New voter verification scheme using pre-encrypted ballots

Victor Morales-Rocha a,*, Miguel Soriano b,c, Jordi Puiggalí a

a Scytl Secure Electronic Voting, Tuset 20, 1-7, Barcelona, 08006 Barcelona, Spain
b Universitat Politècnica de Catalunya, Departament d’Enginyeria Telemàtica, Jordi Girona, 1-3, Barcelona, Spain
c Centre Tecnològic de Telecomunicacions de Catalunya (CTTC). Av. Canal Olímpic S/N 08860 - Castelldefels, Barcelona, Spain

ARTICLE INFO

Article history:
Received 21 January 2009
Received in revised form 27 March 2009
Accepted 28 March 2009
Available online 2 April 2009

Keywords:
Electronic voting
Internet voting
Voter verification
Pre-encrypted ballots
Voting receipt

ABSTRACT

In order for remote electronic voting systems to be reliable, voter verification is an essential feature. Most pre-encrypted ballot schemes proposed to date allow voters to verify that their votes have been properly recorded in the voting server. Such verification makes it possible to detect manipulation of the vote in the voting terminal or during its transmission. However, these schemes do not allow voters to verify that their votes have been accurately included in the counting process. In this paper, we describe a new proposal that solves this problem. Under the proposed scheme, voters can verify that their votes have been properly recorded, and they also have the opportunity to individually verify that their votes have been properly included in the counting process. Voter verification is thus achieved in two different contexts: proper recording and accurate counting.

1. Introduction

An electronic voting system is reliable if it has certain security features, such as privacy and accuracy. To preserve the accuracy of election results, it is not enough for a system to properly record and count votes. There are external factors that could compromise the content of votes and therefore the accuracy of election results. In [1], Rubin highlights potential security risks in remote electronic voting systems. Most of these risks are due to the voter environment, which is not controlled by the voting system. The voter environment is composed of the voting terminal and the communication channel between the terminal and the voting server.

In conventional paper voting systems, voters are able to verify that their votes are properly recorded, since they put the paper ballots in the ballot box. However, they are unable to verify that their votes are counted.

Electronic voting systems can provide voters with the means to verify that their votes are handled properly during the counting process. If voters can verify that their votes have been counted, the reliability of the voting system increases. Several voting systems that address the verification issue have been proposed (for example [2,3,6,13,14]). However, these systems have had problems fulfilling both aspects of verification: proper recording and accurate counting.

We propose a new voter verification scheme that allows voters to verify both proper recording and accurate counting. The scheme is based on pre-encrypted ballots with verification codes, which voters can use to verify that their votes have been properly received by the voting server. Our solution also includes cryptographic voting receipts. This component allows voters to verify that their votes have been accurately included in the tally. In short, our proposal combines pre-encrypted ballots with cryptographic voting receipts in order to allow both kinds of verification.

The next section describes the state of the art of voter verification. Section 3 presents our proposal. Section 4 presents a security analysis and comparison. Finally, Section 5 highlights some conclusions.

2. Basis of voter verification

Pre-encrypted ballot schemes are aimed at preventing malware from manipulating votes in the voting terminal. Such schemes are also useful in verifying that votes have been properly recorded at the voting server. The verification of accurate vote counting is often done using cryptographic voting receipts. Since our solution is based on both pre-encrypted ballots and cryptographic voting receipts, we briefly describe both kinds of voter verification solutions in the following sections.

2.1. Pre-encrypted ballot schemes

These are also known as voting code schemes. Their main characteristics are as follows:
• For each voter, the candidates or voting options are represented by different alphanumeric codes (i.e. the same candidate has a different code for each voter). These voting codes are printed on a ballot that is delivered to the voter prior to the voting phase. Some schemes also include a verification code that corresponds to each voting code. In such cases, the verification codes are printed on the ballot along with the corresponding voting codes, as shown in Fig. 1.

• The electoral authority sends the pre-encrypted ballots to the voters through a secure independent communication channel, e.g. postal mail.

• These schemes are not based on cryptographic means on the voter side. Therefore, voters do not need to trust the software that carries out the cryptographic vote-protection operations before votes are cast.

• The voter’s choice is expressed by means of a code. Therefore, even if an attacker were to intercept a vote at casting time, she would be unable to ascertain the voter’s choice.

• In schemes that include verification codes, voters can be sure that the votes have reached the voting server without alteration. This is done by checking the verification code received from the voting server. It is therefore infeasible that anyone could carry out a successful malware attack that changes a voter’s choice in such a way that the new vote is another valid vote.

As mentioned above, in pre-encrypted ballot schemes that include verification codes, voters can be sure that their votes have been properly recorded by the voting server. However, the schemes proposed to date have offered no efficient and secure way of verifying that an individual vote has been included in the final tally.

Pre-encrypted ballot schemes face certain kinds of threats, and this can be a reason to avoid such schemes. The attacks fall into three main categories:

• **Ballot manipulation.** An attacker who intercepts a pre-encrypted ballot before it is received by a voter can reassign the pairs of codes (voting and verification codes) to different candidates. If the voter does not notice the change in the ballot, she will cast a different vote than she intended. Since the voter receives the verification code that corresponds to the voting code cast, she would not be able to detect that the vote received by the server is different from her intention. Fig. 2 shows an example of this attack. The success of this attack depends on several factors. For example, to get a direct advantage, the attacker would have to know the vote’s code in order to be sure that it is in the voter’s interest to change the voting codes. In elections with three or more candidates, however, it is possible to manipulate pre-encrypted ballots to favor a specific candidate. Specifically, the attacker would switch the main opponent’s pairs of codes with those of one or more of the minority candidates. The main opponent therefore loses votes and the favored candidate receives an advantage. In any case, the security measures employed during the generation and distribution of the pre-encrypted ballots affect the success of these attacks.

• **Privacy.** An attacker who has access to the pre-encrypted ballot could compromise voter privacy by linking the vote to the voter. However, to carry out this attack, particular conditions must be fulfilled. In order to execute this attack, the attacker needs access to the following:
  - The pre-encrypted ballot. Access to the pre-encrypted ballot can take place in two different contexts: a) during the generation and distribution of ballots, or b) in the voter environment.
  - The voting server. The attacker could access the voting server by collaborating with: a) the electoral authority in charge of the voting server, or b) the technical staff involved in the election.

For such an attack to be possible, however, it is assumed that there are no security measures on access control.

Another way to link a vote to a voter is to gain access to the pre-encrypted ballot and capture the voting code during vote casting.

• **Coercion or ballot selling.** As in all remote voting systems, there is a risk of coercion or ballot selling. However, under pre-encrypted ballot schemes, this risk could be increased if voting codes were published after being cast or if the coercer or vote buyer were to obtain access to that information.

Several pre-encrypted ballot schemes have been proposed in recent years, for example [2–11].

### 2.2. Voting receipts

Other voter verification methods for remote voting systems are based on the use of cryptographic voting receipts. Voting receipts allow voters to verify that their votes have been included in the tally. If a voter detects that her vote has not been counted, she can complain to the corresponding authority by showing her voting receipt. The voting receipt usually contains a cryptographic proof that must be secure enough to prevent a third party from retrieving the vote.

According to [19], a voting receipt scheme must be resistant to the following:

• **Manipulation.** Nobody can alter the contents of the voting receipt without detection.

• **Repudiation.** Both the voters and the election authority can verify the authenticity of the voting receipt. Once a voting receipt is issued, the election authority cannot deny its validity or authorship of it. Likewise, once a voter receives a voting receipt, she cannot deny that she has received it and that it corresponds to the vote cast.

• **Generation of bogus receipts.** Neither the election authority nor the voters can generate, without detection, voting receipts that correspond to an eligible vote.

Voting schemes that include voting receipts have been proposed in [12–14]. In general terms, these schemes allow the voter to verify that a vote, presumably her vote, has been included in the final tally. This verification is not reliable because the voter cannot verify that the vote stored by the voting server actually reflects her intent. This is because the voting terminal is prone to a variety of
attacks (e.g. by malware) that can change voter intent before the vote is encrypted and cast.

Recently, a new voting system named Helios has been presented and implemented [21]. This system introduces verification mechanisms that allow a voter to audit her vote (such as a voting receipt). Helios uses a mixnet to anonymize the votes. It also includes mechanisms for verifying the mixnet operation, decryption, and tallying of an entire election. Nevertheless, it can not be used today in big elections due to timing requirements derived of its computational cost. Table 1 shows the timings of a testing carried out with Helios [21]. Furthermore, because of the fact that in the Helios system the voter privacy is not guaranteed, its use is restricted to elections with low possibility of coercion.

3. Description of the proposal

We propose an Internet voting scheme that can be used in large-scale elections with multiple voting channels, i.e. along with other simultaneous voting channels. We believe that pre-encrypted ballot schemes provide significant advantages in terms of voter verification. We have therefore based this proposal on such a scheme.

In the voting phase, the voter sends the voting code corresponding to the chosen candidate. The voter then receives a verification code that corresponds to the voting code. This allows the first verification task to be carried out, as shown in Fig. 3. During the voting phase, in addition to the voting code, the voter sends data that is used to generate a cryptographic voting receipt that is digitally signed by the voting server. The voting receipt contains a verification proof that allows the voter to check whether her vote has been included in the tally. In this scheme, therefore, the vote stays unalterable from the moment it is received by the voting server until it is counted, and the voter can check that the vote has been both properly recorded and accurately included in the tally. The scheme requires that the voting terminals have sufficient capacity to carry out certain cryptographic operations. For this purpose, conventional personal computers would suffice.

In this scheme, the main activities for each election phase are as follows:

- **Pre-election phase**
  - Generation of the pre-encrypted ballots
  - Printing of the voter information
  - Distribution of the pre-encrypted ballots and voter information

- **Voting phase**
  - Generation of the vote and voting receipt
  - Sending of the vote and voting receipt
  - Validation of the voting receipt and calculation of the verification code
  - Voter verification using the verification code

- **Counting phase**
  - Decryption of the voting codes
  - Counting and publication of results
  - Voter verification using the voting receipt

These activities are detailed below. Table 2 describes the nomenclature used in the description of the protocol.

3.1. Pre-election phase

3.1.1. Generation of the pre-encrypted ballots

In contrast to conventional cryptographic schemes for electronic voting, in which cryptographic functions are mainly applied during the voting phase, in a pre-encrypted ballot scheme the main cryptographic functions are carried out during the generation of the ballots, i.e. in an earlier phase.
The generation of the pre-encrypted ballots is a very important aspect of the security of this voting scheme. It is therefore necessary to define some measures and procedures to guarantee a secure environment during their generation. For instance, it is important to generate the ballots in an isolated environment in order to prevent intrusions that could alter the process or obtain the information being generated.

The voting codes must be unique and their length must strike a balance between security and usability. The verification codes must also be unique, but their length is more flexible since the balance between security and usability. The verification codes are generated as follows:

1. Generation of unique ballot identifiers (B-Idi). This is done randomly. The ballot identifier is a numerical value that is converted to its binary representation in order to be operated with other values.
2. Generation of unique candidate identifiers (C-Idj). A numerical value representing the candidate identifier is assigned to each candidate. The candidate identifiers are also converted to their binary representation.
3. For each ballot identifier B-Idi:
   - Two random binary values Ai and Bi are generated. These values are used as seeds in the generation of voting and verification codes.
   - For each candidate identifier:
     - B-Idi and C-Idj are XORed, thereby obtaining an intermediate value Interi:

\[
\text{Inter}_{ij} = (B - \text{Id}_i \oplus C - \text{Id}_j)
\]

- Interi,j and Ai are XORed, thereby obtaining the voting code Codi,j:

\[
\text{Cod}_{i,j} = (\text{Inter}_{i,j} \oplus A_i)
\]

- A random symmetric key k1 is generated.
- The value A is encrypted with the key k1. The key k1 is then encrypted with the public key P-CS, which belongs to the electoral authority in charge of the counting process (CS). The result of these two operations is a digital envelope to protect A:

\[
KA_i = [A(k1); (k1); P-CS]
\]

- A verification code Veri,j corresponding to each voting code Codi,j is generated:
  - B-Idi and Codi,j (their binary representation) are XORed:
  - B, and the previously obtained value are XORed. The result of these operations is the verification code:

\[
\text{Ver}_{ij} = [B \oplus (B - \text{Id}_i \oplus \text{Cod}_{i,j})]
\]

- A random symmetric key k2 is generated.
- The value Bi is encrypted with the key k2. Then, k2, is encrypted with the public key P-VS, which belongs to the voting server (VS). These operations are aimed at protecting the value B:

\[
KB_i = [(B(k2); (k2); P-VS]
\]

4. The value KAi and its corresponding B-Idi; [(B-Id1, KA1), (B-Id2, KA2), ... (B-Idn, KAn)] are transferred to the counting authority. This is done at the end of the ballot-generation process. The value Ai is used by the counting authority during the decryption of codes, as described below.

5. The value KBi is stored in the voting server along with the corresponding B-Idi; [(B-Id1, KB1), (B-Id2, KB2), ... (B-Idn, KBn)]. The value Bi is decrypted and used by the voting server to calculate the verification code, as explained later.

Fig. 4 shows the process of generating voting and verification codes.

Some of the values generated during ballot generation could be used maliciously to violate voter privacy. Therefore, once used, they must be eliminated. These values are Interi,j, Ai and Bi.

The private key used to decrypt the voting codes (S-CS) is split into shares in order to keep it safe. Because the counting authority is usually composed of several parties, each member should hold a share of the key. In this way, the voting codes can only be decrypted through the cooperation of all members, or a predefined subset of members, as in [16,17,20]. The private key is split using a secret-sharing scheme, as described in [18].

In order to prevent an attacker from accessing and controlling the voting and verification codes, the verification codes are not printed on the pre-encrypted ballot. Instead, voters can access the verification codes through a website. This prevents ballot manipulation attacks attempting to change voter intent. To access such a website, a password is generated and associated with the ballot identifier.

Once the necessary elements have been generated, the pre-encrypted ballots are created by printing the following elements:

- Ballot identifier, B-Idi,
- Candidates’ names, parties, affiliations, etc.
- Voting codes.
- Password to get the verification codes (preferably covered with a latex surface).
The pairs of candidates’ names and voting codes are printed in a random order. This randomness prevents some privacy and coercion attacks, as shown later. Fig. 5 shows the pre-encrypted ballot.

3.1.2. Printing of the voter information

Voter information is printed on a separate sheet. The following elements make up the voter information:

- Name
- Address
- Other voter or election information
- Information about how to get the verification codes

3.1.3. Distribution of the pre-encrypted ballots and voter information

The pre-encrypted ballots are assigned randomly. Packages containing a pre-encrypted ballot and a sheet of voter information (without relation to one another) are assembled. Security procedures are required to keep this assignment anonymous. For example, once the sheets of voter information are printed, each one is sealed and then assigned to a package. Additionally, a software tool that makes the task of entering voting codes easier could be delivered to voters along with the packages.

The packages are delivered to voters through a communication channel that is presumably secure, for instance postal mail or another delivery service. In addition to the security measures described above for the generation and distribution of pre-encrypted ballots and voter information, other security characteristics can be employed. For example, special paper with embedded security elements could be used to print the ballots.

Once the voter has received the pre-encrypted ballot and voter information, she can obtain the verification codes by accessing the specified website. The voter enters her ballot identifier and the associated password. The verification codes are then shown in the same order as the voting codes on the pre-encrypted ballot, so that voters can easily make the relation between their voting and verification codes. The password prevents people from getting verification codes by randomly entering ballot identifiers.

3.2. Voting phase

At this point, the voter has the pre-encrypted ballot and her verification codes. The communication channel between the voting
terminal and the voting server is protected by using the TLS protocol. It is assumed that voters have a digital certificate to authenticate themselves in order to access the voting platform. The steps to cast a vote and generate a voting receipt are described below.

### 3.2.1. Generation of the vote and voting receipt

1. The voter chooses her vote by means of the corresponding voting code $\text{Cod}_{ij}$.
2. A unique identifier $R$-$Id_i$ is randomly generated.
3. A hash value of the unique identifier $R$-$Id_i$ and the election identifier $E$-$Id$ is calculated. This value is then digitally signed with the voter’s private key. This information is used to generate a voting receipt $r_i$:
   \[ r_i = [H(R - Id_i \mid E - Id)]_{SV} \]
4. The voter generates a secret $S$, which consists of a hash of the voting code and the unique identifier. This secret is later used to check if the vote is included in the tally. The secret is digitally signed and encrypted with the public key of the counting authority:
   \[ S_i = [H(\text{Cod}_{ij}) \mid R - Id_i]_{P - CS} \]

### 3.2.2. Sending of the vote and voting receipt

The voter sends the voting message $V_i$ to the voting server:
\[ V_i : \text{Cod}_{ij} \mid r_i \mid S_i \mid B - Id_i \]
where:
- $\text{Cod}_{ij}$ is the voting code that indicates the voter’s preference.
- $r_i$ is the receipt information, which is digitally signed by the voting server as a proof of voting receipt.
- $S_i$ is the voter secret, which is later used to check whether the vote has been included in the tally.
- $B$-$Id_i$ is the ballot identifier, which the voting server uses to obtain the value $B$, and calculate the corresponding verification code.

### 3.2.3. Validation of the voting receipt and calculation of the verification code

Upon receipt of $V_i$, the following tasks are carried out in the voting server:

1. Verification of voter eligibility through the digital signature on $r_i$.
2. Validation of the voting receipt $R_i$ by digitally signing $r_i$:
   \[ R_i = [r_i]_{SV} \]
3. Calculation of the verification code $\text{Ver}_{ij}$ that corresponds to the voting code $\text{Cod}_{ij}$:
   - The voting server decrypts the value $B_i$ that corresponds to the received $B$-$Id_i$.
   - The voting server then calculates:
     \[ \text{Ver}_{ij} = [B_i \oplus (B - Id_i \oplus \text{Cod}_{ij})] \]
4. The voting receipt $R_i$ and the verification code $\text{Ver}_{ij}$ are sent to the voter.
5. The voting code $\text{Cod}_{ij}$ is encrypted with the public key of the counting server in order to protect it during the voting phase:
   \[ \text{Cod}_{ij}' = [\text{Cod}_{ij}]_{P - CS} \]
6. The values $\text{Cod}_{ij}'$, $S_i$, and $B$-$Id_i$ are digitally signed by the voting server and stored along with $R_i$:
   \[ [\text{Cod}_{ij}', S_i, B - Id_i]_{SV} \]

### 3.2.4. Voter verification using the verification code

The voter receives the verification code $\text{Ver}_{ij}$ and checks whether it corresponds to the voting code that she cast (verification of proper recording). If it does, then the voter stores or prints the voting receipt $R_i$ and her voting session is finished.

**Fig. 6** shows the steps taken in the voting phase.

#### 3.3. Counting phase

#### 3.3.1. Decryption of the voting codes

The first step in the counting phase is the decryption of the voting codes to determine which candidates they are associated with. The following steps are carried out to decrypt the voting codes:

1. The members of the counting authority download the set of voting codes and voter secrets stored in the voting server.
2. The encrypted voting codes $\text{Cod}_{ij}'$ are decrypted using the counting authority’s private key.
3. The secrets $S_i$ are decrypted to get the hash values of the voting codes. This decryption is also done using the counting authority’s private key.
4. The hash value of each voting code is calculated in order to compare it with the hash value obtained from the secret. In this way, it is possible to verify whether the voting codes have been kept inalterable.
5. The value $R$-$Id_i$ obtained from the secret is compared with the $R$-$Id_i$ included in $r_i$.
6. The voting codes are decrypted to get the corresponding candidate. This is done as follows:
   - The voting codes $\text{Cod}_{ij}$ are converted to their binary representation.
   - The value $A_i$ is decrypted using the counting authority’s private key. To be more specific, this key allows the decryption of the secret key needed to decrypt $A_i$.
   - $\text{Cod}_{ij}$ and its corresponding $A_i$ are XORed to get the value $\text{Inter}_{i}$.
   - $\text{Inter}_{i}$ and the corresponding $B$-$Id_i$ are XORed to get $C$-$Id_i$.

**Fig. 7** shows the steps taken to decrypt the voting codes.

#### 3.3.2. Counting and publication of results

The candidate identifiers obtained in the previous step are associated with the candidates’ names. The votes are then counted. The election results and the list of $R$-$Id$s obtained from the voters’ secrets are published on a website.

#### 3.3.3. Voter verification using the voting receipt

Once the election results have been published, the voter verifies that her $R$-$Id_i$ is included in the list of $R$-$Ids$. This proves that the vote has not been altered since it was cast. If her $R$-$Id_i$ does not appear in the list, the voter can complain to the election authority by showing her voting receipt $R_i$.

#### 4. Security analysis and comparison

This section describes the main threats to the security of Internet voting and pre-encrypted ballot schemes and how they can be mitigated or diminished.
4.1. Malware

One of the main concerns in Internet voting is the risk of malware in the voting terminal, which is usually a personal computer. Malware installed on the voter’s computer could unveil or change the intention of the voter without detection. In our scheme, as in most pre-encrypted ballot schemes, if a voting code is captured by malware, the attacker is unable to know the voter’s real intent. Moreover, if the voting code is modified in the voting terminal, the probability of finding a valid voting code is minimal. Furthermore,
if the modification of the voting code does result in another valid voting code, it is detected when the voter receives the verification code. Therefore, an attacker who attempts to manipulate voting codes by installing malware in order to gain an advantage for a particular candidate would need to know which voting code represents that candidate. Since that voting code is unique for each voter, the attack is unfeasible.

4.2. Ballot manipulation

Ballot manipulation, i.e. an attempt to cast a different vote than the one intended by the voter (without him realizing it), is prevented by separating the pairs of codes. As described above, the voting codes are printed on the ballot and the verification codes are only accessible through a website that requires a password. An attacker who intercepts the pre-encrypted ballot can therefore only change the order of the voting codes; however, such manipulation would be detected once the voter sends the voting code and receives a verification code that does not correspond to the chosen candidate.

4.3. Privacy

The conditions necessary to carry out a successful attack that violates voter privacy were described in Section 2. In this scheme, the random and anonymous assignment of the pre-encrypted ballots diminishes the possibility of voter privacy being violated. Furthermore, our scheme can prevent other attacks that attempt to unmask voter intent. As described in Section 3.2.3, voting codes are encrypted as soon as they are received by the voting server. This encryption is done mainly because the voting server is online during the voting phase and therefore exposed to remote attacks. This encryption prevents attackers from obtaining voting codes, at least while they are stored (and encrypted) on the voting server.

4.4. Coercion

Once voter privacy is kept safe by means of the techniques described above, coercion attacks are considerably diminished. Moreover, in order to prevent coercion in the event that a coerced voter was to get hold of the pre-encrypted ballots, no list of the voting codes that are cast is published. There does remain the risk of a classic coercion attack on the remote voting system, but this requires that the coerced voter be physically present while the voter casts her vote. However, the possibility of carrying out a large-scale coercion attack is eliminated by the techniques described in this proposal.

4.5. Usability

Voting systems based on pre-encrypted ballots are not the conventional way to vote. A scheme like the one proposed here may not be the best option in terms of usability. In conventional electronic voting schemes, the voter chooses a candidate by clicking on the name or picture of her favorite candidate. In contrast, pre-encrypted ballot schemes require voters to enter the voting code that corresponds to the chosen candidate. This situation could be difficult or uncomfortable for some voters. However, a usability study has shown a good level of acceptance among voters in general [15].

In order to evaluate the benefits of the proposed scheme, we offer a comparison. Table 3 compares our proposal with two previous schemes. The first voting scheme included in this comparison is the most significant scheme based on pre-encrypted ballots [8]. The second scheme includes a cryptographic voting receipt [13]; it has the same characteristics as most schemes that feature such receipts.

5. Conclusions

The proposed scheme addresses voter verification issues, i.e. the verification of proper recording and accurate vote counting. Using the verification code, voters can be sure that their votes have been properly recorded. Using the cryptographic voting receipt, they can verify that their votes have been included in the tally. Thus, our remote voting scheme considers a complete voter verification cycle, from the reception of the vote by the voting server to the inclusion of the vote in the tally.

In this Internet voting scheme, malware inserted in the voting terminal cannot by itself carry out a successful attack that attempts to manipulate the vote or unmask voter intent. Furthermore, manipulation of the pre-encrypted ballots is prevented by separating the voting and verification codes. The voting codes are printed on the ballot, whereas the verification codes are only accessible through a website that requires the ballot identifier and password. In order to protect voter privacy, various aspects have been considered. The pre-encrypted ballots are assigned to voters through a random process that prevents malicious authorities from keeping records of the codes associated with individual voters. Additionally, the voting codes are encrypted once they are received by the voting server. This precaution protects the voting codes while they are stored in the voting server.

Acknowledgements

This work has been supported partially by the Spanish Research Council (CICYT) Projects TSI2005-07293-C02-01 (SECONNET), TEC-2008-06663-C03-01 (P2PSec) and by the Spanish Ministry of Science and Education with CONSOLIDER CSD2007-00004 (ARES).

References


