreno-Jiménez: "Security approaches in e-cognocracy". Computer Standards and Interfaces, 32 (5–6), 2010, pp. 256–265.

[10] P. P. Tsang and V. K. Wei: "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. DOI=10.1007/978-3-540-31979-5_5

[11] J.L. Salazar, J. Piles, J. Ruiz and J.M.Moreno-Jiménez: "E-cognocracy and its voting process", Computer Standards and Interfaces, 2008, pp 124–131.

[12] D. Chaum and E. Van Heyst: "Group signatures". In Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques (EUROCRYPT'91), 1991, pp. 257 - 265.

[13] J. Liu, V. Wei, D. Wong: "Linkable spontaneous anonymous group signature for ad hoc groups", in:, ACISP 2004. LNCS, (3108), 2004, pp. 325–335.

[14] C. Adams and S. Lloyd: "Understanding Public-Key Infrastructure – Concepts, Standards, and Deployment Considerations". Macmillan, Indianapolis (1999)

[15] S. Brands: "Rethinking Public Key Infrastructure and Digital Certificates - Building in Privacy". PhD thesis, Eindhoven Institute of Technology, Eindhoven, The Netherlands, 1999.

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

[17] PKCS #12: Personal Information Exchange Syntax Standard. http://www.rsa.com/rsalabs/node.asp?id=2138. Revised 01/04/2013

[18] T. Hobfeld, R. Schatz, M. Varela and C. Timmerer: "Challenges of QoE management for cloud applications," *Communications Magazine, IEEE* , vol.50, no.4, pp.28-36, April 2012 doi: 10.1109/MCOM.2012.6178831

[19] H. J. Kim, D. H. Lee, J. M. Lee, K. H. Lee, W. Lyu and S. G. Choi: "The QoE Evaluation Method through the QoS-QoE Correlation Model," Networked Computing and Advanced Information Management, 2008. NCM '08. Fourth International Conference on , vol.2, no., pp.719-725, 2-4 Sept. 2008 doi: 10.1109/NCM.2008.202

[20] H. J. Kim and S. G. Choi: "A study on a QoS/QoE correlation model for QoE evaluation on IPTV service" Advanced Communication Technology (ICACT), 2010 The 12th International Conference on , vol.2, no., pp.1377-1382, 7-10 Feb. 2010