This is a preprint version that corresponds to the paper published in Computers & Security, Vol. 56, pages 50-62 (Feb, 2016), which includes experiments, timing charts and better explanations in some key aspects.

A flexible e-Voting scheme for debating tools

D. A. López García
Department of Electronic Engineering, Computer Systems and Automatics, University of Huelva, Campus La Rábida, Palos de la Fr. s/n, CP 21819, Spain.
E-mail address: diego.lopez@diesia.uhu.es.

ABSTRACT

In order to protect votes, e-voting schemes provide privacy, verifiability and eligibility among other features. Most e-voting systems focus on the vote itself, considering it a fixed and limited piece of data (e.g. choosing from a list of candidates) which has to be encrypted and repeatedly permuted. In recent years, collective decision systems have been developed in which participants can submit proposals, argue and vote. In these systems the information to be protected is variable and plentiful, inaccessible to classic methods. The scheme proposed in this paper has been designed to cover these needs. Using the technique of blind signature, an alias is granted to the user that guarantees privacy. The novelty is that this alias consists of a public key that gives users the ability to encrypt and sign. From this alias users can access to the system and operate as if no privacy were needed, which greatly simplifies the process of voting, validation and counting. In addition, this article details one improvement that makes the scheme resistant to coercion.

Keywords
e-voting; blind signature; variable length ballot; e-democracy; incoercibility

1. Introduction

In recent years, web tools have been developed in order to help debating and taking decisions. For example Argunet[1] (Schneider et al., 2007), DebateGraph[2] (Tiropanis et al., 2009), Carneades[3] (Gordon et al., 2007) and bCisive[4] (Benn and Macintosh, 2011) create diagrams with the arguments of a debate in an intuitive way (Fig. 1 shows the structure used in DebateGraph). The users of these tools can send opinions (opposing or supportive arguments, neutral opinions or related information) and proposals which will be the nodes of a debate (questions and possible answers, actions to take, rules to establish, etc.), but no votes. This lack is solved with tools like TakeOnIt[5], Ivoting[6] (Tiwari et al., 2013), Debate.fm[7] or Agora voting[8] (Verdejo et al., 2008). Nevertheless, they are focused on voting only one statement. The mix of the two needs (an open debate with multiple proposals and an e-voting system for them) is present in tools like Deliberatorium[9] (Klein, 2011) or MyUniversity[10] (Cucurull et al., 2013) which makes use of a voting system with a platform of ordered forums (Gov2Demoss). In general, these tools allow the user to submit different kind of contributions (opinions, proposals and votes), the issues to vote are not known from the beginning, and the amount of contributions per user is variable (López et al., 2013). In these flexible systems, the wishes of every user can be expressed by their votes on a subset of proposals which in turn can have different formats: yes/no, a number from an interval, a name from a list, etc. With these features, it's difficult to design an application able to provide strong security.

A lot of e-voting methods have been designed in order to accomplish several security constraints. However, it does not seem useful to consider all of them. For example in Helios (Adida, 2014), based on Benaloh (2006), some privacy is given up in exchange for the integrity of votes. Users encrypt their votes and send them to a server, which collects the votes of all participants and posts them on a table next to the author. Thus, everyone can see that what they voted is exactly what is going to be counted. Next, the server uses a mixnet (Chaum, 1981) to shuffle the votes and to delete the relationship with the voter. Then the votes are decrypted and added. Helios is a system designed to be practical, so it suffers from certain weaknesses. The author suggests other systems (Juels et al., 2005) that would improve security but at the same time they would turn it complex and cumbersome for the user. Moreover, this system is focused on votes of fixed structure, with a number of predefined alternatives.

Numerous published e-voting systems (Sampigethaya and Poovendran, 2006) offer a high level of security. They can be classified as hidden user (Boyd, 1990; Sako and Killian, 1995), hidden vote (Benaloh, 1987; Cramer et al., 1997) or both (Fujioka et al., 1993; Jakobsson et al., 2002). In hidden vote systems, a part of the process is operated directly with the encrypted votes. For example, with ElGamal encryption, if the product of all the encrypted votes is done, then encrypted sum is obtained. Thus, only the end sum is decrypted, and maximum privacy is obtained. For this purpose all votes have to share the same structure. Therefore these systems are not suitable for variable length messages. Closest to the flexible utility would be the first voting type, where each vote is individually processed. In these systems, a user identifying string (token) is attached to votes in order to avoid duplication. The vote and the token are encrypted and sent anonymously. This is the approach used in this work.

The novelty contributed in this paper is to use blind signature to validate a public key, which is used as an identifier alias (token) but also serve to sign the votes cast. An authority performs the blind signature of the alias encrypted by the user. This one

will use his/her alias to access the debating tool. Since there is no relationship between user and alias, mixnets are not required, avoiding complex and slow systems of multiple encryptions and permutations. The validation of the vote will be transparent to the user, since no encryption will appear in the final bulletin board. The main advantage of this scheme is its independence of the information to be encrypted, which opens the door to an anonymous interaction with the system, so that everyone can propose, criticize, inform or vote. There are no constraints on the exchange of information. Flexibility is achieved because any format and size is possible for the contributions (proposals, opinions and votes) within the space allocated to each user on the bulletin board. A signed hash with the alias validates the contents of that space.

In this sense Chaum (1981) already used keys as user identifiers and called them "digital PSEUDONYMS", but through a mixnet to provide privacy. The idea of blind signature (Chaum, 1983) has been used later on more complex voting systems (Chaum, 1988; Boyd, 1990) where the signed item ceases to be the public key and becomes the vote itself. While newer systems are evolving on vote cryptography (Ribeiro and Joaquim, 2012), there are several works focused on the user (Cranor and Cytron, 1997; Fujioka et al., 1993). Among the latter, one of the closest to this work and also most debated is that published by Mu and Vaharadharajan (1995), which like Chaum's uses a key as an identifier of the user. However, this system has security faults (Chien et al., 2003), and attempts to address them (Lin et al., 2003; Yang et al., 2007; Rodríguez-Henríquez et al., 2007; Asadpour and Jalili, 2009; Baseri et al., 2011; Mateu et al., 2014) increase the complexity and number of parameters required for each vote significantly. The simplicity of the proposed system allows easy scalability and can fit any of the existing systems of decision, since it does not require any format.

This paper is organized as follows. First, a brief explanation of essential elements used in this scheme is placed in section 2. Next, the new e-voting method is described in section 3. Section 4 lists the security features of the method and suggests improvements. A comparative with the closest approaches takes place in section 5. Conclusions of section 6 close this work.

2. Preliminaries

2.1. Communication channels

It is possible to establish private channels on the internet by means of TLS protocol. These channels can be listened by others, but the information cannot be copied without breaking an RSA key. In order to stablish these channels, each end shows a digital certificate. In this scheme users must attend a public office to get his/her digital certificate and obtain some other information like the digital certificate of the authority. Under these conditions the channel is considered fairly secure and free of well-known techniques like spoofing or MITM attacks.

Anonymous channels are defined as channels in wich there is no way to identify the sender of messages. There are different solutions proposed like DC-nets and mixnets. This privacy is achieved taking into account two facets: physical access and logical identification. For example, mixnets shuffle the encrypted ballots of voters. The link voter-ballot is broken by shuffling (physical facet) and encrypting (logical facet). The logical link must be considered in all the schemes, because voters must be elegible and therefore their identity must be checked. The physical link can be broken in many ways, and doesn't have to be specificed in each scheme. For example, a voter can connect from a public place: libraries, hotels, universities, Internet cafes, venues, etc. Authorities could offer them by means of a wifi access point in schools, parks, airports, etc. In addition, there are many tools to break physical traceability from home: private proxies, TOR network (Snader and Borisov, 2008), VPNs, I2P (Herrmann and Grothoff, 2011), etc. Even if voters choose to connect without any protection, it's difficult to trace them because common IP addresses are private and dynamic, and only the border routers have this information. In summary, physical traceability can be considered apart from e-voting schemes.

2.2. Blind Signature

A blind signature is similar to a digital signature except that it allows an entity (in this case a voter) to get another entity (the authority that checks identity of voters) to sign a message without revealing its content. Blind signature (Chaum, 1983) is used to authenticate the voter without disclosing the content of a ballot (Chaum, 1988; Boyd, 1990). Hence, the authority whose function is to verify the eligibility of a voter, will not know who is being voted. The security level of blind signature has been analyzed (Ibrahim et al., 2003; Bellare et al., 2003) with satisfactory results.

To ensure the secrecy of his/her message, a voter blinds it by using a random number r in a blinding function and sends it to the authority. For example, if the authority A has a pair of RSA private/public keys denoted by K_A^{-1} / K_A, the blind function for a message m, $Blind(m,r,K_A)$, can be defined as:

$$Blind(m, r, K_A) = m \cdot r^{K_A} = m' \tag{1}$$

Equation (1) and the followings represent operations in modular arithmetic. The authority receives this blinded message and cannot disclose it without r. Next, the authority signs the blinded message with its private key K_A^{-1} and retrieves it to the voter:

$$Sign(m', K_A^{-1}) = (m')^{K_A^{-1}} = (m \cdot r^{K_A})^{K_A^{-1}} = m^{K_A^{-1}} \cdot r \quad (2)$$

Finally, the voter removes the blinding factor by multiplying all by r^{-1} and obtains the message signed by the authority.

$$Sign(m', K_A^{-1}) \cdot r^{-1} = m^{K_A^{-1}} \cdot r \cdot r^{-1} = m^{K_A^{-1}} = Sign(m, K_A^{-1}) \quad (3)$$

In the method presented here, a voter will contact with the authority directly, revealing his/her identity. The authority will check the voter in a list and, for every voter, it will allow one and only one blind signature. The authority can store the blinded messages, but it will not be able to disclose them. So that, privacy is garanted.

3. Protocol

In this section the main stages of the new e-voting method are described. This scheme considers only two entities in addition to the voters. This is the simplest scheme, easy to implement and with maximum privacy and verifiability. In the analysis section different improvements are suggested with more entities and higher security, but more complex too.

The participants in this scheme are:

i. Voters, denoted by V_j, where $j \in \{1,2, ..., m\}$. Each voter has a digital certificate with a pair of RSA keys, that is denoted as $\{D_{Vj}, D_{Vj}^{-1}\}$. This certificate is delivered in an official place, where voters are physically identified, previously to any voting. Once delivered, it is useful for multiple polls.

ii. The census authority '*C*', devoted to check whether a voter identified by his/her digital certificate is in the registered voter list and to make blind signatures. This entity maintains a bulletin board where voters, blind messages and blind signatures are issued. The RSA key pair of *C* is $\{K_c, K_c^{-1}\}$. In order to avoid spoofing, voters are warned about K_c when they receive their certificates.

iii. The voting system '*VS*', where the users send their contributions. *VS* has its own bulletin board where all the contributions are published. *VS* knows the public key of *C* in order to verify the users, as it will be explained below. *VS* has to make the final tally.

The protocol considers three stages: register, debate and tally. Fig. 2 shows a brief scheme of them.

a) Register stage: The objetive in this stage is to provide voters with an alias key signed by the authority. All the steps are depicted in Fig. 3.

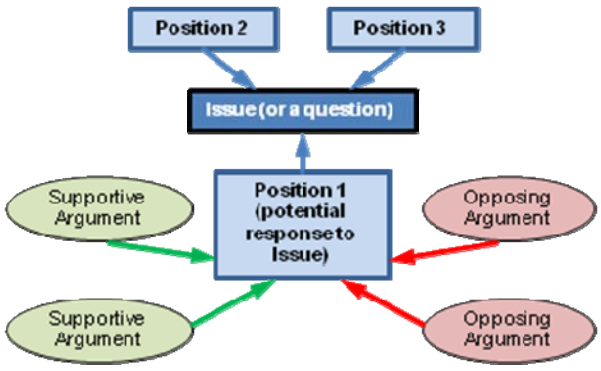

Fig. 1. Logic structure used in Debategraph.

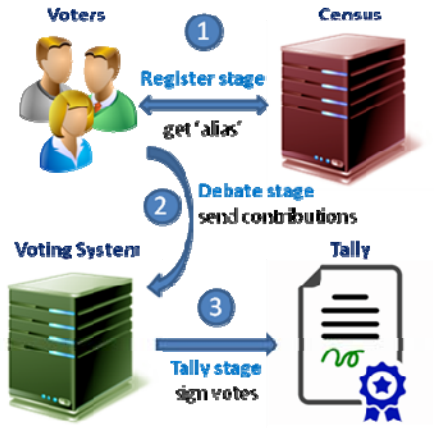

Fig. 2. Stages of the new e-voting system.

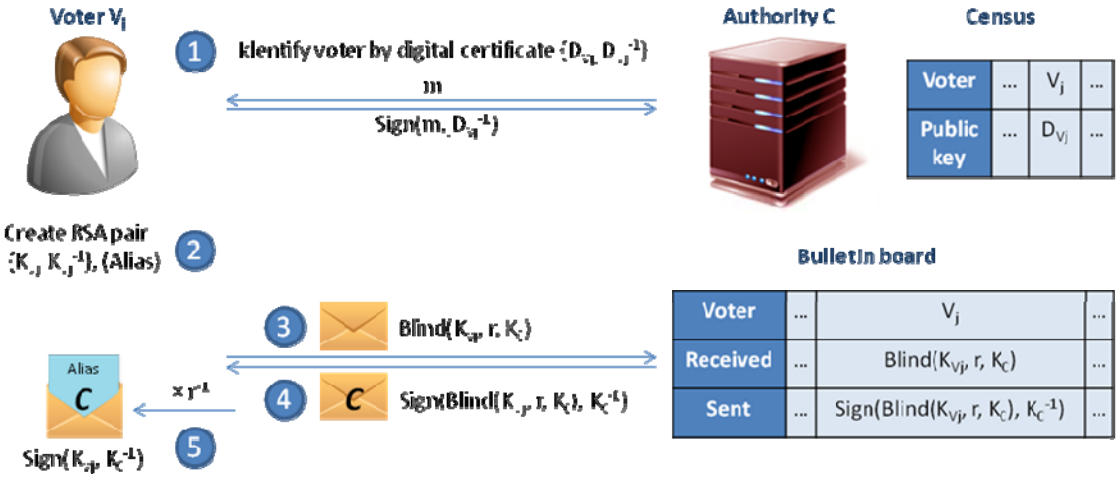

Fig. 3. Registration stage: Voter opens a secure channel and is identified (1); voter creates alias keys (2); voter sends the public alias key blinded (3); Authority C signs it (4); voter disclose the message obtaining the alias key signed by C.

a.1. Voter establishes a private channel (e.g. by TLS) with *C* and uses his/her digital certificate to prove the identity.

a.2. Each voter generates a pair of RSA keys, denoted by $\{K_{Vj}^{-1}, K_{Vj}\}$, from now on 'alias' keys.

a.3. Each voter blinds his/her public alias key, K_{Vj}, with a random number *r* and the public key of *C*. Then, the message $Blind(K_{Vj}, r, K_C)$ is sent to *C*.

a.4. *C* issues the received message on the bulletin board.

a.5. Voter checks that the message issued is right and confirms it to *C* (e.g. by signing it). This step is a shared period of time for all voters. Any claim must be resolved before the next step.

a.6. *C* makes a blind signature of the message, publishes it on the bulletin board and sends it to the voter.

a.7. Voter unblinds the message obtaining his/her alias key signed by *C*.

b) Debate and voting stage: The voting system *VS* checks each new user by the *C* signature, and allows them send contributions and cast votes (Fig. 4).

b.1. Voter sends his/her alias public key signed by *C* to *VS* by means of an anonymous secure channel (detailed in section 2.1).

b.2. *VS* removes *C* signature, and stores the public alias key, K_{Vj}, of the voter. Then, *VS* sends a random message m to the voter.

b.3. Voter probes he/she has the private alias key, K_{Vj}^{-1}, by returning the random message signed.

b.4. *VS* discloses the signed message and check whether it coincides. If all is right, the new voter is added to the bulletin board, with the pair $\{K_{Vj}, Sign(K_{Vj}, K_C^{-1})\}$. If there is a key collision (K_{Vj} was registered by another user yet), *VS* warns each user so that they can obtain a new signature from *C*. For this poupose, *C* must receive from *VS* the key banned (K_{Vj}), and from users the *r* parameter. This way, *C* can open the blind signature and check whether K_{Vj} was the key signed.

b.5. From now on, each voter registered in the bulletin board has access to the debate and voting system. They can send opinions, proposals and votes. Each voter can see opinions and proposals of the rest of the voters, but only his/her own votes. This interaction goes on a TLS channel which any application can use. No format of messages is required. Therefore no constraints are imposed on the application except that at the end it must show all the activity of the user in a sheet (see left side of Fig. 5).

c) Tally stage:

c.1. When the time is up, *VS* doesn't accept contributions any more. Each voter has a period of time to check whether his/her contributions (votes included) have been registered in the bulletin board of *VS* correctly. For this pourpose each voter gets his/her contributions sheet, and the hash computed by *VS*.

c.2. If the contributions sheet is right and the hash computed by the voter matches, then the voter signs the hash and sends it to *VS* (see right side of Fig. 5).

c.3. When the checking period finishes, VS computes the tally, that is, for every proposal VS counts the votes cast and valid (from sheets signed). The list of proposals with its votes added is appended in the bulletin board.

At the end, every one can check the tally because the contributions of voters, signed by them, are shown in the bulletin board.

4. Analysis

Some of the security features usually analyzed in other voting systems (Sampigethaya and Poovendran, 2005; Wu et al., 2014) are: eligibility, privacy, verifiability, accuracy, fairness, incoercibility, mobility and scalability.

4.1. Privacy

In e-voting systems, any traceability between the voter and its vote must be removed. Clearly, if the rest of voters reveal their vote, the latter would be exposed. When this is the only case in which the vote can be related with its voter, the highest level of privacy is achieved and it is called "maximum privacy".

In this e-voting system, privacy exists as long as alias is not linked to the voter. In the first step this is achieved with the blinding function and the random number. In the second, the voter provides a remote connection using the alias key. The traceability to the voter can only be physical. That is, it could be revealed by following communications to its source. As it's mentioned above (section 2.1), there are different tools to avoid physical traceability which are not considered here. Therefore, this method permits maximum privacy.

4.2. Elegibility

It refers to the ability of the system to determine that users are valid voters. Usually, this means that the user belongs to the list of registered voters (census). The security level here depends on the requirements to infiltrate false voters. To this end, somebody outside the authorities should have to deceive *C* or *VS*:

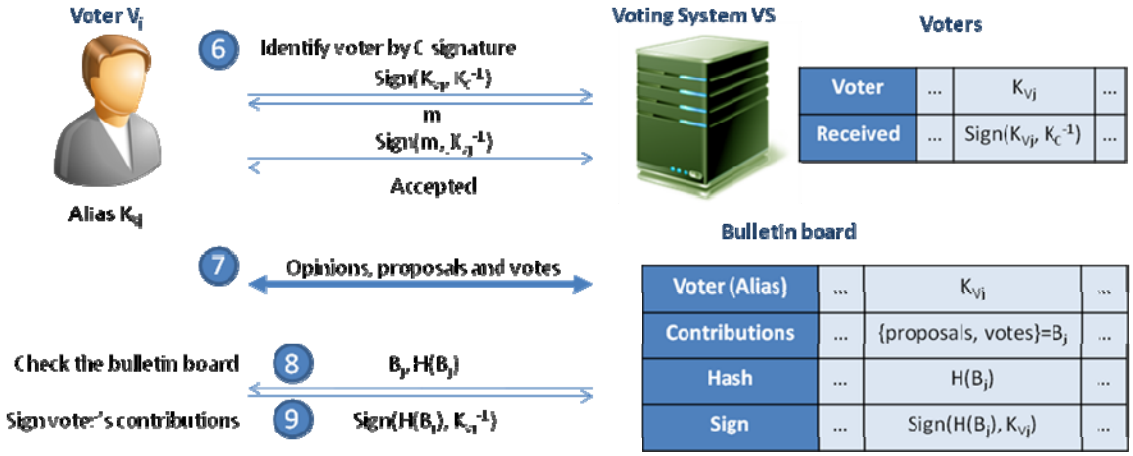

Fig. 4. Debate and Tally stages: Voter opens a secure channel and is identified (6); voter debates and votes (7); voter takes from the bulletin board his/her sheet of contributions and the hash (8); If all is right, voter sends the hash signed (9). Whoever can make the tally taking into account the signed sheets.

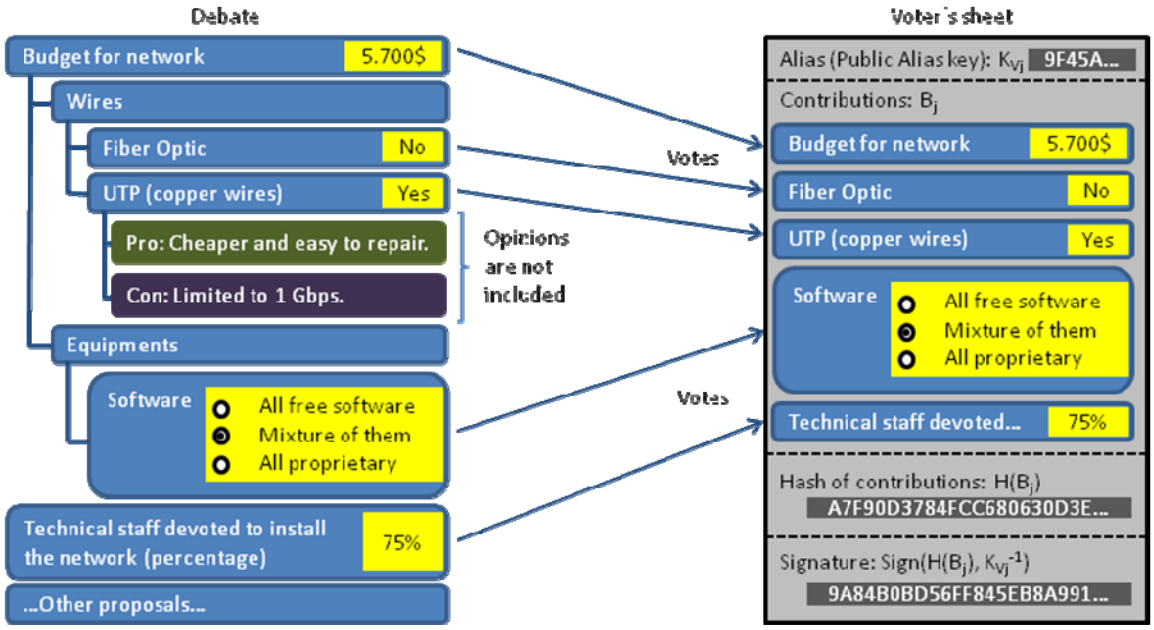

Fig. 5. Debate and Voter's sheet. Left side shows a debate with proposals arranged as folder tree. Some of them have been voted by the user (notice heterogeneity of votes). The sheet that *VS* shows at the end for every user has the appearance of the one represented in the right side.

i. For the first case the digital certificate of a voter must be stolen or broken, because *C* only accepts registered voters. Moreover, due to the fact that *C* only allows a signature per voter, the right user would be rejected and would claim (step 'a.5' in section 3), preventing the fraud.

ii. In the second case, a signature of *C* is needed to deceive *VS*. However, *C* signs only one alias per voter. Therefore the *C* private key must be broken or the alias of another user stolen. The last case implies two users trying to access *VS* with the same alias key. *VS* would warn the two users that they will be banned and they should complaint. The right user could show the exchanged messages to *C* in order to obtain a new signature.

Taking into account corrupt entities, *C* could register as much fake voters as wanted. Nevertheless, *VS* cannot accept fake voters without being detected, because all users must appear in the bulletin board next to the *C* signature. Therefore, *C* is the weak link of this scheme.

In order to improve the method in this aspect, the role of *C* can be distributed in *n* entities. Each one will have the same function as *C*. Voters will follow the same steps with all of them. *VS* will accept new users only if they send all of the signatures, which will be placed in the bulletin board. With this improvement, only a collusion of all *n* entities could get fake aliases. Thus, as more entities are added the system gets safer, but more complex too.

In summary eligibility property of this method is guaranteed, unless *n* authorities conspire or *n* RSA keys are broken.

4.3. Verifiability

This is the voter's ability to verify that their vote is casted as intended and correctly accounted in the final tally. There are two flavors of this requirement (Sako and Killian, 1995). One is the individual verifiability where only the voter can verify its vote in the tally. The second is universal verifiability, where after the tally is published, anyone can verify that all valid votes were included, and the tally process was accurate.

The bulletin board of *VS* allows voters verify their votes, thus individual verifiability is accomplished. At the end, the document published in the bulletin contains the sheets of the voters, signed by them only when they have verified their votes. The bulletin board is public; therefore whoever can check the signatures and whether the sum of sheet votes matches the final tally. That is, universal verifiability is achieved.

In addition, everybody can check the bulletin board of *C* (or n entities) to verify which voters have been admitted. Thus, the percentage of abstention can be deduced. The list of alias signatures can be checked too, in order to verify that every user is a valid voter. All these checks can be done by everyone: authorities, users and external observers.

Verifiability implies **accuracy**, which is mentioned in other papers as an additional feature. An e-voting system must be error free, that is, votes must be registered correctly and votes of invalid voters must not be counted. The way to achieve this accuracy is by verifiability. If an error happens, it can be detected in the bulletin board and corrected.

4.4. Dispute-freeness

Voting methods must provide a mechanism to resolve disputes. In this method every critical step is followed by a record in the bulletin board, and a time to check it and claim when it is wrong. Voters must keep a private record of their actions too. For example, the step 'a.5' avoids wrong blind messages and *C* cannot sign anything without voter's confirmation. If there is a problem, electoral authorities can supervise the record of the voter and check whether it's registered in the bulletin board correctly. If it isn't, they can determine who is wrong. This can be done in any other step and no entity neither voter can deceive or be wrong without being detected. In addition, the treatment of every voter is independent, therefore a problem with a voter cannot affect to the others, and the process hasn't to be redone.

4.5. Fairness

In order to conduct a fair voting, no one should be able to compute the partial tally as the election progresses. This is the reason to allow voters to watch only their own votes at the step 'b.5'. The votes of sheets in the *VS* bulletin board can be seen only by their owners. Once the period of time to sign them is up, everyone can read them. In other words, no one can compute a partial tally except *VS*. Nevertheless, voters can change their votes at any time; therefore early tallies may not have to do with final tally.

This scheme is designed for debating tools, where knowing the most voted proposals (by showing a partial tally) could help the rest of voters to contribute in the most important decisions for the community. Thus, fairness could not be a desirable feature. However, an easy improvement can be done so that fairness is increased. This consists on sending each vote encrypted, and revealing the key used after all the voters' sheets have been signed. With this improvement even *VS* cannot compute a partial tally. The drawback is that the debate tool has to accept an encrypted format for every type of vote, and this means loss of flexibility and more complexity.

4.6. Mobility

This scheme is designed to operate on the internet. For this purpose, a digital certificate is needed. Digital certificates and certain information like K_C, are provided once in a public office where the authorities check the physical identity of voters. Since then, voters only require a browser to start the process. All the security requirements can be achieved from any place of the world and multiple debates and decisions can be carried out with the same digital certificate.

4.7. Incoercibility

This feature means that nobody can coerce or buy a voter. The classical solution is the use of booths. Benaloh and Tuinstra (1994) proposed an e-voting scheme that makes use of booths, but it clashes with mobility. Sako and Killian (1995) has incoercibility in their scheme with the assumption of untappable channels and that no private voter's information is revealed. In spite of the fact that untappable channels are impractical, the second point depends on voters, turning them in the weakness of that scheme.

In general coercion is possible as long as voters receive something (records, or receipts) to probe their votes to the coercer. Some schemes are receipt-free in order to prevent coercion (Lee et al., 2003; Kiayas and Joung, 2004, Juels and Jacobson, 2002; Acquisity, 2004). Receipt-freeness is often a standalone feature analyzed in e-voting, because is related with privacy too. However, receipt-free methods use to get more complex and lose scalability.

In addition to receipts, coercers can deduce ballots from bulletin boards or the messages between voter and entities. Therefore, individual verifiability conflicts with incoercibility. For example Chen et al. (2004) trades verifiability with incoercibility, by preventing voters to check their ballots. In this scheme a supervisor entity has this function, taking part in the communication between voters and the election center. Nevertheless, the scheme uses a temporary pseudonym which coercers can steal or buy from the voter. Juels et al. (2005) proposed a method that faced incoercibility deeply and which was improved by Smith (2005) and Schweisgut (2006). The idea is to allow voters to build a legal credential resembling the one created by the authority. Nevertheless, only the vote encrypted with the original credential is counted. This way, even if a ballot is cast according to the coercer's wish, this one cannot discern it. This method is complex (Weber, 2006) and losses verifiability too.

In the e-voting scheme presented here, if voters carry out all the process in a protected place (booth) like other systems, no receipt is needed. A coercer can ask any voter his/her alias key, but he/she cannot prove which is. Nevertheless, this method is intended to be used from any place. In this case, the coercion is possible. Even so, an improvement can be added. It is based on the fact that if voters can change their votes after being proved to the coercer without being detected, then coercion is avoided.

The improvement starts in the registration stage, where a PIN code (a random and short number, easy to remember) is delivered to the voter with the digital certificate. There are two cases:

i. The coercer attacks before the alias key is signed (step a.6 in section 3).

   a. The voter is forced to give his/her digital certificate. From this point on, coercer follows all the steps as a right voter and sends the alias key blinded (step a.5 in section 3).

   b. Meanwhile, the voter can connect with his/her digital certificate and show the right PIN code. This step cannot be carried out by the coercer because the voter can give a PIN, but there is no way to know whether it is the right one.

   c. *C* now has two sessions opened with the same digital certificate, one with the right PIN (true voter), and the other one with the false PIN (coercer). Then, *C* sends each user a different public key (K'_C for the right user and K''_C for the coercer), so that users discard the old blind messages and prepare other ones with the new keys. A random set of users can be called on to change the public key. This way coercer cannot determine whether he/she has been revealed. Next steps are the same ones that are explained in the next case from point d.

ii. The coercer attacks after alias key is signed. The steps, which are depicted in fig. 6, are the followings:

   a. A coercer force a voter to give all the information: digital certificate, alias keys $\{K_{Vj}^{-1}, K_{Vj}\}$, blinding parameter *r*, and PIN code. Coercer can check all of them except PIN code. From now on coercer can vote impersonating the user with the alias keys $\{K_{Vj}^{-1}, K_{Vj}\}$.

   b. When the coercer is not present, voter warns *C* that the alias $\{K_{Vj}^{-1}, K_{Vj}\}$ has been stolen. Voter offers also blind parameter *r*, the digital certificate, and the right PIN code.

   c. *C* checks that one of the blind signatures of the list corresponds to the *r* parameter and the public alias key, K_{Vj}. If the PIN code and the certificate are right, then *C* creates a RSA key pair, $\{KM_{Vj}^{-1}, KM_{Vj}\}$, which will be called the 'mirror' alias. *C* transmits it and the stolen alias to *VS.*

d. The voter is allowed to restart the process with a new alias ($\{K'_{Vj}{}^{-1}, K'_{Vj}\}$), but signed with another *C* public key (K'_C). Otherwise, if PIN code is false, *C* will follow the same process with another public key (K''_C). Then *C* sends to *VS* a set of valid signature keys $\{K'_C\}$ and a set of coercer signature keys $\{K''_C\}$. For every alias signed with coercer keys($\{K''_{Vj}{}^{-1}, K''_{Vj}\}$) a mirror alias will be created ($\{KM''_{Vj}{}^{-1}, KM''_{Vj}\}$).

e. At the end, *VS* copies all the votes of the stolen alias and computes just the opposite values for the mirror alias, and signs the mirror alias sheet, just as any user. Thus, the coercer will see a valid sheet for the stolen alias, and cannot check whether a mirror alias has been created.

With this improvement the level of security for preventing coercion is the same as for eligibility, i.e., *C* or one of the *n* entities that has the role of *C* must be corrupted. Some verifiability is sacrificed, due to the fact that the number of aliases will not match with the number of registered voters. Only *C* can check if *VS* shows more voters than the ones who have been registered. Privacy is implicated too, because the new signing keys (K'_C) are individual for each voter and therefore he or she can be traced by *C*. In order to prevent this, *VS* has to hide signed messages from *C*. This way, privacy can be broken only if *C* and *VS* collude.

Another detail is that coercer can look for mirror alias sheets in the final document, checking if the sheet values are just the opposite of the stolen alias. On the one hand, the more people who participate, the higher likelihood of finding mirror sheets of true voters. On the other hand, more than one mirror alias can be created at step 'd', so that the addition of all the votes of mirror alias is null and no one mirror alias has the opposite values to the stolen alias.

4.8. Scalability

The complexity of voting systems is a major factor in its practical implementation. An efficient voting scheme has to be scalable with respect to storage, computation, and communication needs as a fraction of the number of voters. In this system no mixnets are needed, avoiding multiple nested encryptions. The most complex computation, in the blind signature, consists of an encryption followed by a product. More users imply more resources in a linear way. Therefore this system is scalable easily.

4.9. Robustness

It is the resistence to attacks from corrupt authorities or voters as well as faults (non-participating authority/voters). In this

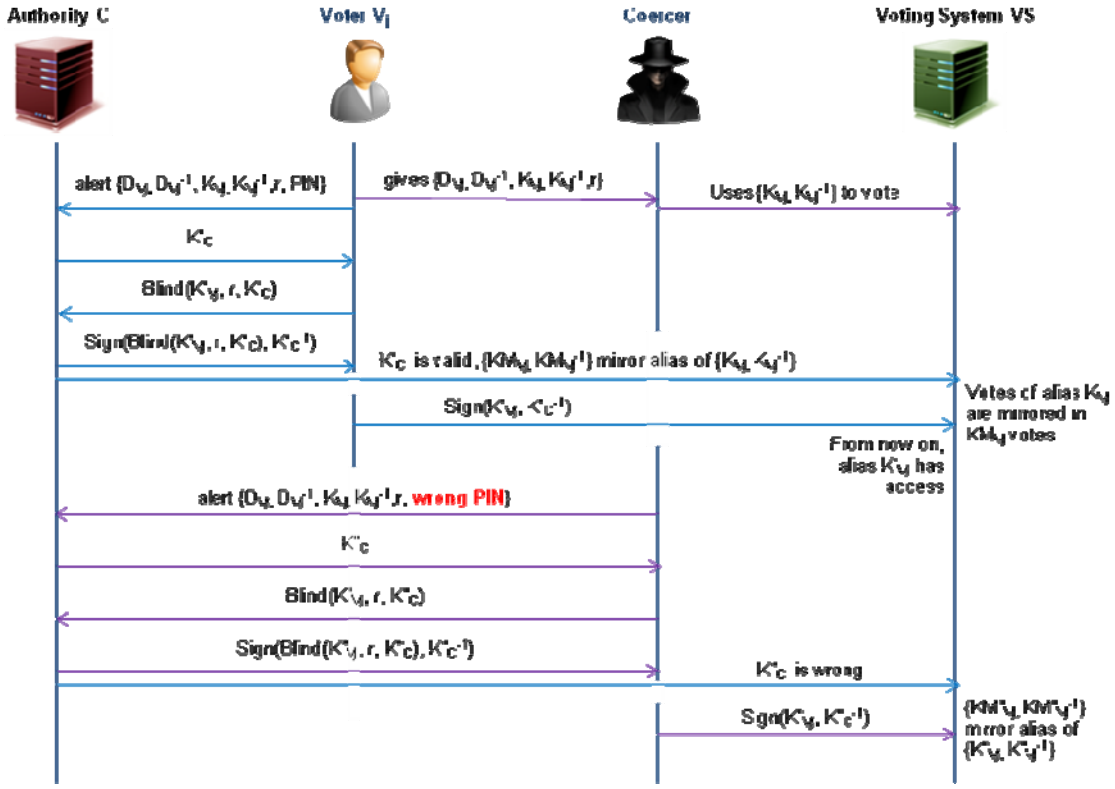

Fig. 6. Behaviour when a coercer stoles voter's information once the alias key has been signed.

scheme the process stops if *C* or *VS* fails. Nevertheless, there are enough transparency to point who fails and when. In these cases, ellection authorities can address the problem. The absence of any voter doesn't affect the poll. Thus, faults from any participant is solved.

Attacks from authorities (*C* or *VS*) have been analyzed along the previous properties, demostrating that *VS* cannot break anyone without beeing detected by *C* or a voter. Attacks from voters can only affect to their own sheet, because system treats them separatedly. *C* is the weakest link because it can authorize fake users. This happens also in any other e-voting system with the entity who guards the census. If multiple entities shares the rol of *C* (improvement detailed in 4.2), only a collusion of all of them can achieve successful attacks. Therefore, the robustness degree of this scheme is defined by the number of entities wich has the rol of *C*, because is the minimum number of corrupted entities needed to break the security. This suggests the limit case when every voter is, at the same time, a *C* entity. This kind of *C*-users nets could be called 'c-nets'. In this situation maximum robustness is achieved: a collusion of all participants is needed to disrupt the poll and a fault of any participant is detected by the rest. These c-nets are suitable for small groups which have to take decisions with a high security degree.

5. Comparison

This section is focused in the nearest schemes to the approach presented here (see table I). One of them is Chaum (1981). In this scheme the authority checks eligibility of voters, and then let them access to a mixnet. At the end, a private key is published on a bulletin board, just like the alias key defined in this work. In a second stage, voters send by means of the same method (authentication plus mixnet) their votes encrypted with the alias key. The bulletin board shows the alias keys and the encrypted votes, so that everyone can open the ballots. This method has some failures (Sampigethaya and Poovendran, 2006): there is no way to resolve collision of keys, the mixnet has vulnerabilities (errors on a mix cannot be detected) and voters don't have proofs to claim and solve individually when problems arise. These weaknesses are not present in our scheme because mixnet has been substituted with blind signature, each step has a proof that guarantees claim capability, and collision of private keys is solved.

The idea of using blind signature for obtaining privacy was applied in Chaum (1988) and in Boyd (1990). However, in these schemes the ballots are directly blinded and thrown (no alias keys are present). With these schemes, the one presented here shares many features: simplicity, maximal privacy, eligibility, individual verifiability and the need of an untraceable channel. Nevertheless, these schemes consider only one entity. Therefore, it could send false ballots instead of any abstainee without being detected. In addition, these schemes inherit weaknesses like the problem of collusions and addressing disputes.

The last scheme close to our method was issued by Fujioka et al. (1993). It uses blind signature on encrypted ballots in a first stage. Next, voters transmit their ballot through an anonymous channel (a mixnet). The ballots are listed in a bulletin board where each voter can check them. If a ballot is right the voter sends the key needed to open it. The similarity consists of the key used at the end to reveal the ballots, which has a similar function as the alias key used to sign the results at the end. The main weaknesses of this scheme are however, accuracy and robustness. Any abstainee can be detected by registration authority, and it could add votes for them. Collusion between voter and registration authority, and token collisions can also create inaccuracies. This method was improved by Baraani-Dastjerdi et al. (1995) and by Juang et al. (2002), increasing the complexity in order to offset the faults.

Table I: Comparison with other schemes

	Closest approaches				Hidden Vote schemes:	Recent Schemes:
	Chaum 1981	**Chaum1988**	**Boyd 1990**	**Fujioka et al. 1993**	**Cramer et al., 1997**	**EVIV, 2013**
Consist of	Alias key that encrypts the vote	Blind signature of ballot	Multiple blind signature of ballot	Blind signature of an encrypted vote. Once verified in the board, alias key is sent to unencrypt	Uses homomorphyc properties to add votes without unencrypting	Encrypted ballot and coded recipt. Homomorphyc tally.
Weaknesses	-Alias key collisions -No way to claim	One entity	Token collision	-Alias collision -One entity	Unable to decision tools, because are inflexible.	Complexity, Rigidity.
Privacy	✓	✓	✓	✓	✓	✓
Elegibility	✓	✓	✓	✓	✓	✓
Verifiability	✓	✓	✓	✓	✓	✓
Dispute-freness	×	×	×	×	×	✓
Fairness	×	×	×	✓	✓	✓
Mobility	✓	✓	✓	✓	✓	✓
Incoercibility	×	×	×	×	p	p
Scalability	✓	✓	✓	✓	×	p
Robustness	×	×	×	×	✓	✓
Advantages of the new scheme	Fairness, Dispute-free, incoercibility, robustness			incoercibility, simplicity	flexibility and scalability	flexibility and scalability
	Open to any network application whose information must be protected (not oriented to ballots)					

Legend: ✓, satisfied; ×, not satisfied; p, partially satisfied.

In addition, the four methods haven't any variant that avoid coercion. Acquisity (2004) presented a method that prevents coercion with certain conditions: the coercer isn't present in the registration phase neither in other key stages. As the voter can submit more than one vote without being traced and coercer cannot confirm token or abstention, a certain degree of incoercibility is assumed. Variant one of this method provides a higher degree of incoercibility since no assumptions are related to coercer except that voter can have at least one chance to connect to the system without being watched, and coercer cannot be with the voter when personal certificates are delivered.

Hidden vote schemes operate with the encrypted votes. This means that votes must share the same format. Therefore these schemes are not suitable for flexible debate applications. Moreover the scalability of such systems is worse. One example of this kind of schemes is present in the comparative table (Cramer et al., 1997), which shows these disadvantages. On the other hand, it has strong robustness because in the process a shared public key is issued by a set of entities. This is achieved in such way that more than one entity has to collude or fail to break the security of the system.

Finally, EVIV scheme (Joaquim et al., 2013) has been considered. There are two reasons: it is a recent method and it has been specifically designed to work on the internet. EVIV needs a smart card that encrypts the ballot and a receipt. Once voter verifies his/her choice, both of them are sent to the bulletin board. The scheme makes use of MarkPledge technique (Joaquim and Riveiro, 2012) that avoid corrupt ballots. At the end, homomorphic properties help to reveal the tally without showing individual votes. This scheme accomplishes quite a few features of e-voting methods. Nevertheless, it's another hidden vote scheme that requires a fixed format; therefore it isn't flexible. Voters could change their vote once cast (with some modifications in the method), but coercer could notice that change in the bulletin board. In other words, incoercibility is partial and weak. Although MarkPledge has evolved to be faster, it implies cryptographic computations that increase with the number of voters. In addition, a debate implies more than a question to vote. Thus, multiple voters and multiple questions will imply heavy computational requirements for the server, which affect scalability.

6. Conclusions

The scheme presented here is, as far as we know, the first one that provides an open frame to allocate any network application. Particularly, it can support the recent debate tools where users contribute with opinions, proposals and votes. It is designed to provide an anonymous access to users who can exchange information with a server in a protected way. In addition, it allows each user to validate his/her contributions to the community, regardless of the format or amount of them. This can be carried out in a simple way, which makes the system scalable, and especially suitable for the Internet.

As with most e-voting schemes, this one achieves privacy, eligibility, verifiability and mobility. Not so common are fairness, accuracy, dispute-freeness, scalability and robustness, which are difficult to find together in the same degree. This scheme accomplishes all of them without sacrificing simplicity. In addition, incoercibility can be obtained with some modifications, trading privacy and verifiability. The special case of c-nets reaches maximum robustness where all participants must collude to break the process, and is suitable for small groups.

References

Acquisti A. Receipt-free homomorphic elections and write-in ballots. Cryptology ePrint Archive, Report 2004/105, <http://eprint.iacr.org/>; 2004
Adida, B. Helios: Web-based Open-Audit Voting. https://www.usenix.org/legacy/events/sec08/tech/full_papers/adida/adida.pdf (2014).
Asadpour M, Jalili R. Double voting problem of some anonymous e-voting schemes. J Inf Sci Eng 2009;25(3):895–906.
Baseri Y, Mortazavi AS, Asaar MR, Pourpouneh M, Mohajeri J. Double voter perceptible blind signature based electronic voting protocol. ISC Int J Inf Secur 2011;3(1):43–50.
Bellare, M., Namprempre, C., Pointcheval, D., & Semanko, M. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology (2003), 16(3), 185-215.
Benaloh J. Verifiable secret-ballot elections, Ph.D. thesis, Yale University; 1987.
Benaloh, J. Simple Verifiable Elections. In EVT'06, Proceedings of the First Usenix/ACCURATE Electronic Voting Technology Workshop, August 1st 2006, Vancouver, BC,Canada.
Benaloh, J,Tuinstra,D. Receipt-freesecret-ballotelections.In:26thannualACM symposium onthetheoryofcomputing,pp.544–553; 1994.
Benn, N., & Macintosh, A. Argument visualization for eParticipation: towards a research agenda and prototype tool. Electronic Participation (2011) 60-73.
Boyd C. A new multiple key cipher and an improved voting scheme. In: Advances in cryptology – EUROCRYPT '89. Springer-Verlag; 1990. p. 617–25.
Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 1981;24(2):84–8.
Chaum, D. Blind signature system. In: Advances in cryptology – CRYPTO '83. Plenum Press; 1984. p. 153.
Chaum, D. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In: Advances in cryptology – EUROCRYPT '88. LNCS, vol.330. Springer Verlag; p. 177–82. 1988a
Chaum, D. The dinning criptographers: unconditional sender and recipient untraceability. Journal of Cryptology, 1988b; 1(1):65–75
Chen YY, JanJK, Chen CK. The design of a secure anonymous internet voting system. Computer & Security 2004;23:330–7.
Chien HY, Jan JK, Tseng YM. Cryptanalysis on Mu–Varadharajan's e-voting schemes. Appl Math Comput 2003; 139(2):525–30.
Cramer Ronald, Gennaro Rosario, Schoenmakers Berry. A secure and optimally efficient multi-authority election scheme. In: Advances in cryptology – EUROCRYPT '97. LNCS, vol. 1233. Springer-Verlag; 1997. p. 103–18.
Cranor, L., Cytron, R. Sensus: a security-conscious electronic polling system for the Internet, Proceedings of the Thirtieth Hawaii International Conference on System Sciences, vol. 3, 1997, pp. 561– 570.
Cucurull, J., Álvaro, A., & Puiggalí, J. MyUniversity: E-Participation and Decision Making for Higher Education. EGOV/ePart Ongoing Research 2013, 218-225.
Dingledine R., Mathewson, N., & Syverson, P. Tor: The Second-Generation Onion Router. In the Proceedings of the 13th USENIX Security Symposium, August 2004.
Fujioka, A. Okamoto, T. Ohta, K. A practical secret voting scheme for large-scale elections. Advances in Cryptology, AUSCRYPT'92, Lecture Notes in Computer Science 718 (1993) 244– 251.
Gordon, T., Prakken, H., & Walton, D. The Carneades model of argument and burden of proof. Articial Intelligence 2007, 171, 875-896.
Herrmann, M. & Grothoff, C. Privacy-implications of performance-based peer selection by onion-routers: a real-world case study using I2P. In Privacy Enhancing Technologies 2011; (pp. 155-174). Springer Berlin Heidelberg.
Ibrahim, S., Kamat, M., Salleh, M., & Aziz, S. R. A. Secure E-voting with blind signature. In 4th National Conference on Telecommunication Technology Proceedings, 2003. NCTT 2003. (pp. 193-197). IEEE
Jakobsson M, Juels A, Rivest R. Making mix nets robust for electronic voting by randomized partial checking. In: Proceedings of USENIX security '02 2002. p. 339–53.
Joaquim R., Ribeiro C. An efficient and highly sound voter verification technique and its implementation. In: Kiayias A, Lipmaa H, editors. E-voting and identity.vol. 7187 of Lecture notes in Computer Science. Berlin/Heidelberg: Springer; 2012. p. 104e21.
Joaquim, R., Ferreira, P., & Ribeiro, C. EVIV: An end-to-end verifiable Internet voting system. Computers & Security (2013), 32, 170-191.
Juels, A., Catalano, D., and Jakobsson, M. Coercion-resistant electronic elections. In Vijay Atluri, Sabrina De Capitani di Vimercati, and Roger Dingledine, editors, WPES, pages 61–70. ACM, 2005
Juels A, Jakobsson M. Coercion-resistant electronic elections. Cryptology ePrint Archive, Report 2002/165, <http://eprint.iacr.org/>; 2002.
Kiayias Aggelos, Yung Moti. The vector-ballot e-voting approach. In: Financial cryptography. LNCS, vol. 3110. Springer-Verlag; 2004. p. 72–89.
Klein, M. The mit deliberatorium: Enabling large-scale deliberation about complex systemic problems. Collaboration Technologies and Systems (CTS), 2011. International Conference, 161-161
Lee B, Boyd C, Dawson E, Kim K, Yang J, Yoo S. Providing receiptfreeness in mixnet-based voting protocols. In: Proceedings of the ICISC '03, 2003. p. 261–74.
Lin IC, Hwang MS, Chang CC. Security enhancement for anonymous secure e-voting over a network. Comput Stand Interfaces 2003; 25(2):131–9.
López, D., Mateo, T., & Cortés, E. Enhancing Learning Through a Novel Decision Tool. VIII Conferencia Iberica de Sistemas e Tecnologías de Informacao 2013 (CISTI'13), Actas, 2, 49-52.
Mateu, V. Sebé, F. Valls, M. Constructing credential-based E-voting systems from offline E-coin protocols Journal of Network and Computer Applications 42(2014) 39–44.
Mu Y, Varadharajan V. Anonymous secure e-voting over a network. In:14th Annual computer security applications conference (ACSAC). IEEE Computer Society 1998; p.293–9.
Ribeiro C., Joaquim, R. An Efficient and Highly Sound Voter Verification Technique and Its Implementation. E-Voting and Identity, 3º Int. Conf. VoteID. 2012 Springer-Verlag. pp. 104–121.
Rodríguez-Henríquez F, Ortiz-Arroyo D, García-Zamora C. Yet another improvement over the Mu–Varadharajan e-voting protocol. Comput Stand Interfaces 2007; 29(4): 471–80.
Sako, K. & Kilian, J. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In EUROCRYPT 1995, pages 393–403.
Sampigethaya, K. Poovendran, R. A framework and taxonomy for comparison of electronic voting schemes. Computers & Security 25 (2006) 137–153
Snader, R. & Borisov, N. A Tune-up for Tor: Improving Security and Performance in the Tor Network. In NDSS 2008 (Vol. 8, p. 127).
Schneider, D. C., Voigt, C., & Betz, G. Argunet – A software tool for collaborative argumentation analysis and research. CMNA VII - Computational Models of Natural Argument 2007, 10, 57-61.
Schweisgut, J. Coercion-resistant electronic elections with observer. In: Second International workshop on electronic voting (2–4 Aug2006), Bregenz.
Smith, WD. New cryptographic voting scheme with best-known theoretical properties. In: Workshop on frontiers of electronic elections, Milan, Italy; 2005.
Tiropanis, T., Davis, H., Millard, D., & Weal, M. Semantic Technologies for Learning and Teaching in the Web 2.0 Era. Intelligent Systems, IEEE 2009, 24(6), 49-53.
Tiwari, A., Mewada, R., Mehta, K., & Mohite, V. I-Voting: Democracy comes home. International Journal of Research in Advent Technology 2013, 1(4), 42-50.
Verdejo, M. F., Celorrio, C., Lorenzo, E. J., Millán, M., Prades, S., & Vélez, J. Constructing mobile technology-enabled environments for an integrated learning approach. Innovative Mobile Learning: Techniques and Technologies 2008, (8), 147-171.

Weber, S. A coercion-resistant cryptographic voting protocol —evaluation and prototype implementation, Darmstadt University of Technology, Department of Computer Science Cryptography and Computer algebra, Diploma Thesis; July 2006.
Wu, Z. Y., Wu, J. C., Lin, S. C., & Wang, C. An electronic voting mechanism for fighting bribery and coercion. Journal of Network and Computer Applications 2014, 40, 139-150.
Yang CC, Lin CY, Yang HW. Improved anonymous secure e-voting over a network. Inf Secur 2004; 15(2):185–91.

Links

[1] http://www.argunet.org/
[2] http://www.debategraph.org/
[3] http://carneades.github.io/
[4] https://www.bcisiveonline.com/
[5] http://www.takeonit.com/
[6] http://www.ivotingtool.com/
[7] https://angel.co/debate-fm
[8] https://agoravoting.com/
[9] http://cci.mit.edu/klein/deliberatorium.html
[10] http://www.myuniversity-project.eu/
[11] http://freehaven.net/anonbib/