

Internet voting: a utopia?



Adrià Vilanova Martínez

ICT Department
2nd of Baccalaureate

St. Paul's School, Barcelona

October 28, 2016

A research project submitted to St. Paul's School in partial fulfilment of the requirements for the completion of the baccalaureate studies.

Acknowledgements

I would like to thank a lot of people who have helped me complete this research project, and that without their help, this project would never have been completed. First of all, thanks to all my friends (especially Jan Escorza and Sarah Gomez), my colleagues in the Google Product Forums (especially fpardo), and my family who, after hearing about my project, have been asking me questions and helping me have a greater perspective of the voting system which has led to a lot of improvements to it. Secondly, thank you to the 2nd of baccalaureate ICT class for helping me test drive the voting system with them, which was done smoothly and their collaboration has been of great importance to this project. And last, but not least, thanks to Enric Ventura, my tutor, who has guided me through the making of this project, and Hannah Margrett, who offered to correct my written work even though I wrote it a week before the deadline.

Table of contents

1. Introduction	6
1.1. The expected benefit	7
1.2. Personal motivations	7
1.3. How was this research project carried out?	8
1.4. Content of this work	8
2. Voting systems	9
2.1. Voting systems in the sense of executing a vote	9
2.1.1. Paper-based voting	9
2.1.2. Electronic voting	9
2.2. Voting systems in the sense of determining the winning option(s)	10
2.2.1. Majority	10
2.2.2. Relative majority	10
2.2.3. Absolute majority	10
2.2.4. Qualified majority	11
2.2.5. D'Hondt method	11
3. Building an electronic voting system	12
3.1. Authentication	13
3.1.1. Situations 1 and 2 (log in with a DNle in the website)	16
3.1.2. Situation 3 (automatically generate a code with a DNle in a kiosk)	18
3.1.3. Situation 4 (manually generate a code)	25
3.2. Administrator dashboard	25
3.3. Citizen dashboard and votes	28
4. What does the world think about electronic voting?	32
5. Conclusion	33
Annex A. Emails interchanged with the Spanish police	34

Annex B. APDU commands	42
Annex C. RSA	44
Annex D. Secure Messaging	45
Annex E: Survey	47
Annex F. Source code	48
Bibliography	49

1. Introduction

We use a lot of paper. Whether at school, at work, or to write a shopping list, we definitely use too much paper. And it is because of this that one day I started to think about the enormous amount of paper being used for ballots in elections around the world.

As an example, 900 million ballots were printed for the Spanish 2011 general elections.^[1] Those were roughly 29,7 x 9,3 cm in size and assuming they had a density of approximately 60 g/m², we can deduce that all of them weighed a little bit less than 1500 metric tons. And, in the case that they were made of virgin fiber and no recycled paper at all, that would mean that more than twenty thousand trees were cut.^{[2][3][4]} Could an alternative solution be found?

I believe one of the alternatives which we should take into account is voting via a computer in the electoral colleges or from the comfort of our home. This way, we would help stop the deforestation that the printed ballots were contributing to, which would imply a social benefit on a global level. But obviously, in any computer system security is key, so we need several conditions to be accomplished by the voting system: votes must remain anonymous and every eligible voter must only be able to vote once. In order to ensure this, a system should be developed to verify the votes and do the count in a way that those votes can not be modified. This has already been taken into account in the current voting system by the Spanish government, autonomous communities and city halls, but nowadays there is not any electronic voting system via the Internet which meets the standards described above due to security issues.

Thus, taking into account this issue, I have asked myself the following question: **is it possible to create a web app which allows Internet voting in a comfortable and secure manner? If so, how?**

1.1. The expected benefit

First of all, I would like to explain the results I expect from this research project. One of the goals is –after having researched how to accomplish the various points described in the last section and having learned about the different voting systems, secure connections and data encryption– to conclude if the web app could be built or if we do not still have the technology to be able to develop it. This could benefit all the community, because in this project I will conclude if the use of this system would be effective today or if other alternatives should be researched.

My initial hypotheses is that although it would be quite difficult, it would be possible to create a web app that is more secure and comfortable than the current voting system.

In the case I verify my initial hypotheses, I have the additional goal of creating a web app that could substitute the current "physical ballots" voting system. This would have an enormous benefit in society, because it could be the beginning of the migration to Internet voting and therefore we would not use as much paper as before, which is the problem I detected in the beginning. Furthermore, this would imply other benefits: for instance, the count would be done much more precisely and in much less time.

1.2. Personal motivations

After reading several blog posts^{[5][6]} about electronic voting written by Ricardo Galli, the founder of the website *Menéame*, I realized that electronic voting is the future, because it solves one big problem which is the amount of trees which need to be cut down to print the ballots.

Since I was young, the fields of computer sciences and mathematics have always fascinated me, and when I saw in that article that the electronic voting systems which exist nowadays are very insecure and they do not live up to the classical (and physical) voting methods, I decided to take advantage of my passion for computer sciences along with my passion for mathematics to help solve this problem.

1.3. How was this research project carried out?

In order to complete this research project and extract some conclusions, there are several things that have been done. First of all, theoretical research about the current voting systems and methods to encrypt and decrypt data was done in order to answer the question “how do we actually vote?”.

After this, I started thinking about how an electronic voting system would be developed, so I started to research ways to authenticate the citizen who wants to vote and the security related to this process of authentication apart from security related to the voting process.

When I had finished all this, I started to design and program the voting system taking into account the previous research. This voting system consists on a website in which citizens can vote, and an app for mobile phones and tablets, and a web interface so an administrator can configure the votings. Also, while developing it some issues arose and so naturally I continued researching while developing the software.

When the development of the system was almost complete, I conducted one voting simulation with the ICT class students at St. Paul's School to gather some feedback and test if the system was working, and I also sent a survey to my acquaintances who had not participated in the voting simulation in order to know better how the world felt about electronic voting.

At the end, when the development of the program was complete, I started writing this dossier in order to record all the information about this research project.

1.4. Content of this work

This work consists of various chapters, which are structured according to the various phases of the research project. In Chapter 2, I discuss the current situation of voting systems. In Chapter 3, I talk about the research in terms of the building of an electronic voting system. In Chapter 4, I show the results of a survey I have conducted. And finally, in Chapter 5 I exhibit my conclusions.

2. Voting systems

The phrase “voting systems” can be interpreted in different ways. It could be thought of as the way to execute a vote or as the way to count votes and generate a final result which determine the winning option(s),^[7] and it is essential that we know about both of these to better know to develop the software.

2.1. Voting systems in the sense of executing a vote

Nowadays we have a dualism of voting systems: on one hand we have paper-based voting, which is the one we are all more used to, and on the other hand we have electronic voting, which is the one which consists in being able to vote from our own home over the Internet or at a polling place. Despite this, paper-based voting outweighs electronic voting around the world, in part because electronic voting is particularly new and in part because of general fear as it is believed to be less secure.^{[8][9]}

2.1.1. Paper-based voting

The process of voting in a paper-based vote consists in people submitting paper ballots into a box at a polling place. Later on, these ballots are counted manually by some people.

Different paper-based votes are executed in different ways, but in the Spanish general election, political parties send ballots to the citizens' residences some weeks before the voting so they can bring their ballot inside the envelope with them. Apart from this, ballots and envelopes are also available at the polling place.

2.1.2. Electronic voting

In electronic votings, as the name states, votes are cast from an electronic device such as a computer, a smartphone or a tablet. This term is very broad as some electronic votes only permit voting using controlled devices at a polling place while other votes are carried

out in their entirety over the Internet, and other votes allow people to vote both over the Internet and at a polling place.

2.2. Voting systems in the sense of determining the winning option(s)

There are a lot of voting systems which determine the winning option(s), and most of them have variants, but only the most important ones are detailed in this section:

2.2.1. Majority

In this system, the winning option is the one which obtains more than 50% of the total amount of votes cast. Therefore, if none of the options obtain more than 50% of votes, there is not a winner.

For example, the results in figure 1 indicate that option B wins, while the results in figure 2 indicate that none of the options wins.

Option A	Option B	Option C
7 votes	24 votes	4 votes
20%	68.6%	11.4%

Figure 1. Voting results sample #1

Option A	Option B	Option C
10 votes	9 votes	16 votes
28.6%	25.7%	45.7%

Figure 2. Voting results sample #2

2.2.2. Relative majority

In this system, also named “plurality vote”, the option with the higher number of votes is the winning one, without any restrictions such as a threshold.

Referring to the previous examples, applying relative majority to the results in figure 1, the winning option is still option B, while applying it to figure 2 the winning option would be option C.

2.2.3. Absolute majority

In this system, the winning option is the one with a majority of all voters. That is, the option which obtains more than 50% of votes out of all the people that can vote. If 100

people had the right to vote in the sample of figure 1, then option B would not win because it did not surpass the 50 votes threshold (one half of the people able to vote).

2.2.4. Qualified majority

The term qualified majority is used in votes in which the winning choice is determined by majority or absolute majority, but with a threshold greater than 50%. For example, the threshold could be two thirds of the votes cast, or three quarters of all the people who can vote.

2.2.5. D'Hondt method

The d'Hondt method, named after Belgian mathematician Victor d'Hondt, is slightly more complicated and different from the previous ones. Instead of determining the winning option, it is focused on the proportional distribution of seats in a house.^[10]

In order to use this method, we will have to gather some data beforehand. We need to know how many seats there will be in the house, and count the number of votes for each party/option.

After finding this out, we can proceed to distribute the seats. To do that, we have to follow these steps:^{[11][12]}

1. Take the party with the most votes (also referred to as “quote”).
2. Add one to the number of seats allocated for that party.
3. Replace the number of votes of that party with the result of the following formula, where V is the original number of votes that the party received and S is the number of the seats that the party has been allocated.

$$quot = \frac{V}{s + 1}$$

Figure 3. Equation for determining the quota of a political party after they receive a seat^{[10][11]}

4. Repeat the steps above until all seats have been allocated.

3. Building an electronic voting system

The electronic voting system that I have developed is focused in providing a way to hold the Spanish general elections via the Internet, although it is true that I developed it with other types of votes in mind so the system is flexible. Despite this, as I mainly focused on the Spanish general elections, all the details discussed in this section are related to this specific vote.

To actually start developing the system, I drew a very simple diagram of how the voting process would work like for citizens so I could have a starting point from which I could begin to build the system.

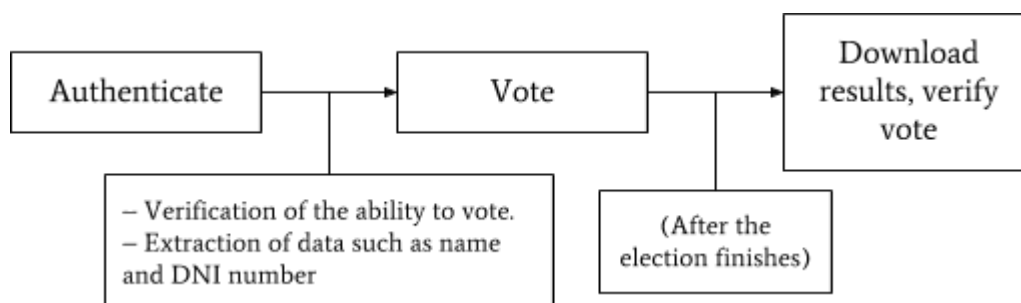


Figure 4. Diagram showing the first iteration of the voting system

This diagram shows in general terms what almost all votes (paper-based and electronic) work like, with the exception of the last step (“download results, verify vote”). This last step, which consists in that citizens should be able to verify if their vote was counted, is one of the goals explained in the introduction because it brings transparency to the vote and gives the final results more credibility.

The first part I started researching into was the one related to authentication. In this phase, the computer system should be able to know who is voting in order to prevent duplication of votes and check that the citizen is allowed to vote.

The second part I started working on was the admin dashboard, where the administrators would be able to configure everything.

And the last part I researched and worked on was the part related with the actual process of voting, as it is the most important one.

3.1. Authentication

In order to authenticate the voters, I decided to investigate which methods I could use. These are the methods which I thought were feasible for the system:

- **Username and password combination:** this method consists in giving every voter some unique credentials in order to log in. The main disadvantage of this method is that in order to distribute these credentials, the only secure way would be to give them in paper, because an attacker could intercept them much more easily if sent via another method, for example via email.
- **Electronic DNI (e-ID):** this method consists in introducing an electronic DNI (the Spanish Identification Card) into a DNI reader and enter its PIN code in order to use an authentication certificate included inside the DNI to log in.
- **Password/code:** similar to the first method, the citizen would have to enter a password/code which uniquely identifies the voter (without the need to enter a username) in order to log in to the platform.
- **Biometric information:** in this method, citizens would authenticate themselves using a part of their body in conjunction with a special device. For example, a fingerprint reader or eye scanner could be used.

After some time, I concluded that the “electronic DNI” method was the best of them in terms of security and comfort. That is because almost everybody has a DNIE (electronic DNI) and the DNIE includes an authentication certificate which as of today is cryptographically very secure, so it is a better solution than using a password or a username/password combination. Also, although the “biometric information” method is much more secure because it does not depend on an ID card and a PIN but instead on your own body, I do not find it factible as people do not have biometric devices at home and they are quite expensive.

So after determining that the DNIE was the ideal solution, I now faced a problem: as I have said, not everybody has a DNIE. What do we do with the citizens that are in the census and should be able to vote but have an old DNI without an integrated electronic chip? Moreover, what about the citizens who can not vote from their own homes because they do not have a DNI reader?

In order to solve this problem, apart from integrating an authentication system via DNIE, I thought there should be a secondary system. I decided that it was going to be authentication via a code, which will be explained later on.

Also, there is another challenge, and it is that not everybody knows how to use a computer and sometimes technology stops working, so the election day polling places should be opened in order to provide a place where citizens can vote from the website just as they would from home.



Figure 5. DNIE with the electronic chip^[13]



Figure 6. Old DNI without an electronic chip^[14]

Taking into account all the previous points, I designed the following authentication system:

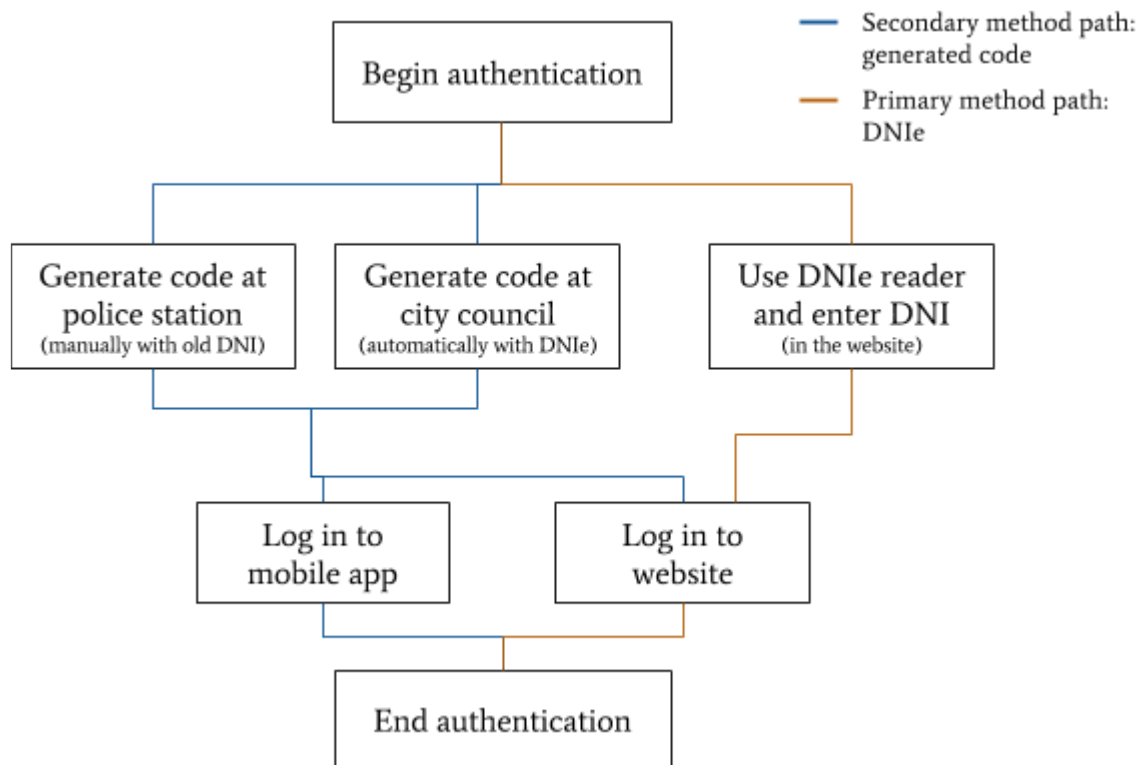


Figure 7. Diagram showing the final authentication system

The authentication system is based on two authentication methods as mentioned earlier on: the main one concerning the DNIe, and the secondary one concerning generated codes. This means that there may be multiple ways for a voter to successfully authenticate before voting, depending on the situation. These are all the situations covered by the authentication system:

- **Situation 1:** the voter visits the website from their own home, the website requests them to insert their DNIe into the reader, afterwards they enter the PIN code and after successfully completing this step they are authenticated.
- In the case the voter does not have a DNIe reader or a computer but does have a DNIe, they can do one of the following:
 - **Situation 2:** the voter goes to a polling place on the election day, where there are computers with DNIe readers and they can vote via the website as they would from their own home.

- **Situation 3:** the voter goes some days before the election day to a kiosk¹ in the city hall where there is a DNIe reader. In there they can automatically generate a code linked to the information of that DNIe following the steps shown in the screen. After completing all the steps, this code can be scanned with a mobile phone in order to save it and use it to vote on the election day.
- **Situation 4:** In the case the voter has an old DNI without an electronic chip, they have to follow a procedure similar to the one described above, with the exception that it is manual. The voter must go to a police station where a public worker will manually generate a code linked to the data that appears in the DNI, and the voter will be able to use this code to authenticate.

After having designed this system, I started programming it.

3.1.1. Situations 1 and 2 (log in with a DNIe in the website)

In this case, I will describe the process to program this part theoretically, as I could not program it due to the fact that the server that I used can not be configured in a specific way that is required for the code to work.

First of all the design consists in the following: a user visits the voting website. Afterwards, they click a button in order to tell the website that they want to vote with their DNIe. The website asks the user to enter the DNIe into the reader, and later asks the user their PIN code. In the end they have successfully authenticated.

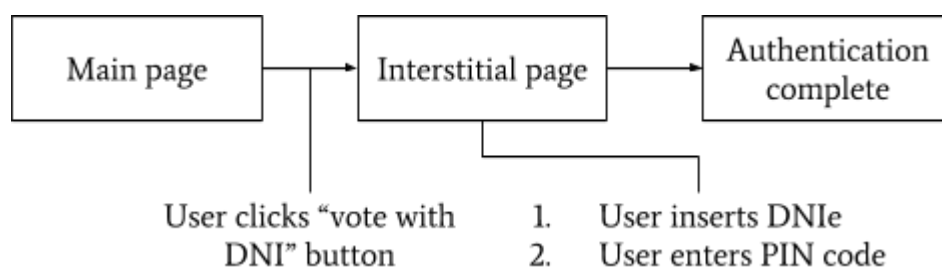


Figure 8. Diagram showing the procedure.

¹ A kiosk is a controlled public computer in which one particular kind of transaction can be done. In this case, the transaction would be generating a code in order to be able to authenticate and vote on the election day.

To develop this, we have to generate a pair of SSL keys (public and private) in order to establish the HTTPS protocol, which is required so the website can request the authentication certificate from the DNIe. We can request this pair of keys from a company such as LetsEncrypt or generate them by ourselves with a program such as OpenSSL (although the keys generated by OpenSSL will not be trusted by the browser, so generating them with LetsEncrypt is highly recommended).^[14]

After having generated these keys, we now can configure the server to request the DNIe authentication certificate when the user visits an interstitial page, which will be loaded when the user clicks the “vote with DNI” button. In order to do that, first of all we have to download the police CA certificate, and convert it to PEM format. Then, later on we will add the following code to the `/etc/apache/sites-available/default-ssl.conf` file:^[15]

```
# Code extracted from reference [15].

NameVirtualHost *:443
<VirtualHost *:443>

    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/apache.pem

    SSLCACertificateFile /etc/ssl/certs/acraiz-dnie.cer
    SSLVerifyClient require
    SSLVerifyDepth 2
    SSLOptions +StdEnvVars +ExportCertData

    DocumentRoot /var/www/

    ErrorLog /var/log/apache2/error.log

    LogLevel warn
    CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

Now we can program the interstitial page in order to retrieve the contents of the authentication certificate. To know whether the user has selected an authentication certificate, we check with PHP if the variable `$_SERVER['SSL_CLIENT_S_DN']` is set to true. If it is not, then the user has not selected any certificate and we should not proceed with the authentication process.

Afterwards, we check if the certificate is still valid via the `openssl ocsp` command, which sends the certificate details to the police OCSP server and returns whether the certificate is still valid or not (for example, if a DNI gets stolen). Afterwards, we use the `$cert = openssl_x509_parse($_SERVER['SSL_CLIENT_CERT'])` function to parse the certificate and we then access the certificate details via the `$cert["subjectDirectoryAttributes"]`, `$cert["subject"]["serialNumber"]` and `$cert["subject"]["CN"]` variables to get the birthday, identification number, and name of the DNIE. Now, the user would be successfully logged in.

3.1.2. Situation 3 (automatically generate a code with a DNIE in a kiosk)

I decided to program the code related to this situation as a form of a Chrome kiosk app² which reads the contents of the DNI and via AJAX/XHR connects to an API/web service hosted in the server which stores the generated codes and the data linked to them.

The design of this app and its interaction with the API would be the following:

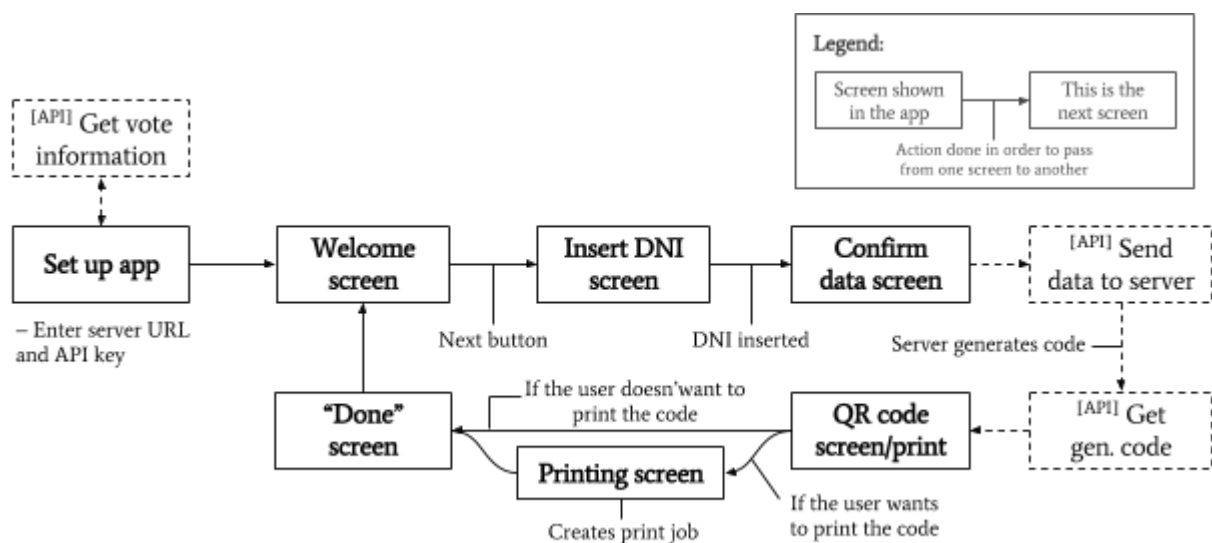


Figure 9. Diagram showing the lifecycle of the kiosk app.

The rationale behind developing a Chrome kiosk app^{[16][17]} instead of a regular program is that this app is able to communicate with the DNI reader natively and a Chrome app can be installed and run from any operating system that supports Chrome.

² A Chrome app is an app developed using HTML, CSS and Javascript which the Chrome browser and Chromebooks can run.

Also, if this design were to be implemented, Chromeboxes along with DNI readers and touch screens could be distributed between the city halls in order to run this app. As Chromeboxes and DNI readers are very cheap devices, a lot of money would be saved (one big difference is that Chromeboxes include Chrome OS, which is free software, while other computers include Windows, which is not free). Also, instead of Chromeboxes, Raspberries PI with Chromium could be distributed (they are slightly cheaper than Chromeboxes), but the deployment would be a little bit more difficult.

As for the development of this part, it has been one of the most difficult parts of this research project. Without proper information in the Internet about how the DNI chip works and how to communicate with it, programming this kiosk app has been like an odyssey.

Before describing how I programmed the kiosk app, I will explain why it has been an odyssey. First of all, information about how to communicate with the DNIE is very scarce, and the Spanish police, which is the organization that manages the DNIE and its distribution, is very secretive about how it works. Almost all the information found concerning how to communicate with the DNIE is from individuals who have been researching this in their own free time and have published how to do it online or from the police directly, via a document which was leaked in the Internet.

Moreover, I have tried contacting the police to ask some questions about this, but only after several months and a lot of perseverance did they answer. A copy of the emails interchanged with the police can be read in annex A.

In order to communicate with the DNIE, I have used a middleware app called "Smart Card Connector",^{[18][19]} which is able to transmit commands to the DNIE (which is considered a smart card), and to send the responses back to the kiosk app. A command is a series of bits (0s and 1s) which indicate the smart card what to do. They are called APDUs (Application Processing Data Units). For example, an APDU could indicate the smart card to browse and list the files of one folder or read those files, apart from being able to perform other operations.^[20]

But before going any further, we need to understand the internal folder/file structure of the DNIE. The DNIE has three areas where it stores files: the public area, the private area, and the security area:^[21]

- **Public area:** read-only area accessible without any restrictions. It contains:
 - Intermediate CA certificate
 - Diffie-Hellman keys
 - x509 component certificate
- **Private area:** accessible if the PIN is entered. It contains:
 - Non-repudiation certificate (for signing purposes)
 - Authentication certificate
- **Security area:** accessible if the PIN is entered from the DNI kiosks in police stations. It contains:
 - All the data printed in the DNI
 - Image of the photography
 - Image of the handwritten signature

We want to be able to access this authentication certificate in order to extract some data it contains: the citizen's name, their birthday and DNI number. In order to do that, we have to access the private area, and in order to do this we have to establish a secure channel and communicate via this secure channel because we are dealing with confidential information. Before continuing, though, the reader may want to take a look to annex B for a brief introduction to APDU commands.

In order to establish the secure channel and extract the information we need, the program should follow the steps below:^[23]

1. Retrieve the smart card's ATR (Answer To Reset), and check if it matches 3B:7F:00:00:00:00:6A:44:4E:49:65:00:00:00:00:00:03:90:00 (for running cards) or 3B:7F:00:00:00:00:6A:44:4E:49:65:00:00:00:00:00:00:0F:65:81 (for invalidated cards).

This is in order to prevent communicating with a smart card that is not a DNIE.^[25]

2. Send the APDU command “GET CHIP INFO” (90 b8 0000 07). This will return the serial number of the DNIE, which will be used later on to establish a secure channel.^{[29][30]}
3. We read the file 3F00/6020 (3F00 is the root folder and 6020 is the file), which is the intermediate component authority, and save the contents into the application memory. We send the commands “SELECT FILE” (00 a4 0000 02 6020) and “GET RESPONSE” (00 c0 0000 0e), and later on we read the file by sending various “READ BINARY” commands (00 b0 0000 ff ... 00 b0 03fc 30).
4. We repeat step 3 but with file 3F00/601F, which is the component certificate.
5. We extract the public key (modulus and public exponent) from the component certificate. (Read annex C for more information on RSA and public and private keys).
6. We use the “MANAGE SECURITY ENVIRONMENT” command in order to select the smart card’s Root CA public key (00 22 81b6 04 8302020f).
7. We send the intermediate authority certificate (C_CV_CA) which the smart card can verify:

```
002a00aed27f2181ce5f3781803cbadc3684bef32041ad15
5089258dfd20c69115d72f9c38aa99ad6c1aedfab2bfac90
92fc70ccc00caf482a4be31afdbd3cbc8c8382cf06bc0719
baabb56b6ec80760a4a93fa2d7c347f34427f9ff5c8de6d6
5dac95f2f19dac0053df11a507fb625eeb8da4c0299e4a21
12ab7047588b8d6da7592214f2dba140c7d122579b5f383d
2253c8b9cb5bc3543a55660bda80946afb0525e8e5586b4e
63e89241497836d8d3ab088cd44c214d6ac856e2a007f44f
83743337371add8e030001000142086573524449600006
```

8. We select the certificate we just loaded into the card with the command “MANAGE SECURITY ENVIRONMENT” (00 22 81b6 0a 83086573534449600006).
9. We send the terminal certificate (C_CV_IFD), which can be verified by the card, using the command “PERFORM SECURITY OPERATION”.

```
002a00aed17f2181cd5f378180825b69c6451e5f51707438
5f2f17d64dfe2e68567567094b57f3c578e830e425572de8
28faf4de1b01c394e345c2fb0629a393492f94f570b00b1d
677729f755d107022bb0a116e1d7d7659db5c4ac0ddeab07
ff045f37b5daf1732b54eab238a2ce17c9794187759cea9f
92a17805a27c1015ec56cc7e471a488e6f1b91f7aa5f383c
adfc12e856b202346af8226b1a882137dc3c5a57f0d2815c
1fcd4bb46fa9157fdffd79ec3a10a824ccc1eb3ce0b6b439
6ae236590016ba690001000142086573534449600006
```

10. Again, we select the public key we just loaded for the authentication with the command “MANAGE SECURITY ENVIRONMENT”, and in the same command we also select the component private key that will also be used in the authentication commands (00 22 c1a4 12 830c00000000200000000000000018402021f).
11. Now we proceed with mutual authentication, in order to establish the secure channel. We start with the card's authentication, so we send the “INTERNAL AUTHENTICATION” command (00 88 0000 10 [RND.IFD]20000000000000001). RND.IFD is a random 8 bytes challenge.
12. We request the response with the command “GET RESPONSE” (00 c0 0000 80).
13. The response will be in the following form:

```
data = E[PK.IFD.AUT] (SIGMIN)
SIGMIN = min ( SIG, N.ICC - SIG )
SIG= DS[SK.ICC.AUT] (
    0x6A || - padding according iso 9796-2
    PRND1 || - (74 bytes) random data to make buffer 128 bytes
length
    Kicc || - (32 bytes)- ifd random generated key
    sha1_hash(
        PRND1 ||
        Kicc ||
        RND.IFD || - (8 bytes) response to get_challenge() cmd
        SN.IFD - (8 bytes) serial number from get_serialnr() cmd
    ) ||
    0xBC - iso 9796-2 padding
) - total: 128 bytes
```

14. We decipher this message using RSA and after series of operations we obtain the `Kicc` value, which will be used later on to calculate the channel keys. (You can obtain more information about how `Kicc` is extracted by reading the source code of the file `kioskapps/dnie/app.js`).
15. Afterwards we form another message similar to the previous one so that the smart card can obtain `Kifd`, which is a random 32 bytes string that the terminal generates before sending the message with the command “EXTERNAL AUTHENTICATION” (from here we can extract the value of `RND.IFD`, used later on).
16. Now that we know the values of `Kicc` and `Kifd`, we proceed to generate the `Kenc`, `Kmac` and `SSC` values, which will be used to send secure APDUs (via the secure channel).
17. To do so, we first calculate `Kifdicc`. This value is the result of performing the binary operation `Kicc XOR Kifd`.
18. The value of `Kenc` is the first 16 bytes of the hash of `Kifdicc` concatenated with the value `00000001`. We save this value.
19. Afterwards, the value of `Kmac` is the first 16 bytes of the hash of `Kifdicc` concatenated with the value `00000002`. We also save this value.
20. Finally, to get the value of `SSC`, we concatenate the last 4 bytes of `RND.ICC` with the last 4 bytes of `RND.IFD`.
21. Now that we have the values of `Kenc`, `Kmac` and `SSC`, we can send secure APDUs using the steps described in annex D.
22. We read the file `3F00/5015/6001` with insecure APDUs (they do not need to be encrypted) in order to know where is stored the authentication certificate.
23. After requesting the user to enter the CHV/PIN, we send it to the smart card inside a secure APDU with the command “VERIFY” (`00 20 0000 [Lc] [Hex(CHV)]`).
 - a. If the response is `9000`, the CHV is correct and we may proceed.
24. We load the authentication certificate from the path we obtained in step 22 via secure APDUs.
25. We decompress the binary response with the `zlib` library.^[32]

26. We read the certificate and extract the citizen's name, birthday and DNI number.

After extracting this information, the program will ask the citizen for confirmation and the code will be generated, being linked with this data. In practice, the program only works until step 24, as the decompression fails, so currently the kiosk app is only able to retrieve the name and DNI number from a file in the public area, and verify the PIN code.

Also, in the website there is a place where citizens can introduce their codes in order to sign in.

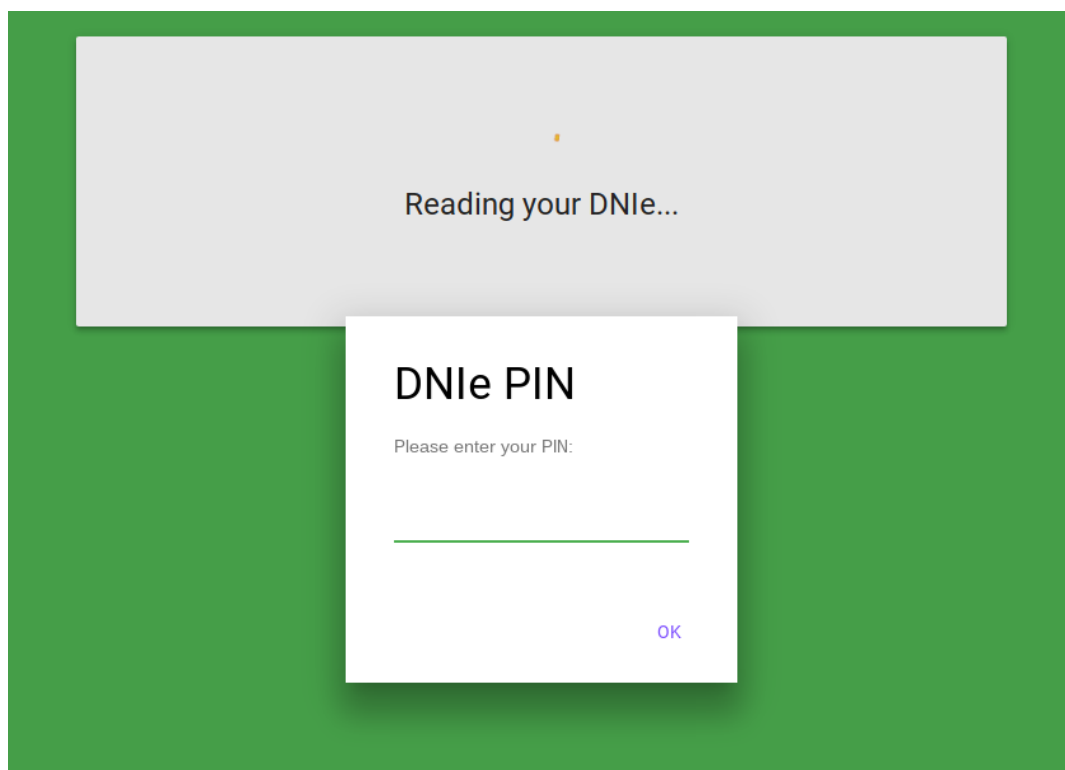


Figure 10. Screenshot showing the “enter PIN/CHV” screen

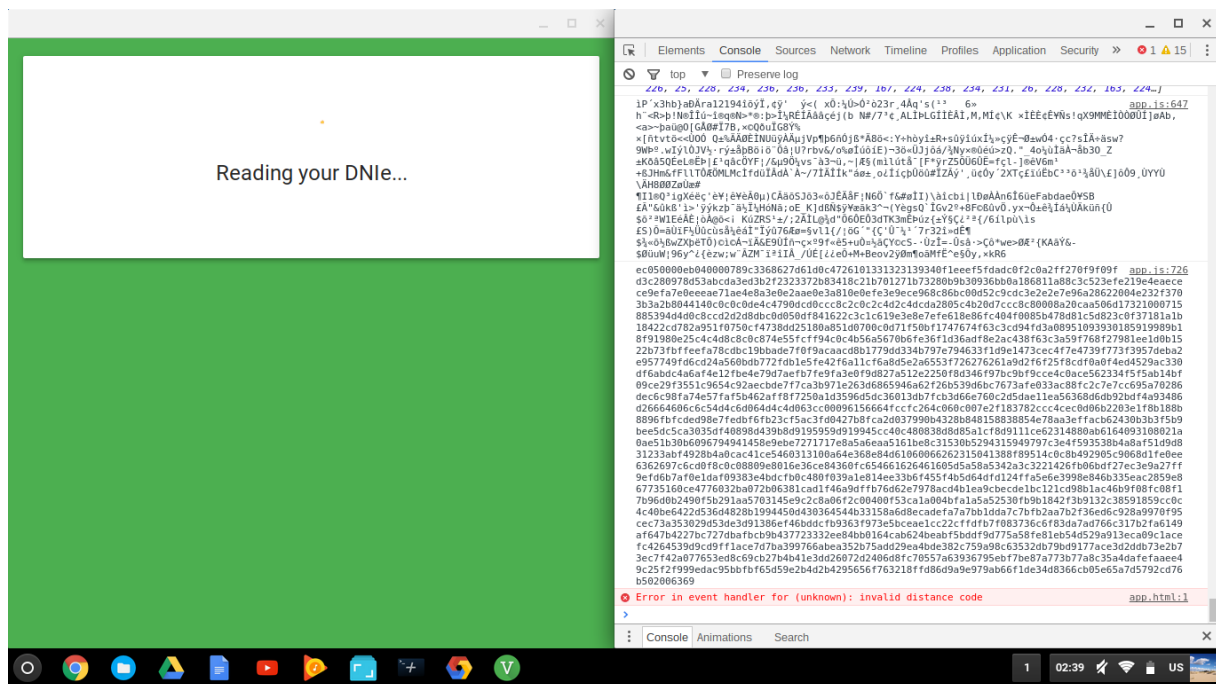


Figure 11. Screenshot showing the error that occurs when zlib tries to decompress the authentication certificate: “invalid distance code.”

3.1.3. Situation 4 (manually generate a code)

In this situation, the design is based in that an administrator can generate a code directly from the admin dashboard in the website. It is the same idea of situation 2, but with the difference that in this situation the citizen's data is entered manually in a form.

3.2. Administrator dashboard

In this dashboard, as mentioned before, administrators will be able to configure all the system, so its security is very important, as an intruder exploiting some vulnerability in the code or forging authentication details of administrators might be able to boycott a voting. Keeping that in mind, I decided to implement an username-password login page in order to authenticate users, with an option to enable 2-step verification.³

³ Two-step verification is a system which consists in that users have to successfully complete a challenge after correctly introducing a username and a password in order to log in. This challenge usually consists in that the user must enter a code generated in or sent to their mobile phone.

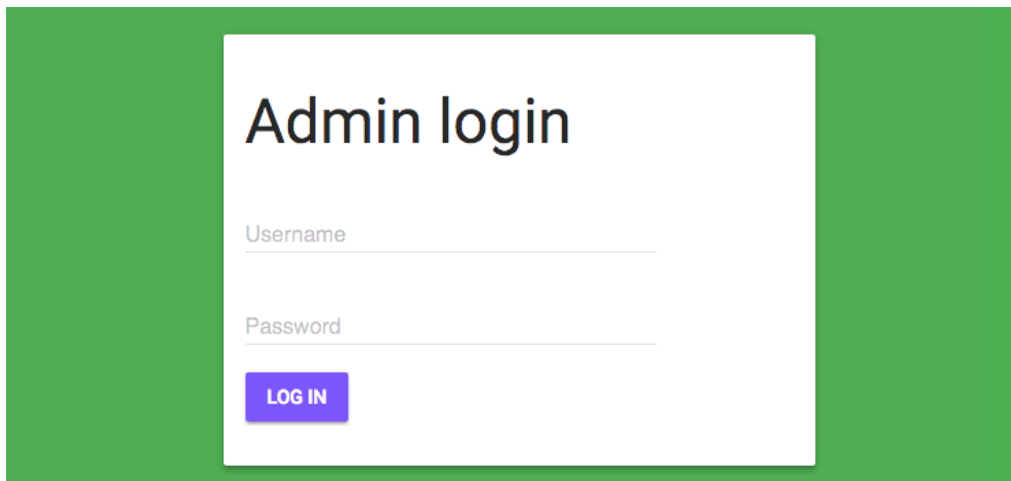


Figure 12. Administrator login page

In order to fulfill all the necessities that the administrator could have, I decided to divide it into 5 sections:

- **Users:** where administrators can add more administrators with different privileges.
- **Votes:** where administrators will be able to create, modify and publish votes.
- **Census:** where administrators will be able to manually add voters to the census, and also view and control all the generated codes and their usage.
- **Results:** where administrators can generate the results of a voting and generate the ballots file: a file which contains all the ballots submitted.

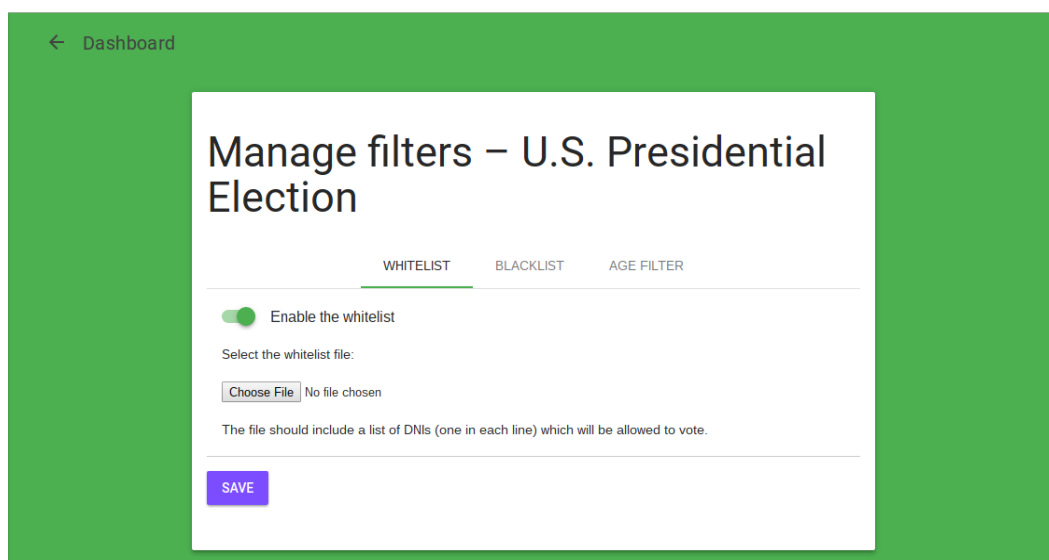


Figure 13. Screenshot of the filters page.

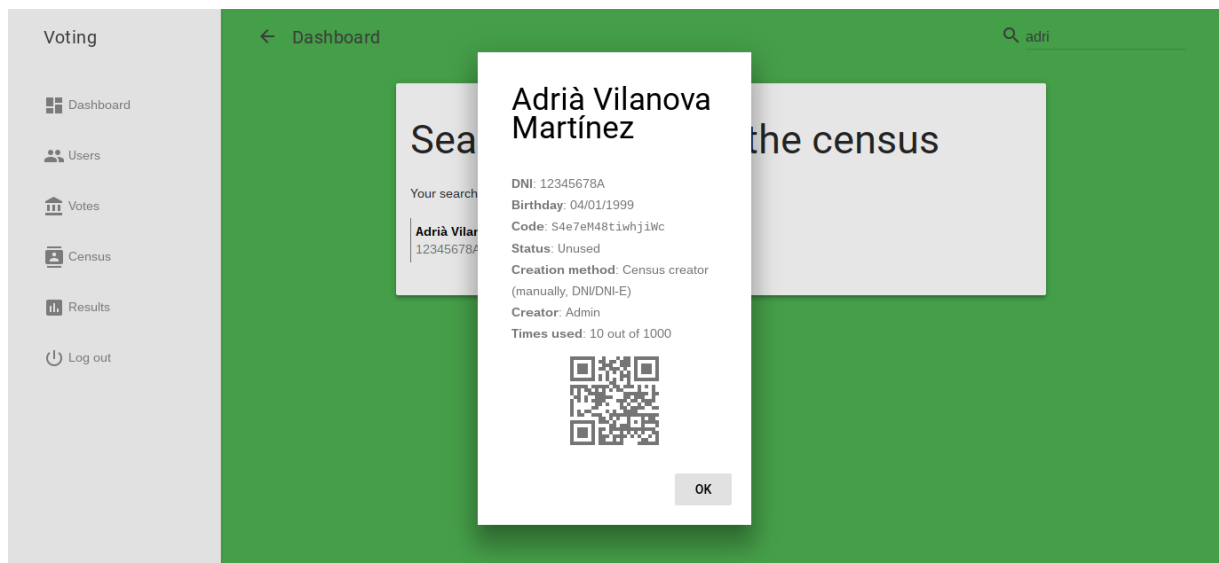


Figure 14. Screenshot of a generated code in the census

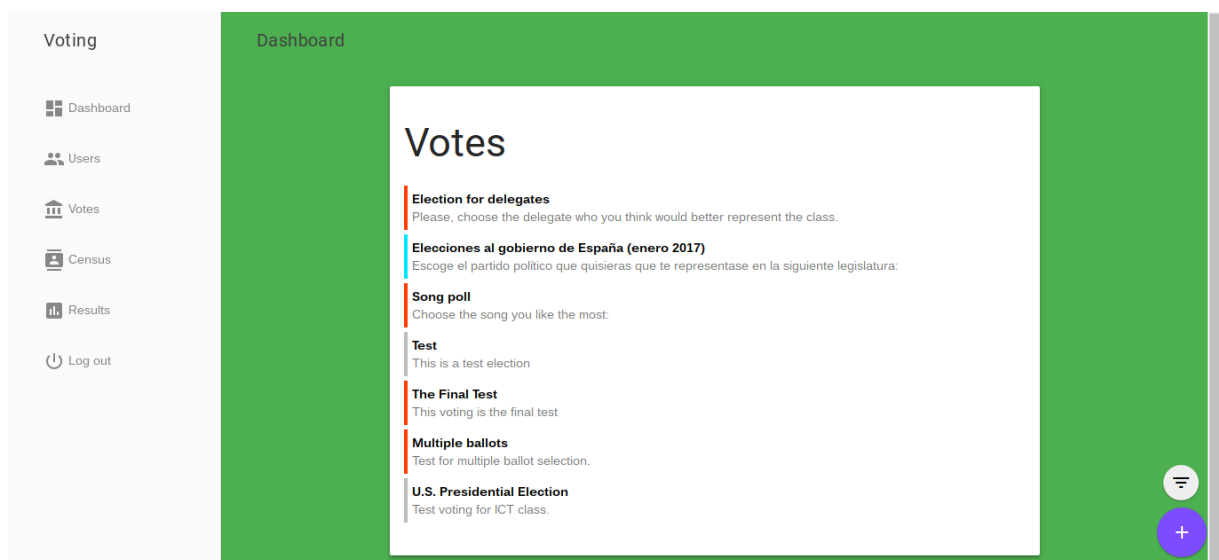


Figure 15. Screenshot of the list of votes. Gray represents draft votes, blue represents upcoming votes, green represents running votes and red represents finished votes.

3.3. Citizen dashboard and votes

The citizen dashboard is the most important part of the software: it is where voters vote. When developing this part, I had to keep in mind the standards I mentioned in the introduction: “votes must remain anonymous and every eligible voter must only be able to vote once”.

In order to ensure this, I designed the voting system trying to make it the most secure possible. When users vote, their votes are stored securely in a file/table which does not contain any information about the voter and along with a hash which uniquely identifies the vote but not the voter. After voting, the user is shown the salt of the hash which they can save in order to verify later on the authenticity of their vote.

This unique string is generated with the following function: `sha512(DNI || BALLOT NAME || SALT)`;

At the end of the voting, every citizen can download a list of all the votes with the identification strings. Since the identification strings are only known to the voters and they are unique, they serve two functions: they can be used to verify if the vote is in the list, and they can be used to verify that the vote was not in any way modified. Also, these identification strings only identify the vote, not the voter, so votes continue to be anonymous.

Finally, in order to guarantee the uniqueness of the votes, a list of people who have voted is saved in another file/table. This is the exact procedure in which vote uniqueness is ensured in the general elections in Spain, where the people in the voting table cross out the name of the voter when they vote.

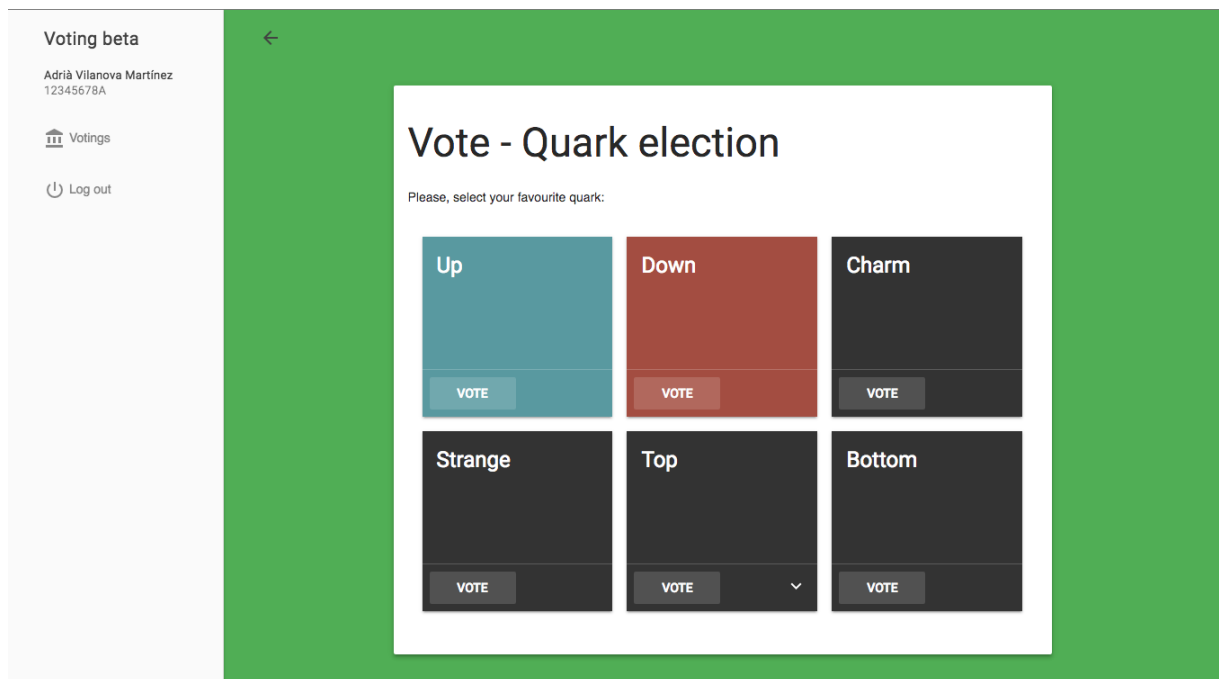


Figure 16. Screenshot of the vote. In this case, voters have to choose an option from the six that are shown.

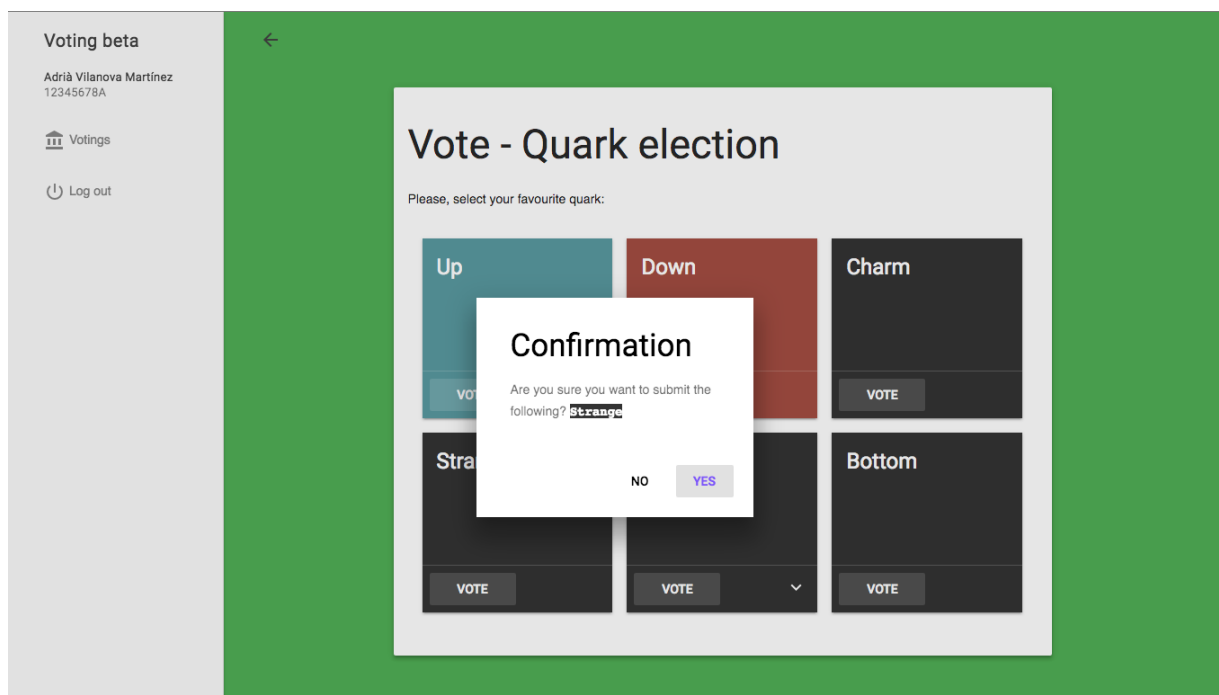


Figure 17. Confirmation screen when the voter selects an option to be sure that it is what they want to vote.

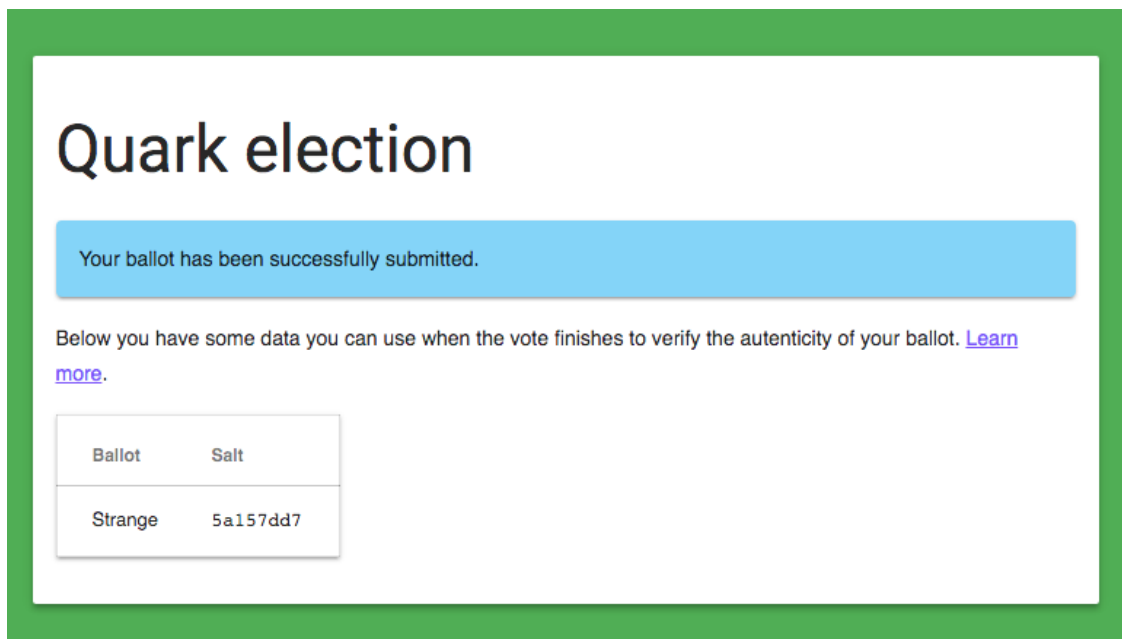


Figure 18. Screenshot showing the screen after voting. It shows the salt that can be used to verify the authenticity of the vote later on.

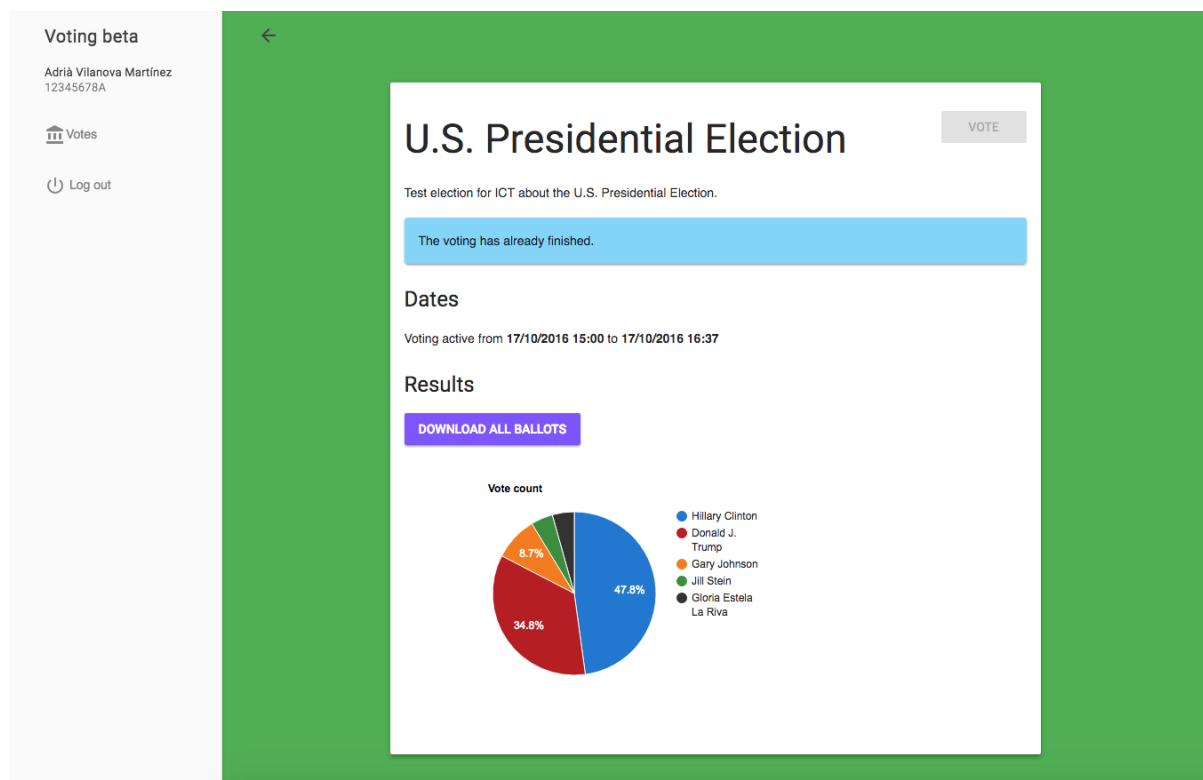


Figure 19. When the election ends, the vote home page shows the results and a link to download all the ballots. This is the page of the vote that was done in the ICT class to test drive the system.

```
{
  "available_ballots": {
    "7": "Hillary Clinton",
    "8": "Donald J. Trump",
    "9": "Gary Johnson",
    "10": "Jill Stein",
    "11": "Gloria Estela La Riva"
  },
  "submitted_ballots": [
    {
      "ballot": "10",
      "shahash": "58822c8dbaef1ecd923dc24f5e118bbd02e2de220b264393297dca952220ff69e5de71a9476ab0ef67d044019b34f214e8bc3e943cc06150cf7927174dc84bd7"
    },
    {
      "ballot": "8",
      "shahash": "9bd76642ecdb8be8e29478df96fce8e536fa4c2edddb1102fb7f3df1843fb897f2a17c2afd2abd25e7c634f92dd14cf1525087256e093dfcb91816031dc5b080"
    },
    {
      "ballot": "7",
      "shahash": "e4f4f79a31021083a2ad948658601f44239da9875e3d692f0fe2a9cdcd4fe9b8d5d7e1fa87920206dd5da0de2f2488be18d7bb111f5618882c578d69335168d7"
    },
    {
      "ballot": "7",
      "shahash": "5ef81da68c70e9404833f56a4cdbc22538d8db22d36e4a258f61e85a3ebaae8fa701eb23513e8246649d6cf9caac9c3ea968fbc7e3fec1089edebd7ad2b52a2a"
    },
    {
      "ballot": "8",
      "shahash": "3a2d2c98a006588d4c9c89bdc77908e3d40f4de3a092691a571e90aa0577c892001d2b749a1682beffa769eb77595897dd894d3676ee127c1ee939f3d106d8cf"
    },
    {
      "ballot": "7",
      "shahash": "2849c97199e43f8facb815c50fcf7387f57506eb39c36f1330068d1128c59d9868decfe2927e37295974b39ee88821d75de18997c6d641709d9de399d9c7cead"
    },
    {
      "ballot": "7",
      "shahash": "e6e17c59f97452fb4043076ae27cdb0b2624b0429e6098481044953364c63a342b927672740fffb30a8d9eb1666241bc6ae5617e4aa1b6dca8f7f37b8343c96c"
    },
    {
      "ballot": "7",
      "shahash": "5ee4c9640c7b3b897912ccfeb53c0b539ffb316c2a203e301bfcd8b1915baa32b426f49006b5beebcee9317cee11d729ec7fbfa6a4c27ba41e36b33053eb30ba"
    },
    {
      "ballot": "8",
      "shahash": "46c82b8d7ac8ca4c936cdc87d157bfd3d422b244a43bd128d9e3a20e1620ec500bb186b324f74b4baa2ba1b209a8772fc30058fb9ff3d0fa1b5ae63b467207b"
    },
    {
      "ballot": "7",
      "shahash": "a1f52022ad705d36aab9642015dd993f9b9bcd1084dc3668a9a10484e890261459ea05816554454feded5bcdd6752111b20b1bbcb84faad92f029f1f09ab63a3c"
    },
    {
      "ballot": "8",
      "shahash": "3367566fde815ed8c29baeba81d45775c2fbc343ab3347918360ab964ac07ac8e62519ee06a8cdb96ealecb07cf5bd4dba1f69a6a7e7d6dae7036c8cb951634"
    },
    {
      "ballot": "9",
      "shahash": "ff6bb7c4abe4a7f87181f329da9c7f88d33afd5de1b9dfc2f9f7eab9ce1988e9775f6b73511ed3f4c9a239b1420233a9471d521772d2a7f62c1c7a1491bac761"
    },
    {
      "ballot": "11",
      "shahash": "9b191c0821075b62a50e55f8d2039919663a5bd0954377ff5534a658ce02ffca4eca366e6359628cd7e84234397c67cdd3676a43e545a3388fa7600819ad2384"
    },
    {
      "ballot": "8",
      "shahash": "18e79d6aa9ef6a738d376672faa20cca3e31206faeee63c8470b40a8884453e9eca8e7b5eed29d01919e4afddaab2796d7b2157d5e9637ab7f8cfe00dbad2473"
    },
    {
      "ballot": "8",
      "shahash": "fd8d54a67419bb77c5a0c3d105e306c220bdf7c9bc3c741d0d969ff90110d4eaecaedf60e7522e0526669d696ffaa97749b41435386f91f8fedf1d5dfbcb2673"
    },
    {
      "ballot": "7",
      "shahash": "42964e492fd54269202024a36fc73b0095ea3bd32c597f689bd22dbacde2269d41a9640ba3a5fb223e97b6d9de070f4145861df1ca87d9d3bad2b52db0307d59"
    },
    {
      "ballot": "8",
      "shahash": "0b26917aca216676068d031559e8bbc63adbe0535c1f32ae5a9151329a284048bc66477180e64e3d5a00cee2f38285cd05254662e1987a8f745f40143144986a"
    },
    {
      "ballot": "7",
      "shahash": "9fbb54e4a79b25fe3611804e10bbe5b1cf1d54b34c15b18b69465dad3da6cac0cbd9f1f8b4a3a85bab3786ff0d6654fa3e9390c7bb40971c4a9392fe30bac6b1"
    },
    {
      "ballot": "7",
      "shahash": "eee221925a3ca91f1b47e8f7faffd0e2c873167caf15546e3ba13fed1c985fb2c3d16f97985dd2f2ca402c63ab568de42579f02d6ab7600f9ef296714a7aaa52"
    },
    {
      "ballot": "7",
      "shahash": "c2c17966929b0a26ec55be05c3e60800ef634f307a53387faf88a4452a0f488ccc11c7f7df052807f2d5b2cbf1a5cf373a848244b536d4c56a68954dab929495"
    },
    {
      "ballot": "7",
      "shahash": "746de95411b4d252a7da7d0d9cfb7a57cd65792c59ef341a34392641d839f25029ebf6edeb62b723d620fb5f3dd7287eb03e4ea5b572f4857a91948276ed032b"
    },
    {
      "ballot": "8",
      "shahash": "56c02709c37406897198e5c8637fd43ff899968429a89fbbdb25f44fb243278d27631f67e79429b0d52863b708272e10d39617f799912bf8d49b21bf6f41252f"
    },
    {
      "ballot": "9",
      "shahash": "0f7bea866d4d215408f0ac1765846819bc365bc481a0d059e29edbe6335ce2379fcce404e1ae761adf7c656aaef95d99aaaaa1f637d4c577721348a84b6d22d3"
    }
  ]
}
```

Figure 20. Sample ballot file in JSON format. It corresponds to the results shown in figure 19.

4. What does the world think about electronic voting?

That was the question I was trying to answer with a survey I conducted. The survey focused on the differences between the actual voting system, a hypothetical electronic voting system, and my proposed solution.^[32]

The results, which are shown in Annex E, are based in the responses from 146 people of a wide variety of ages and equality with regard to sexes. Meanwhile, the predominant type of computer user is the medium user.

Something somewhat interesting is the difference of the average of the punctuation of security between the actual voting system and a hypothetical electronic voting system. In the first one, the level of security is around 6.57 while in the second one it is around 6.31, only with a difference of -0.26. Comparing the levels of comfort, the second one doubles more or less the levels of comfort of the first one. When comparing these two with my proposed solution, the security increases +0.47 points when compared to the actual voting system and +0.73 points when compared to a hypothetical voting system, which I would say is due to the fact that people were convinced by the effectivity of the SHA512 hash. Also, when comparing them with comfort, it still is more comfortable than the actual voting system (+2.85), but it is not as comfortable as a hypothetical voting system (-0.95). Therefore, I think sometimes security reduces comfort and comfort reduces security.

Moreover, the distribution of answers in the question “Would you trust the outcomes of a vote held via an electronic voting system?” is very homogeneous, showing that there are various points of view in this area.

I can conclude from this survey that people believe that my proposed system is more secure than others and that it is much more comfortable than the actual voting system, so this system is viable.

5. Conclusion

To conclude my research project and answer the question asked at the beginning (“is it possible to create a web app which allows Internet voting in a comfortable and secure manner? If so, how?”), we should look back at all the work done and how it turned out.

First of all, in matters of authentication, I think in Spain we do not yet have the technology that would allow a secure and comfortable process of authentication. This is because the best solution, which would be the DNIe, is not sufficiently documented and therefore not viable. Other solutions, as discussed earlier on, are worse, so we should wait until further improvements are made to the DNIe and official documentation starts becoming available.

Apart from the problem with authentication, although I think a very secure system can be developed, there is also another problem: viruses. If some virus accesses your computer it may change your vote without you knowing, and when you try to verify your vote it could change the outcome so it tells you that it was counted even if it was not.

Therefore, after doing all this research project, I have concluded that it is feasible to hold electronic votes, but at small scale (because in this case it is easier to see if the results were in any way manipulated) or in a controlled environment (because viruses would not affect the outcome of the vote). That means that for now, the Spanish general elections should not allow electronic voting except if it is done in a controlled manner at polling places, and with the risk that a virus could affect all the system. Nevertheless, if Chromebooks/Chromeboxes were chosen as the computer to show the website, that would not be a problem as their operating system (Chrome OS) has never been compromised by a virus, so there is no risk of infection.

Annex A. Emails interchanged with the Spanish police

In order to complete my research project, I interchanged some emails with the Spanish police in order to ask them some questions. Below you can find a copy (in Spanish) of those. Note that the oldest messages are in the bottom, the newest in the top.

Adrià Vilanova Martínez <jocdeladria@gmail.com> Thu, Oct 27, 2016 at 12:38 AM

To: "soporte.sacdni" <soporte.sacdni@policia.es>

Muchísimas gracias por la respuesta, ahora el hash funciona correctamente.

Adrià Vilanova Martínez (@avm99963)

2016-10-18 15:14 GMT+02:00 soporte.sacdni <soporte.sacdni@policia.es>:
Buenas tardes:

Las operaciones de "hashing" no se llevan acabo sobre una codificación determinada de la cadena (hexadecimal, B64, ASCII,...) sino sobre el valor binario de la misma. Sea el ejemplo que Vd. cita en el que los datos sobre los que se va a calcular el hash son:

PRND1=c6d2534290ca9938a5c69149d2b8ae2eb565a76b973455cb203484d893218df38454720dcdcc777460938bfcc888e113b3d7aa1f703e8ce1091ef1824c87df4e7c1be5da3a9c0521ba75

KIcc=886b2b015ce1c9529d9463695fabd1a4a4b85e0alccd4f55292c20e21e954d31

RND.IFD=231dac737ba0fe6e

SN.IFD=200000000000000001

La función SHA1 se invocará con la contatenación de tales valores

h[PRND1||KIcc||RND.IFD||SN.IFD]=c6d2534290ca9938a5c69149d2b8ae2eb565a76b973455cb203484d893218df38454720dcdcc777460938bfcc888e113b3d7aa1f703e8ce1091ef1824c87df4e7c1be5da3a9c0521ba75886b2b015ce1c9529d9463695fabd1a4a4b85e0alccd4f55292c20e21e954d31231dac737ba0fe6e20000000000000001

Almacenamos la representación hexadecimal en un fichero auxiliar llamado

fichero_1 y transformamos la representación hexadecimal a binario. Para ello utilizaremos el comando xxd y la salida quedará en fichero_2

```
$ xxd -r -p fichero_1 > fichero_2
```

Utilizaremos OpenSSL para efectuar el resumen criptográfico de fichero_2

```
$ openssl dgst -sha1 fichero_2
```

```
SHA1(fichero_2)= 624cca198a2463ec73f17bbc12461f182176a8cc
```

Un saludo

FirmaSAC.png

Soporte Servicio de Atención al Ciudadano

soporte.sacdni@policia.es

De: Adrià Vilanova Martínez [mailto:jocdeladria@gmail.com]

Enviado el: miércoles, 12 de octubre de 2016 13:31

Para: Soporte DNI

Asunto: Re: RV: RV: RV: ZonaTic y establecimiento del canal seguro (LRA)

Buenos días.

Llevo esperando respuesta desde hace un poco más de un mes y todavía no he obtenido respuesta alguna. Además mi trabajo de investigación concluye en una semana.

Por ello, ¿tienen alguna novedad sobre mi cuestión?

Reciban un cordial saludo.

Adrià Vilanova Martínez (@avm99963)

2016-09-08 15:44 GMT+02:00 Adrià Vilanova Martínez
<jocdeladria@gmail.com>:

Muchísimas gracias.

Adrià Vilanova Martínez (@avm99963)

2016-09-08 15:42 GMT+02:00 soporte.sacdni
<soporte.sacdni@policia.es>:

[VJM]

Buenos días

Le enviamos su consulta al departamento correspondiente. Tan pronto tengamos una respuesta se la haremos llegar por este medio.
Reciba un cordial saludo.

FirmaSAC.png

Soporte Servicio de Atención al Ciudadano

soporte.sacdni@policia.es

De: Adrià Vilanova Martínez [mailto:jocdeladria@gmail.com]
Enviado el: jueves, 08 de septiembre de 2016 15:32
Para: Soporte DNI
Asunto: Re: RV: RV: ZonaTic y establecimiento del canal seguro (LRA)

Hola de nuevo.

Exacto. La verdad es que no fui muy preciso al usar mis palabras. Me refería a leer el certificado X509 de autenticación, tal como se muestra en la página 65 de la guía. Aun así, no puedo llegar hasta allí sin antes establecer el canal seguro.

Un saludo.

Adrià Vilanova Martínez (@avm99963)

2016-09-08 15:28 GMT+02:00 soporte.sacdni
<soporte.sacdni@policia.es>:

[VJM]

Buenos días:

El certificado de autenticación no se puede extraer del dni. Éste lleva un sistema de seguridad en el que todos los procesos se ejecutan dentro del chip, por lo que no es posible extraer los certificados del mismo. Por eso es necesario que el dni esté dentro de un lector cada vez que necesite autenticarse o firmar, todos los procesos se realizan dentro del chip.

Reciba un cordial saludo.

FirmaSAC.png

Soporte Servicio de Atención al Ciudadano

soporte.sacdni@policia.es

De: Adrià Vilanova Martínez [mailto:jocdeladria@gmail.com]
Enviado el: jueves, 08 de septiembre de 2016 15:04
Para: Soporte DNI
Asunto: Re: RV: ZonaTic y establecimiento del canal seguro (LRA)

Buenos días.

La consulta es perteneciente al establecimiento del canal seguro con el DNI electrónico. Estoy desarrollando un software de votación electrónica como trabajo de investigación de bachillerato y para poder extraer del DNI electrónico el certificado de autenticación es necesario que antes establezca un canal seguro y pase el PIN mediante el canal seguro.

La cuestión es que en el proceso de establecer este canal seguro (que explicáis en la guía de referencia técnica del DNI electrónico) me quedo estancado al verificar el hash SHA-1 obtenido del comando INTERNAL AUTHENTICATION. Este, según está explicado en el documento, es el resultado de hacer `h[PRND1 || KICC || RND.IFD || SN.IFD]`. Entiendo que hay que concatenar PRND1, KICC, RND.IFD y SN.IFD y después calcular el hash SHA-1 de esta concatenación, pero no consigo calcularlo correctamente. ¿Me podrían decir si estoy erróneo en algo de lo que he dicho o qué podría estar ocurriendo para que no coincidan los hashes?

Siguiendo el ejemplo que está en el final de la guía (página 49), concateno las 4 variables obteniendo `c6d2534290ca9938a5c69149d2b8ae2eb565a76b973455cb203484d893218df38454720dcdcc777460938bfcc888e113b3d7aa1f703e8ce1091ef1824c87df4e7c1be5da3a9c0521ba75886b2b015ce1c9529d9463695fabd1a4a4b85e0a1ccd4f55292c20e21e954d31231dac737ba0fe6e2000000000000001`, pero ahora que tengo concatenados todos los bytes no sé si hay que hacer el hash de la representación hexadecimal de estos, de la representación ascii de los bytes o qué.

Espero que me entiendan y puedan solventar mi duda. Les muestro mi agradecimiento por adelantado, ya que gracias a su ayuda podré completar satisfactoriamente el trabajo de investigación que estoy realizando en un área tan poco explorada como es el de las votaciones electrónicas y gracias a un componente tan innovador como es el DNI electrónico.

Un saludo,

Adrià Vilanova Martínez (@avm99963)

2016-09-07 14:35 GMT+02:00 Soporte Atención Ciudadano (DNIE) - C.N.P. <Soporte.sacdni@policia.es>:

Buenos dias.

Por este medio puede hacernos las consultas que desee sobre todo lo concerniente al dni electrónico y la instalación de los software necesarios para su funcionamiento.

Un saludo.

De: Adrià Vilanova Martínez [mailto:jocdeladria@gmail.com]
Enviado el: miércoles, 07 de septiembre de 2016 14:03
Para: Soporte DNI
Asunto: Re: ZonaTic y establecimiento del canal seguro (LRA)

Hola de nuevo.

Gracias igualmente por responderme e intentar ayudarme.

Hablando con un inspector de Hacienda, me dijo que el tema del DNI electrónico lo llevan desde El Escorial. ¿Sabe si me podría poner en contacto con ellos para poder realizarles a ellos la pregunta?

Gracias.

On Mon, 29 Aug 2016 at 08:54 Soporte Atención Ciudadano (DNIE) - C.N.P. <Soporte.sacdni@policia.es> wrote:

[LRA]

Buenos días:

Lo único que nosotros le podemos ofrecer es lo que aparece en la página del DNIE dentro del Area de descargas http://www.dnielectronico.es/PortalDNIE/PRF1_Cons02.action?pag=REF_1120

O bien en la Oficina técnica http://www.dnielectronico.es/PortalDNIE/PRF1_Cons02.action?pag=REF_082&id_menu=0

Si en estos enlaces no encuentra usted lo que busca, sintiendolo mucho no podemos ofrecerle otra solución.

Reciba un cordial saludo.

Soporte Servicio de Atención al Ciudadano

soporte.sacdni@policia.es

De: Adrià Vilanova Martínez [mailto:jocdeladria@gmail.com]

Enviado el: domingo, 28 de agosto de 2016 13:30

Para: Soporte DNI

Asunto: Re: ZonaTic y establecimiento del canal seguro

Gracias por su respuesta.

Entonces, ¿ustedes saben dónde se encuentra la información perteneciente al desarrollo de aplicaciones que interactúen con el DNI electrónico o con quién me podría comunicar que entendiera sobre esto? Todo lo que he encontrado en la página web de la policía se limita a ofrecer información general sobre el DNI electrónico.

De nuevo, gracias.

Adrià Vilanova Martínez (@avm99963)

2016-08-18 9:00 GMT+02:00 Soporte SAC DNIE
<soporte.sacdni@policia.es>:

soporte.sacdni@policia.es Buenos días:

Le agradecemos el interés demostrado en el DNIE. A continuación intentaremos dar respuesta a su mensaje. En caso de que la información que le facilitamos no sea la que precisa, no dude en ponerse de nuevo en contacto con nosotros en la dirección soporte.sacdni@policia.es

Hola buenos días la mayor parte de los enlaces que Vd. Nos manda son ajenos a nuestro servicio, lo que nosotros tenemos y es a lo que nos dedicamos.

Es a lo que esta en nuestra Web de la policía, en DNI y pasaporte es donde esta todo lo referente a nuestro DNI y sus certificados de hay en adelante no le puedo facilitar mas información lo sentimos perdone.

Gracias

Esperando haber aclarado sus dudas, quedamos a su disposición.

Reciba un cordial saludo.

Soporte Servicio de Atención al Ciudadano

soporte.sacdni@policia.es

De: Adrià Vilanova Martínez [mailto:]

Enviado el: miércoles, 17 de agosto de 2016 20:05

Para: Soporte DNI; oficinatecnica@dnielectronico.es

Asunto: Re: ZonaTic y establecimiento del canal seguro

;Hola de nuevo!

No obtuve respuesta desde la dirección de correo electrónico soporte.sacdni@policia.es, así que mando también al correo oficinatecnica@dnielectronico.es.

Un saludo.

Adrià Vilanova Martínez (@avm99963)

2016-07-16 14:22 GMT+02:00 Adrià Vilanova Martínez <jocdeladria@gmail.com>:

Buenos días,

Primero de todo agradecerles la respuesta que me dieron hace dos semanas sobre el driver del DNIE para Mac.

Les explicaré el motivo de esta consulta. Aquí en Cataluña en el bachillerato es obligatorio realizar un trabajo de investigación, y decidí que el objeto del mío serían las votaciones por Internet. Después de investigar un poco, he decidido que el DNIE es la mejor manera para poder autenticar a los ciudadanos en la aplicación que estoy desarrollando. Esta aplicación para ordenador se comunica mediante la api de PC/SC lite con el lector de DNIE.

Durante estas semanas en varias especificaciones y páginas web publicadas en Internet he podido aprender cómo enviar comandos APDU al DNIE. Lo que ocurre es que varias de las páginas, que según mi entender estaban mantenidas por la policía o relacionados (por ejemplo red.es) y que me ayudarían mucho a comprender la comunicación entre la aplicación y el DNIE, ya no existen. Me refiero a las siguientes, entre otras que

no apunté: <http://opendnie.morfeo-project.org/>,
<http://www.zonatic.usatudni.webcastlive.es/>,
<http://zonatic.usatudni.es/> y <http://usatudni.es/>.
¿Ustedes saben si estas páginas web van a volver o
dónde se puede consultar la información relativa a cómo
comunicarse con el DNIE mediante los comandos APDU y
ejemplos?

Hace poco vi y puse en marcadores la página
<http://www.zonatic.usatudni.webcastlive.es/>, en la cual
había varios vídeos relativos a la comunicación con el
DNIE. Mediante esos vídeos y otras fuentes de
información de momento he conseguido leer todos los
archivos de la zona pública, pero me faltaba ver la
parte relacionada con cómo establecer el canal seguro
(para poder verificar el PIN y conseguir el certificado
de autenticación). Desafortunadamente, durante estos
días esta página web ha caído y ya no está
disponible... Por eso me he quedado a medias en
entender cómo se establece el canal seguro. Si ustedes
pudieran ayudarme a entender cómo realizar el canal
seguro estaría muy agradecido, ya que la página web
mencionada anteriormente tenía un documento PDF en el
que se explicaba cómo hacerlo, pero lo he perdido
debido a que ya no está disponible y en el manual de
comandos del DNIE no hay una explicación clara de cómo
establecerlo.

Acerca de cómo he conseguido el manual de comandos del
DNIE, intenté firmar el documento de aceptación de
condiciones para poder conseguir el manual, pero
desafortunadamente NO pude firmar el documento debido a
que soy menor de edad y mi DNIE no contempla el
certificado de firma electrónica. Después buscando un
poco por Internet logré encontrar una copia que
enviasteis a otra persona y lo he descargado para
documentarme en el aspecto académico para poder
realizar mi trabajo de investigación.

Por todo esto, estaría muy agradecido si me pudieran
ayudar con el establecimiento del canal seguro ya que
la mayoría de páginas web de referencia del DNIE están
caídas y no hay ningún sitio en el que poder encontrar
la información.

Muchísimas gracias por adelantado, y siento mucho si
este correo era demasiado largo.

Adrià Vilanova Martínez (@avm99963)

Annex B. APDU commands

Application Processing Data Unit (APDU) commands are series of bytes which send specific messages that the smart card can understand and reply back. Each byte can be represented in hexadecimal base from number 00 to number FF, therefore each byte is made up of 8 bits (each bit can be a 1 or a 0, so the combinations are $2^8=FF$).^[20]

Each command has the following structure:^[22]

CLA	INS	P1-P2	L _C	Data	L _e
1 byte	1 byte	2 bytes	0 or 1 byte	L _C bytes	0-3 bytes

- **CLA:** Class introduction. Indicates the type of command (interindustry or proprietary).
- **INS:** Instruction code. It states the type of command which will be run (for example, read data, write data, perform security operation, etc.)
- **P1-P2:** Two instruction parameters for the command (for example, the offset from which we want to read the data of one file)
- **L_C:** The length in bytes of the Data field.
- **Data:** L_C bytes of data that can be passed to some specific commands, such as write file.
- **L_e:** maximum expected length in bytes of the response.

So an APDU command is the concatenation of all those bytes. For example, let 00 20 0000 17 4e6576657220676f6e6e61206769766520796f75207570 be one APDU command (in reality the spaces between the hexadecimal numbers are not present, but I have added them for a better representation). This is its teardown and what each part means:

- 00: this byte is the CLA, and tells us that this command is being sent as plain text (it is not being sent encrypted) and according to the ISO/IEC 7816 standard.

- `a4`: this byte is the INS. It tells us that this APDU corresponds to the “verify” command. This command is used to verify the smart card’s PIN, or more specifically, CHV (Card Holder Verification), because a CHV is an alphanumeric code which can also has special characters and is used to verify the card’s owner and unlock the smart card, while a PIN has the same function but is only numeric.^[23]
- `0000`: these are the P1-P2 bytes. As this command does not accept parameters, each of them are set to 0.
- `17`: this is the L_c byte. This command requires some data to be passed (the CHV), so we specify the number of bytes of data that we are going to send to the smart card.
- `4e6576657220676f6e6e61206769766520796f75207570`: these are the data bytes. It is the hexadecimal representation of the CHV in ASCII.

After the command is send, the application receives a response from the smart card. This response has another structure similar to the one of the sent command:

Data	SW1 - SW2
Max. L_c bytes	2 bytes

- **Data**: (optional) the response data, including information that was requested by the command.
- **SW1-SW2**: Status bytes. Depending on their value they represent if the command was successfully run. If the command was successfully run, then the status bytes would be `90 00`.

For our example command before, the response could be `9000`, meaning that the CHV was correct, `63C1` meaning that it was incorrect and there is one attempt left, or `6581` meaning that there was a memory error, but there are a lot of response codes which are listed in the website of reference 24.^[24]

Annex C. RSA

RSA is a cryptosystem used to encrypt sensitive information. RSA stands for Rivest-Shamir-Adleman, the people who invented it.^[26]

This cryptosystem is based on various things with respect to its security. First of all, it is based in the fundamental theorem of arithmetic. This theorem states that any number can be broken down to the multiplication of prime numbers. If the number can be broken down into many numbers different than 1, then it is a composite number. If not, it is a prime number. Secondly, it is based in the discrete logarithm problem. This states that $a^1 \bmod x, a^2 \bmod x, \dots$, will produce an uniform distribution. And finally, it is based in a one-way function combining the two previous principles: a one-way function is a function which is easy to perform, but is hard to reverse.

The cryptosystem works as follows: before encrypting a message, the receiver generates two prime numbers ("p" and "q"), computes the modulus ($n = pq$) and generates values for the public and private exponents ("e" and "d") according to some restrictions. Then, the receiver sends the public key (numbers "n" and "e") to the sender so the sender can encrypt their message.^[27]

To encrypt the message, the sender uses the following formula, where "m" is the numerical representation of the message, "e" and "n" are the public key numbers (public exponent and modulus) and "c" is the encrypted message:

$$c \equiv m^e \pmod{n}$$

To decipher the message, the receiver uses the following formula, where "d" is the private exponent:

$$m \equiv c^d \pmod{n}$$

Annex D. Secure Messaging

After establishing a secure channel, we have two keys: K_{enc} and K_{mac} , and an incremental number SSC . These numbers are used to encrypt APDU commands and decrypt APDU responses.

To encrypt APDU commands we have to follow the steps shown in the following diagram:

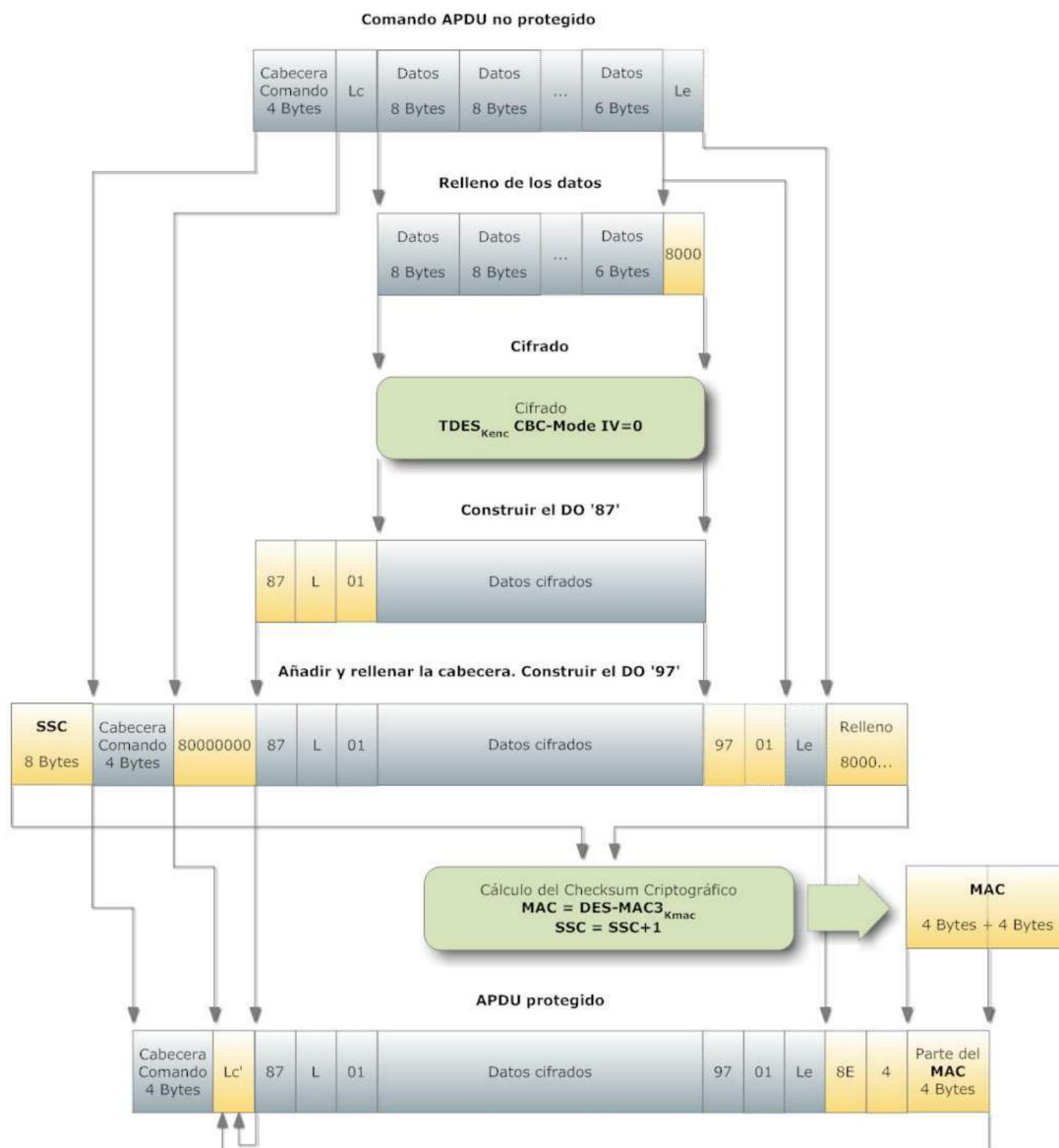


Figure 21. Official method to encrypt an APDU command for the DNIe (*in Spanish*)^[31]

To decrypt APDU commands we have to follow the steps shown in the following diagram:

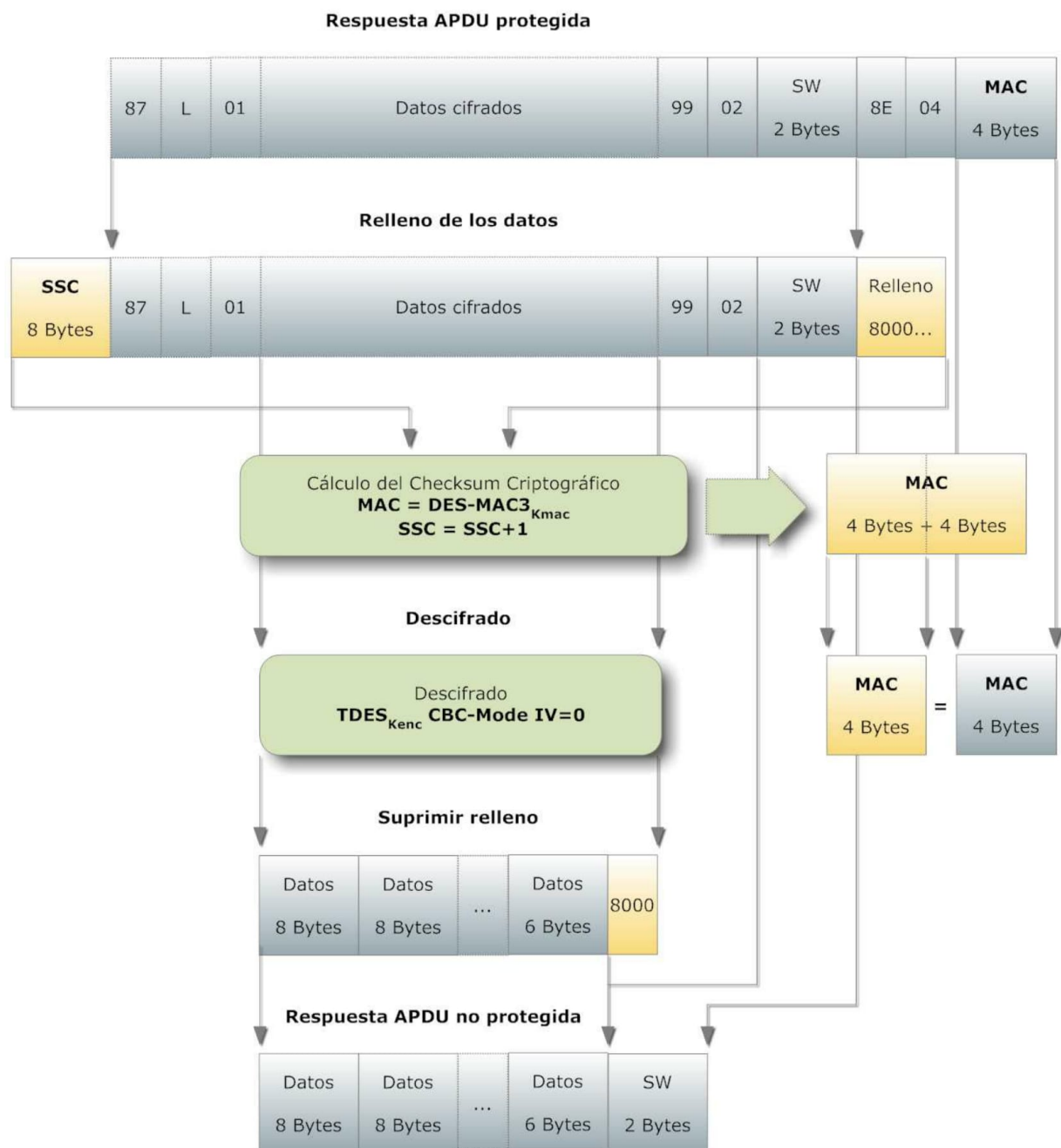
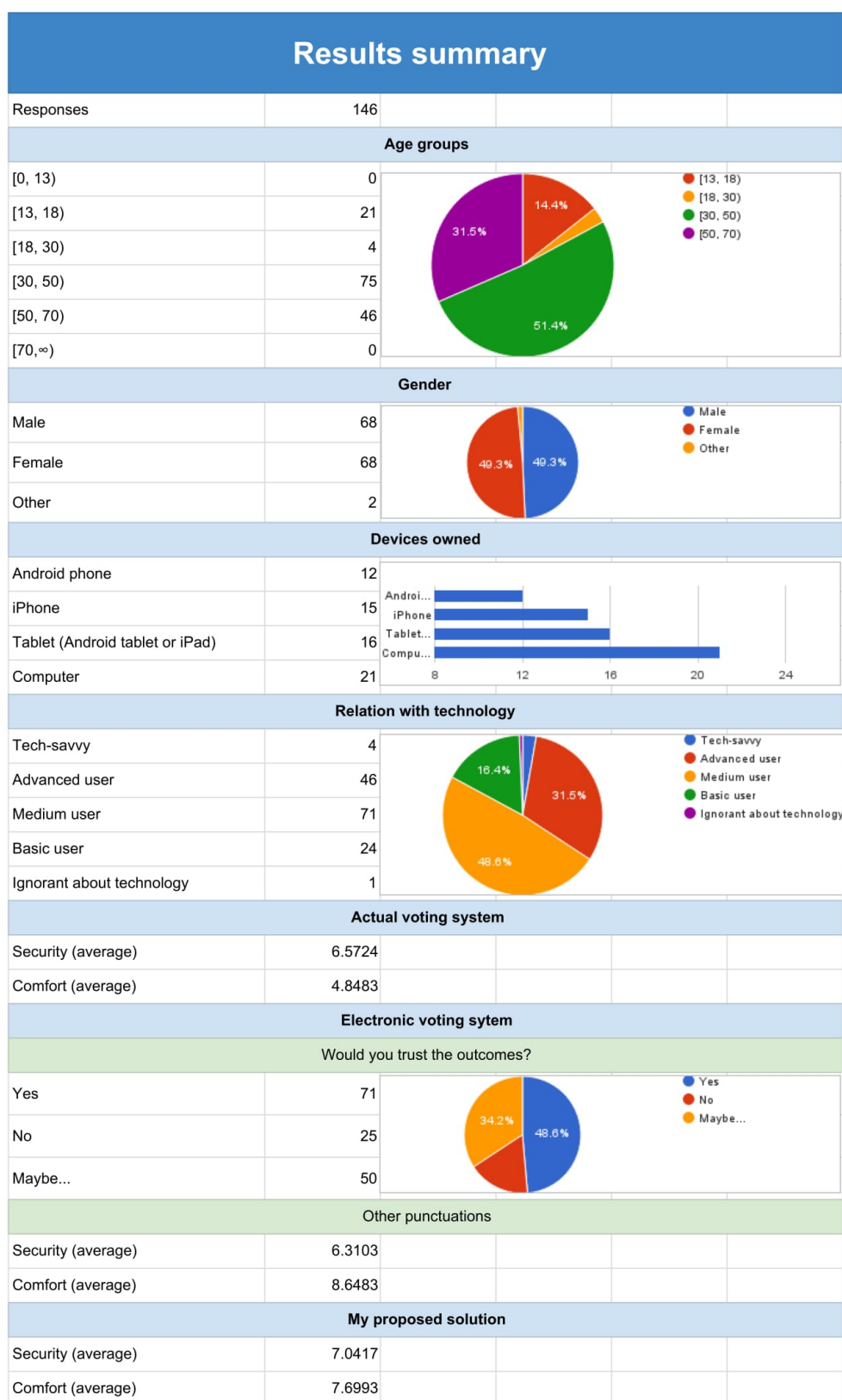


Figure 22. Official method to decrypt an APDU response for the DNIE (*in Spanish*)^[31]

Annex E: Survey



Annex F. Source code

The source code of the proposed electronic voting system can be found in the following GitHub repository:

<https://github.com/avm99963/voting/>



Also, you can visit the Knowledge Base with information on how to use the software in the following website:

<https://voting-kb.herokuapp.com/>

Bibliography

- [1] Junquera, Natalia. "El 20-N habrá 500 millones menos de papeletas." (November 5, 2011) http://elpais.com/diario/2011/11/05/espana/1320447630_850215.html
- [2] Romero, H. "Unidad Didáctica del papel." (November 14, 2011) http://greeningbooks.eu/index.php?option=com_docman&task=doc_details&gid=88
- [3] "¿Cuánto papel se puede obtener de un árbol?" (September 8, 2014) <http://www.imprimirmirevista.es/blog/cuanto-papel-se-puede-obtener-de-un-arbol/>
- [4] "¿Sabías cuántos árboles hacen falta para fabricar papel?" (April 21, 2014) <http://losporquesdelanaturaleza.com/sabias-cuantos-arboles-hacen-falta-para-fabricar-papel/>
- [5] Galli, Ricardo. "El voto electrónico y las elecciones de Podemos." (October 30, 2014) <https://gallir.wordpress.com/2014/10/30/el-voto-electronico-y-las-elecciones-de-podemos/>
- [6] Galli, Ricardo. "Las elecciones de Podemos, voto electrónico, y AgoraVoting (y 2)." (November 6, 2014) <https://gallir.wordpress.com/2014/11/04/las-elecciones-de-podemos-voto-electronico-y-agoravoting-y-2/>
- [7] Wikipedia contributors. "Voting system." (October 15, 2016) https://en.wikipedia.org/w/index.php?title=Voting_system&oldid=744403994
- [8] Ribeiro, Ricky. "Paper-Based vs. Electronic Voting: States Move in Different Directions." (March 29, 2016) <http://www.statetechmagazine.com/article/2016/03/paper-based-vs-electronic-voting-states-move-different-directions>
- [9] Panda Security. "Electronic voting may not be 100% secure (but neither is traditional voting)" <http://www.pandasecurity.com/mediacenter/news/electronic-voting-may-not-be-100-secure-but-neither-traditional-voting/>
- [10] Wikipedia contributors. "D'Hondt method." (September 27, 2016) https://en.wikipedia.org/w/index.php?title=D%27Hondt_method&oldid=741464671
- [11] Wilson, Helen J. "The D'Hondt Method Explained." <http://www.ucl.ac.uk/~ucahhwi/dhondt.pdf>

- [12] BBC News. "D'Hondt explainer." (November 5, 2009)
<https://www.youtube.com/watch?v=6CU3F3ToIIg>
- [13] Cuerpo Nacional de Policía. "Diferencias DNIE y DNIE 3.0."
https://www.dnielectronico.es/PortalDNIE/PRF1_Cons02.action?pag=REF_038
- [14] Locolandia. "Autenticación DNIE con Apache2 y PHP."
<http://doku.locolandia.net/howto/apache-dnie-auth>
- [15] Rinconastur. "Extraer información de certificados."
<http://www.rinconastur.com/php/php24.php>
- [16] Google Chrome Help Center. "What is a Chrome kiosk app?"
<https://support.google.com/chrome/a/answer/6137028?hl=en>
- [17] The Chromium authors. "Kiosk Apps."
https://developer.chrome.com/apps/manifest/kiosk_enabled
- [18] Chrome for Business and Education Help Center. "Use Smart Cards on Chrome OS." <https://support.google.com/chrome/a/answer/7014689?hl=en>
- [19] Google. "Smart Card Connector."
<https://chrome.google.com/webstore/detail/smart-card-connector/khpfeaanjngmcnplbdlpegiifgpfgdco>
- [20] OpenSCDP. "Read EMV."
<http://www.openscdp.org/scripts/tutorial/emv/reademv.html>
- [21] Cuerpo Nacional de Policía. "Características Chip DNIE"
https://www.dnielectronico.es/PortalDNIE/PRF1_Cons02.action?pag=REF_240
- [22] Wikipedia contributors. "Smart card application protocol data unit." (September 9, 2016)
https://en.wikipedia.org/w/index.php?title=Smart_card_application_protocol_data_unit&oldid=738584393
- [23] Cuerpo Nacional de Policía. "Guía de Referencia Técnica." (September 30, 2010)
<http://myslide.es/documents/20100930-manual-de-comandos-del-dnie-tractis.html>
- [24] EFTlab Ltd. "Complete list of APDU responses."
<https://www.eftlab.com/index.php/site-map/knowledge-base/118-apdu-response-list>
- [25] viktorTarasov, timofonic. OpenSC. "DNIE (OpenDNIE)". (August 29, 2013)
[https://github.com/OpenSC/OpenSC/wiki/DNIE-\(OpenDNIE\)](https://github.com/OpenSC/OpenSC/wiki/DNIE-(OpenDNIE))

- [26] TechTarget. "RSA algorithm (Rivest-Shamir-Adleman)." <http://searchsecurity.techtarget.com/definition/RSA>
- [27] Wikipedia contributors. "RSA (cryptosystem)." (October 24, 2016) [https://en.wikipedia.org/w/index.php?title=RSA_\(cryptosystem\)&oldid=745936677](https://en.wikipedia.org/w/index.php?title=RSA_(cryptosystem)&oldid=745936677)
- [28] Steyn, Barry. "How RSA Works With Examples." (May 26, 2012) <http://doctrina.org/How-RSA-Works-With-Examples.html>
- [29] Seoane, Domingo. "Leer los datos públicos del DNIE." (May 25, 2012) <https://delphi.jmrds.com/node/78>
- [30] Vicente Vallejo, Javier. "Análisis de la estructura interna del DNI electrónico" (March 16, 2010) https://vallejocc.files.wordpress.com/2014/12/dnie_analisis.pdf
- [31] ZonaTic. "Diagrama canal seguro – Navega a través del interior del DNIE" <http://zonatic05.webcastlive.es/> (no longer available)
- [32] Vilanova Martínez, Adrià. "Internet voting: a utopia? (Responses)" (October 27, 2016) https://docs.google.com/spreadsheets/d/1s1_lxndx7JeFyaKnpicqv-CDv0B5J_AEqW09TvbMMEU/edit

