



# Digital Office

03.06.2019

---

Jaime Armando Bastida Prado

Daniel Rafael Solorio Paredes

## Introduction

The CEO of an important organization is trying to digitize several processes. In particular they are trying to generate the following documents automatically:

- Minute

Minute is an official written statement of the motions and resolutions taken in a meeting. It is brief but a complete record of all discussions held among the members of the meeting.

- Memorandum

Memorandum is a note, document or other communication that helps the memory by recording events or observations on a topic such as may be used in a business office.

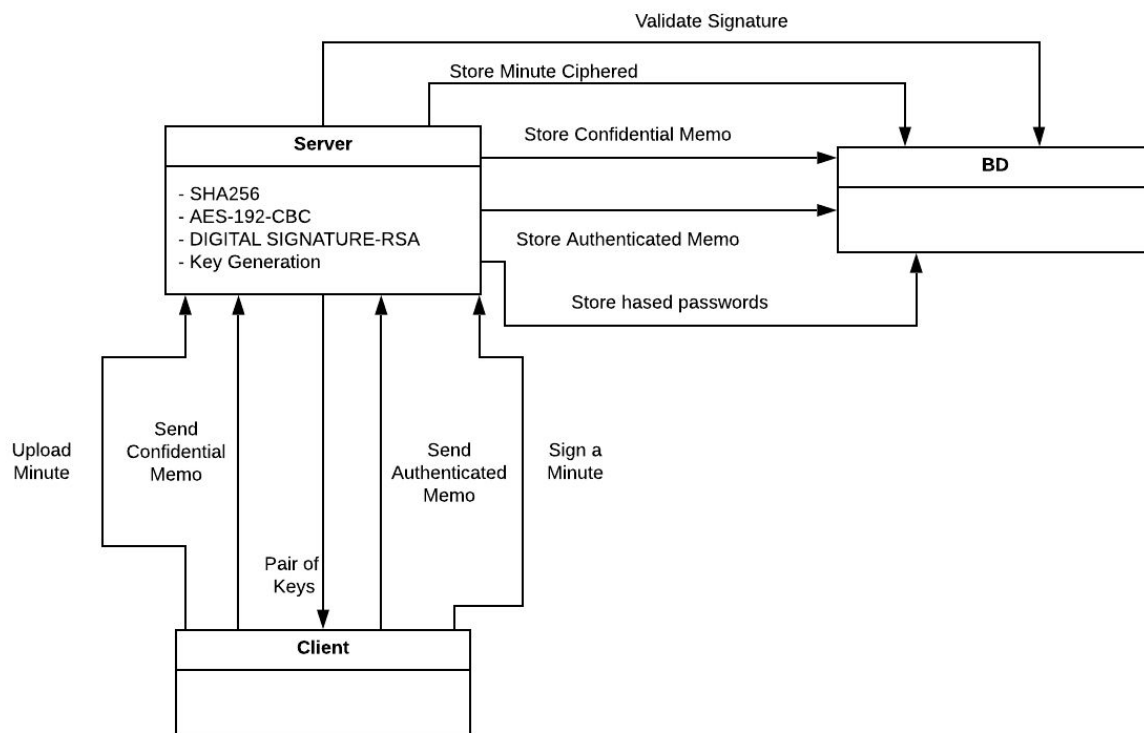
- Confidential Memorandum

Confidential memorandum is the same as a memorandum, but this must be kept in secret. This kind of document must be seen only by the sender and the receiver.

This documents have special attentions:

- Any minute requires to be signed by every participating member in the meeting.
- Memorandums require the signature of the person who write the memorandum.
- Confidential memos, require the signature of the person who writes the memorandum and also that only those persons authorized to see its content can read it.

## Solution



## Cryptographic Services

### I. Authentication

Authentication occurs when the user wants to log in. This is achieved because the user previously has signed up in the system and has registered his mail and password. So, when the user wants to log in, credentials provided by the user are compared to those on file in a database. If the credentials match, and the authenticated entity is authorized to use the resource, the process is completed and the user is granted access.

This service is also provided in the process of signing the minute. Once the CEO of the company has already uploaded the minute, the users that have taken place in

the meeting may sign the minute. The minute, which is a Word Document, is hashed with 'SHA-256' the result of that hash will be called "h". Then, we take "h" and the private key of the signer user in order to make use of a Public Key Algorithm and getting the Digital Signature. At this time, when the Digital Signature has already generated, the user can not deny he has not signed the minute. Because he has already logged in with his password and signed with his private key, which they are supposed to not be known by anyone else.

So, authentication is important because it enables our system to keep the minutes and memos secure by permitting only authenticated users to access its protected resources. Also, it helps by guaranteeing the receiver of the memo that the memo was sent by a user of the system and not from an unknown sender.

## II. Confidentiality

The password of the users is Hashed and then stored in the database. So, the password is not stored in plain text.

When a confidential memorandum is going to be sent. First, the message is encrypted with AES using a 24 bytes long password submitted by the sender. Second, the password is encrypted with RSA using the public key of the receiver. So, a specific receiver only will be able to read the memorandum.

Our system works this way, encrypting sensitive files in order to protect them from being read or used by those who are not entitled to do either. This is very important because a failure in properly secure and protect confidential business information can lead to the loss of business/clients.

In the wrong hands, confidential information can be misused to commit illegal activity (e.g., fraud or discrimination), which can in turn result in costly lawsuits for the employer.

## III. Integrity

This cryptographic service appears in the signing minute process. When the minute is already uploaded, the users may sign the document with their private keys. As is mentioned in the Authentication service, the document is hashed with "SHA-256". This provides assurance that data has not been modified in an unauthorized manner after it was created, transmitted or stored. The Digital Signature that is implemented in this process helps to notice that there has been no insertion, deletion or substitution done with the data.

This cryptographic service is very important because if the minute would have been modified and there is no way to know if it was modified. The users would be confused about the motions and resolutions taken in the meeting. So, some users would agree in something the other users would not know about. And it would result in data inconsistency.

#### IV. Non repudiation

The process of signing a minute uses Digital Signature. This means, the user that sign the minute introduces his private key, then the system verifies if the public key matches with the given private key. As the private key is only known by the user and the user at this point, has already introduced his password to log in. There is no way to deny he has signed the minute.

So, our system has the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated.

The system applies this cryptographic service in order to give the CEO the certain that all the users that assist to the meeting agree with the motions and resolutions that give place in the meeting. The users can not tell that they would not have signed the minute because they were not agree, and try to convince the signature is fake.

### Cryptographic Primitives

#### 1. CBC

This primitive provides information security such as confidentiality or authenticity.

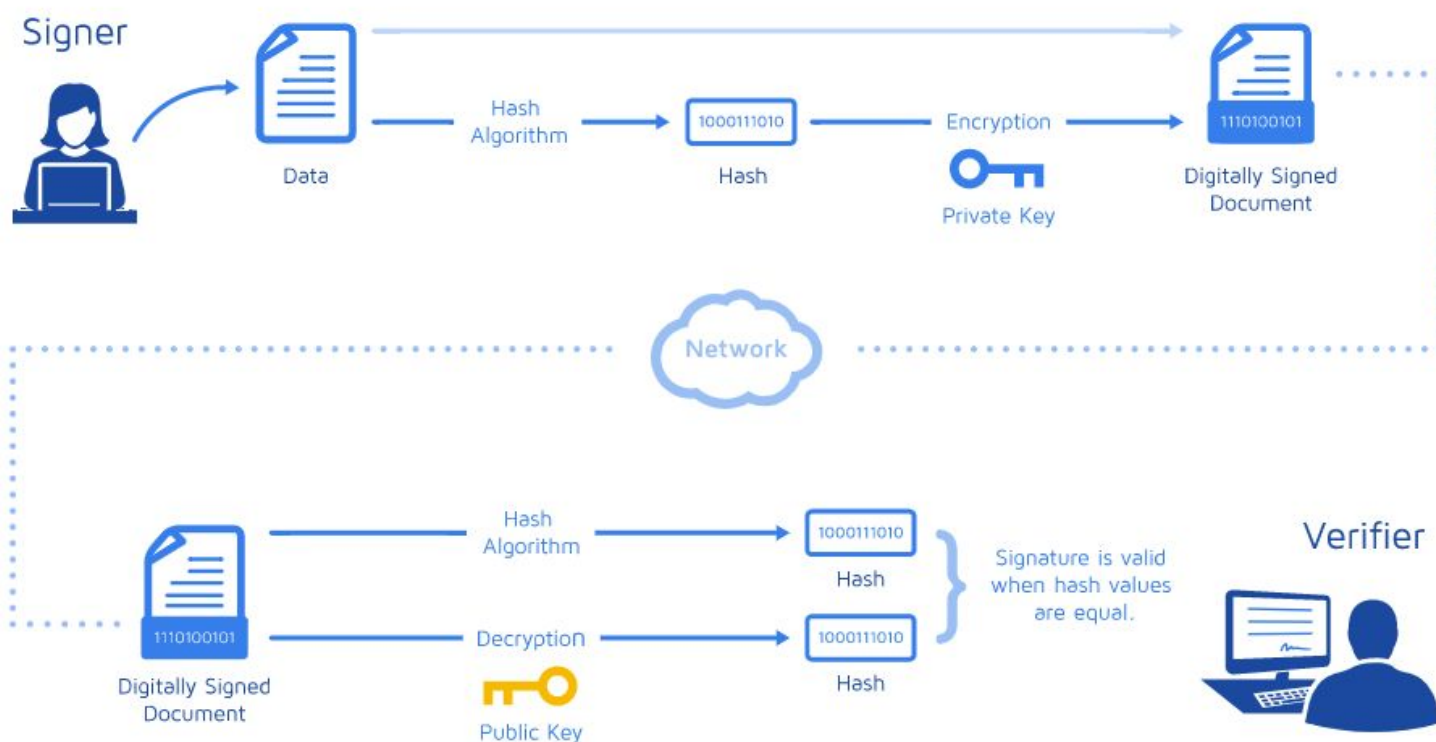
#### 2. AES-192

This is a symmetric block cipher used for protect classified information and is implemented in software to encrypt sensitive data, providing confidentiality.

#### 3. HASH FUNCTION “SHA-256”

A hash function will give the same hash for the same input always no matter when, where and how you run the algorithm. Equally interestingly, if even one character in the input text or data is changed, the output hash will change. Also, a hash function is a one-way function, thus it is impossible to generate back the input data from its hash. So, this primitive provides Data Integrity.

## 4. Digital Signature



This primitive provides Authentication and Data Integrity.

## 5. RSA

In RSA cryptography, both the public and the private keys can encrypt a message; the opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason why RSA has become the most widely used asymmetric algorithm: It provides a method to assure the confidentiality, integrity, authenticity, and non-repudiation.

## Installation Manual

- Operating System
  - Linux
  - Mac OS X 10.7 and higher

- Windows 7 and higher
- Install Node.js

Here is the link where you can download it: <https://nodejs.org/es/>

For best performance, Digital Office recommends using the latest LTS version of Node.js.

1. Open PowerShell Terminal in the path where the folder of the project is.
  2. Enter "npm run dev"
  3. Open an Internet Browser
  4. Enter "<http://localhost:5000/users/index>"
- Datastores
    - MongoDB

Here is the link where you can manage your database in MongoDB:  
[https://cloud.mongodb.com/user?\\_ga=2.80940242.492516833.1559577395-258220178.1559169583#/atlas/login](https://cloud.mongodb.com/user?_ga=2.80940242.492516833.1559577395-258220178.1559169583#/atlas/login)

You have to create a new account. Then log in.

- Internet Browsers
  - Firefox
    - Mozilla Firefox 22 and higher

For best performance, Digital Office recommends using the latest version of Firefox

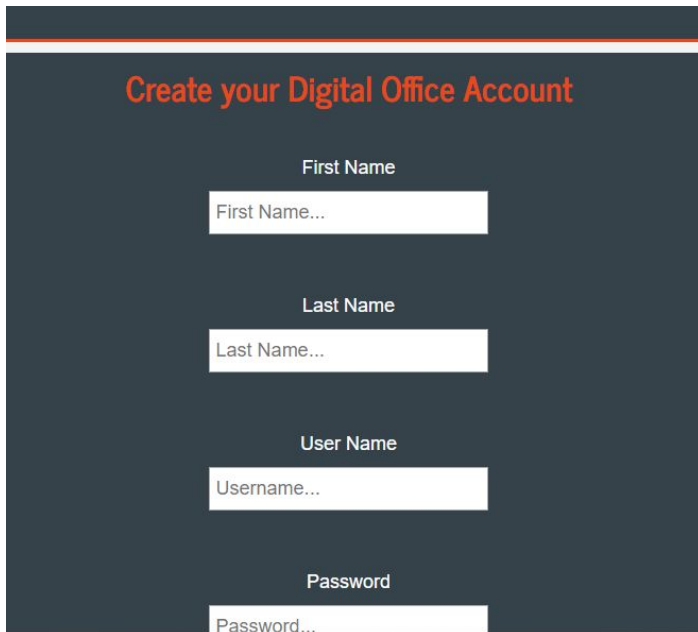
- Google Chrome
  - Google Chrome 18.0.1025.39 and higher

For best performance, Digital Office recommends using the latest version of Google Chrome

## User Guide

### Creating an Account

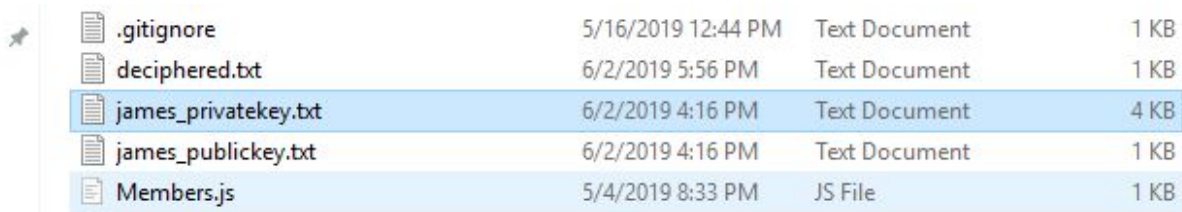
You can create an account on the system simply by writing down all the fields needed and your password:



The screenshot shows a dark-themed web form titled "Create your Digital Office Account" in orange text. Below the title are four input fields, each with a label above it: "First Name" (with placeholder "First Name..."), "Last Name" (with placeholder "Last Name..."), "User Name" (with placeholder "Username..."), and "Password" (with placeholder "Password...").

This password must be kept in absolute secret, and will be used in several cases further in the guide.

The system also will generate a pair of keys, known as Public Key and Private Key, stored into a text file that must be handed to the user.



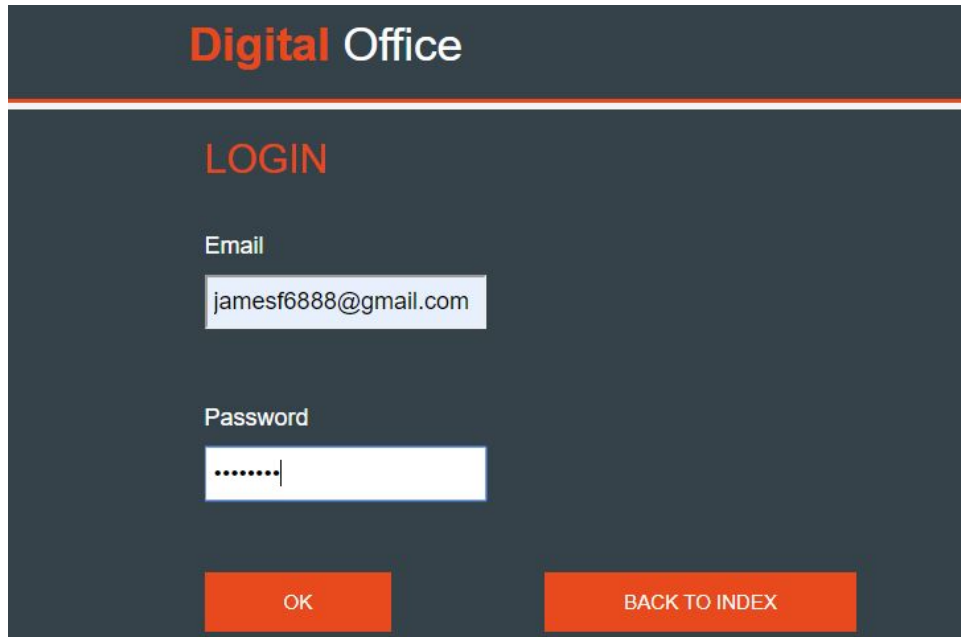
The screenshot shows a file explorer window with a list of files. The files are: .gitignore (5/16/2019 12:44 PM, Text Document, 1 KB), deciphered.txt (6/2/2019 5:56 PM, Text Document, 1 KB), james\_privatekey.txt (6/2/2019 4:16 PM, Text Document, 4 KB), james\_publickey.txt (6/2/2019 4:16 PM, Text Document, 1 KB), and Members.js (5/4/2019 8:33 PM, JS File, 1 KB). The file james\_privatekey.txt is highlighted in blue.

.gitignore	5/16/2019 12:44 PM	Text Document	1 KB
deciphered.txt	6/2/2019 5:56 PM	Text Document	1 KB
james_privatekey.txt	6/2/2019 4:16 PM	Text Document	4 KB
james_publickey.txt	6/2/2019 4:16 PM	Text Document	1 KB
Members.js	5/4/2019 8:33 PM	JS File	1 KB



## Login In

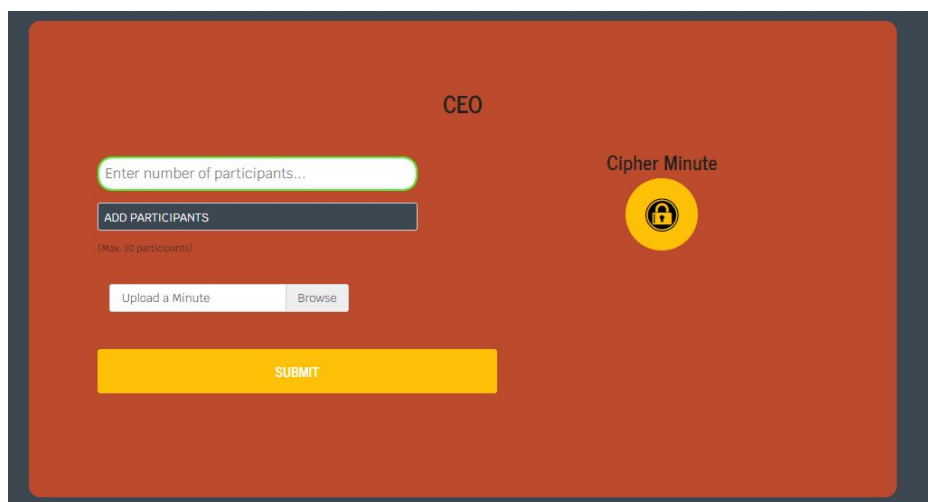
You can log into the system simply by writing your email and the password your selected before:



The image shows a login interface for 'Digital Office'. The header 'Digital Office' is in orange and white text on a dark blue background. Below it, the word 'LOGIN' is in orange. There are two input fields: 'Email' with the value 'jamesf6888@gmail.com' and 'Password' with masked characters '.....'. At the bottom, there are two orange buttons: 'OK' and 'BACK TO INDEX'.

## Visiting the home page

When you enter to the system, it will display all the options available according to your type of user. If you are a "CEO" then you will see all the options including this section:

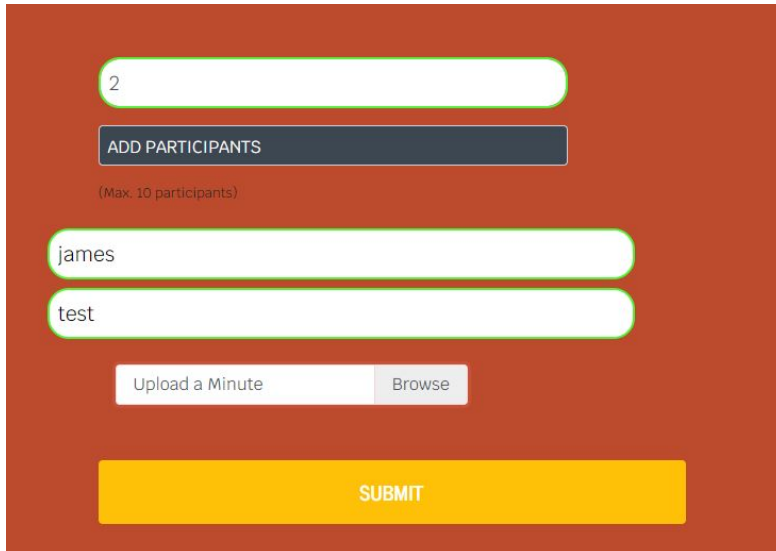


The image shows a home page for a 'CEO' user. The background is dark blue. A large orange rectangle contains the following elements: the text 'CEO' at the top; a text input field 'Enter number of participants...' with a green border; a dark blue button 'ADD PARTICIPANTS' below it, with '(Max. 10 participants)' in small text underneath; a text input field 'Upload a Minute' and a 'Browse' button; and a large yellow 'SUBMIT' button at the bottom. To the right of the orange rectangle, the text 'Cipher Minute' is above a yellow circular icon with a black padlock.

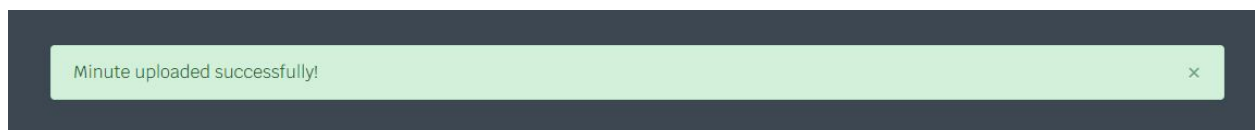
When you log in as a “Common User” you won’t see this section, as you are not capable of uploading or ciphering Minutes.

## Uploading a Minute

When you upload a Minute, you must indicate how many and who are the users that were participants on the meeting, this fields must be filled with the “username” of the user not its “first name”:

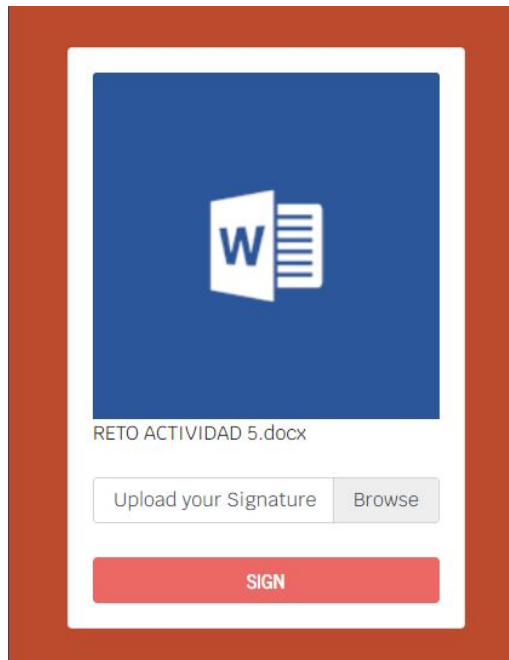
A screenshot of a web form for uploading a minute. The form has a reddish-brown background. At the top, there is a text input field containing the number '2'. Below it is a dark blue button labeled 'ADD PARTICIPANTS'. Underneath the button, in smaller text, it says '(Max. 10 participants)'. Below this, there are two more text input fields, the first containing 'james' and the second containing 'test'. At the bottom of the form, there is a light gray button labeled 'Upload a Minute' and a smaller gray button labeled 'Browse'. At the very bottom, there is a large yellow button labeled 'SUBMIT'.

This message should appear:

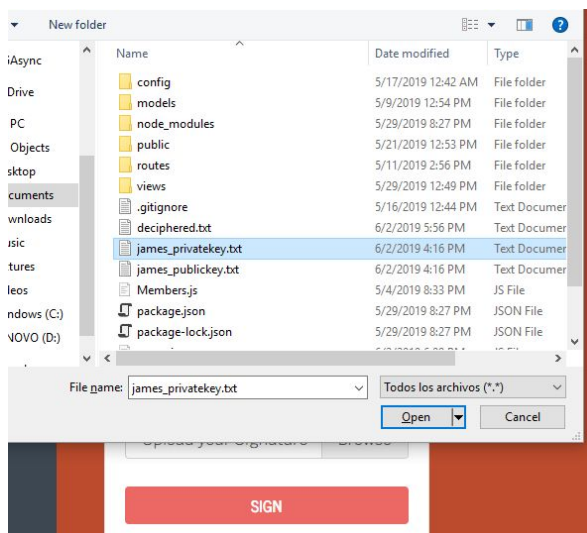


## Signing a Minute

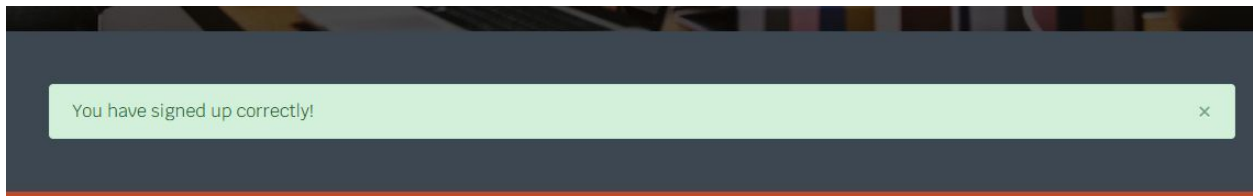
The system will allow you to sign a Minute only if you appear in the list of participants of that Minute and if you have not signed yet:



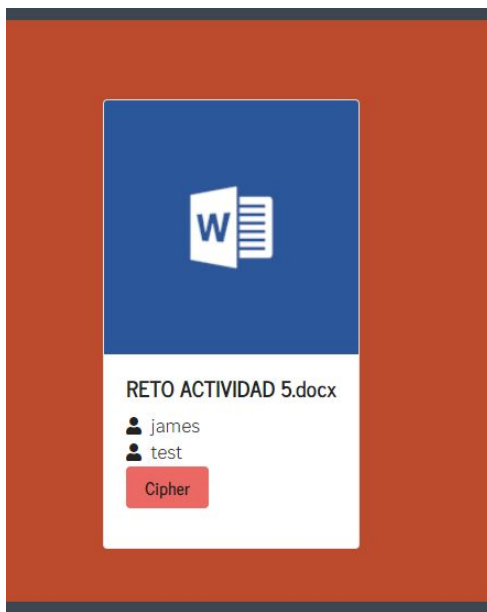
To sign you must upload the file in which you have your "Private Key" this file was given to you when you created your account.



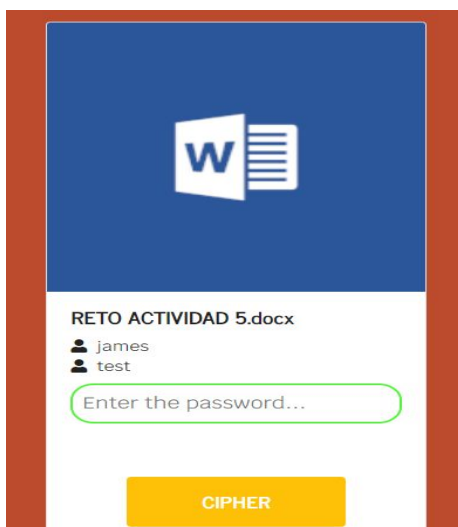
If you signed up correctly i.e. your txt file has not been corrupted by you or you signed with the correct file, then this message will appear:



If you are a “CEO” you have the possibility of cipher a Minute when all the participants of the meeting have signed the Minute. If not, the Minute will appear with a list of the participants but you won’t be able to cipher with the password that was given to you personally:



If all users have signed then:



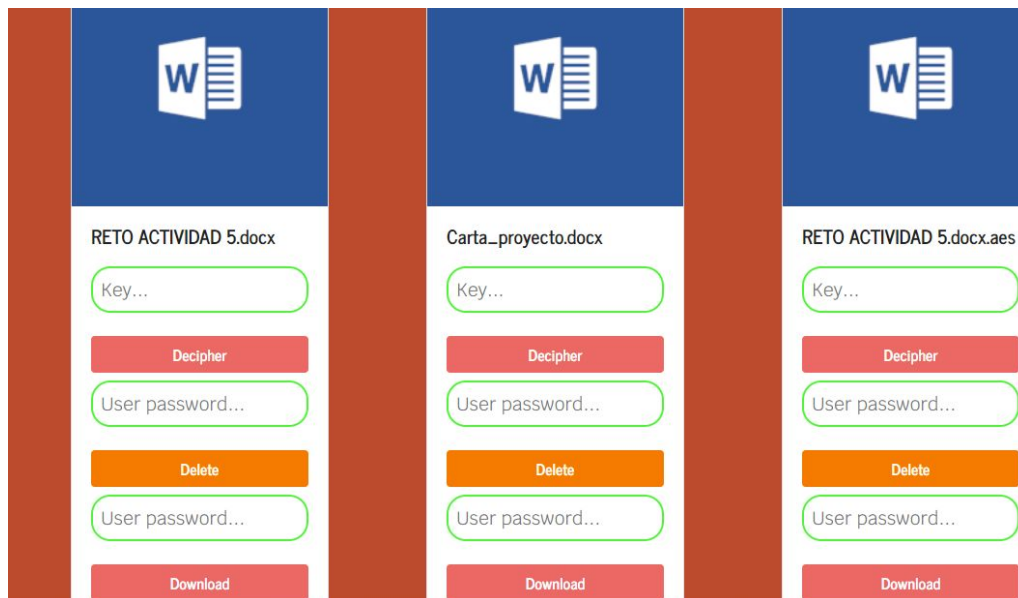
You will be able to cipher using your key. Also mention this key will be used to cipher using “AES-192-CBC” thus the length of the key must be of 24 characters.

If you have ciphered up correctly this message will appear:



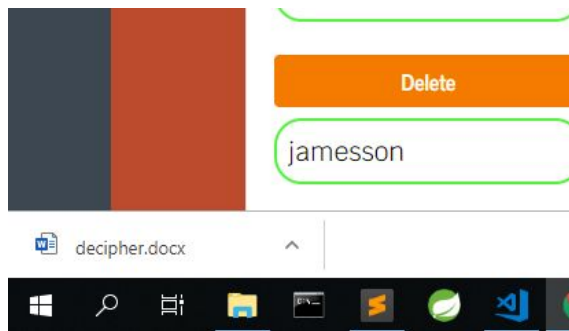
## Visiting all Minutes

When you visit the “Show Minutes” section you will be able to see all Minutes in the system including the ones that are ciphered which have the extension “.aes”. From here you will be able to download, delete or decipher a Minute as you want. If a Minute has not the “.aes” extension you won’t be able to decipher it.



To decipher a Minute you must enter your key as a “CEO”. To download or delete it you must enter your user password.

When you decipher a Minute it will be downloaded to your computer:



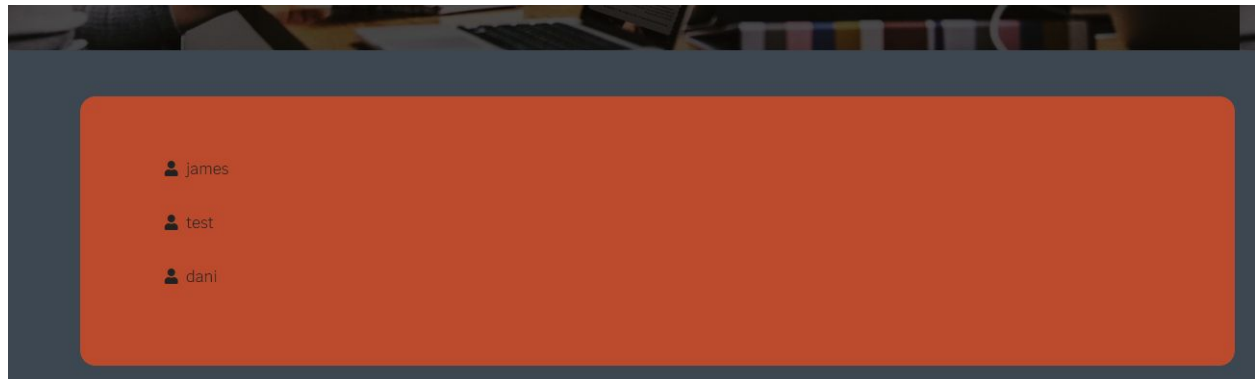
This file always will have the name "deciphered.docx" is up to you to change its name.

When you delete one, a message like this one will appear:



## Sending a Confidential Memo

Sending a Confidential Memo is as easy as choosing the user you want to send it:

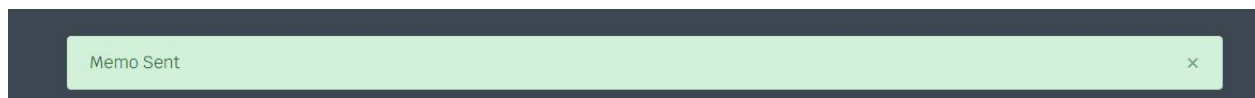


Filling all the information including a password that must be 24 characters long:

A screenshot of a web application form for sending a memo. The form has a dark orange background. At the top, there's a white input field labeled 'Subject...'. Below it is a large, light orange rectangular area labeled 'Memorandum...'. At the bottom, there's another white input field labeled 'Password...'. Below the password field is a yellow button with the text 'SEND' in black capital letters.

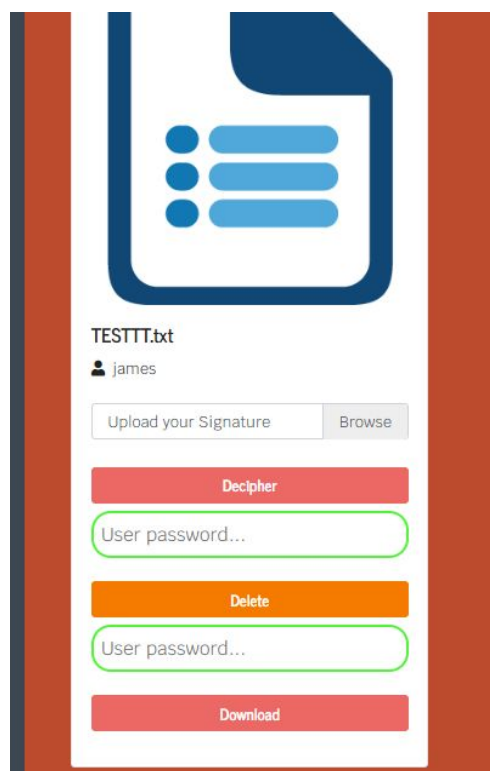
This password will be used to cipher the Memo, and the password will be ciphered using the Public Key of the receiver.

A message like this will appear:

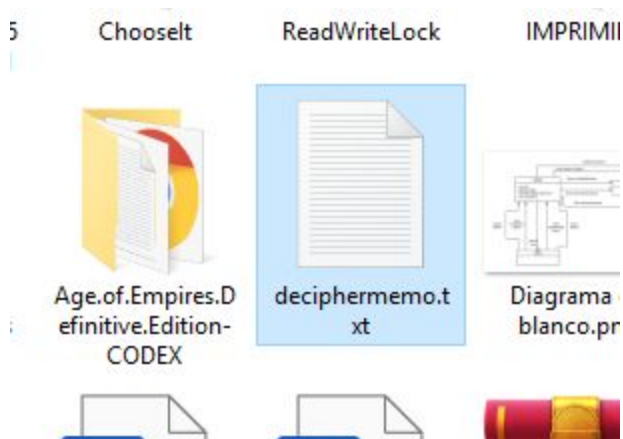


## Visiting the Memos

You can watch all Memos (Confidential or Authenticated) sent to you in the “View Memos” page. You will be able to see the user who sent it to you as well as deciphering it, download it or delete it:




To download a Memo or delete it you must enter your user password. To decipher it you must provide your signature “Private Key” file, and it will be downloaded to your computer:



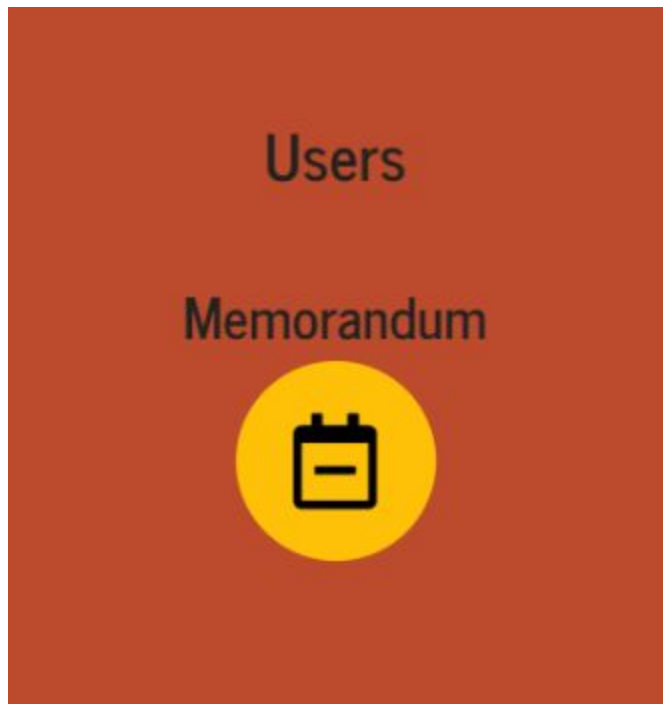
You can open it and see the content:



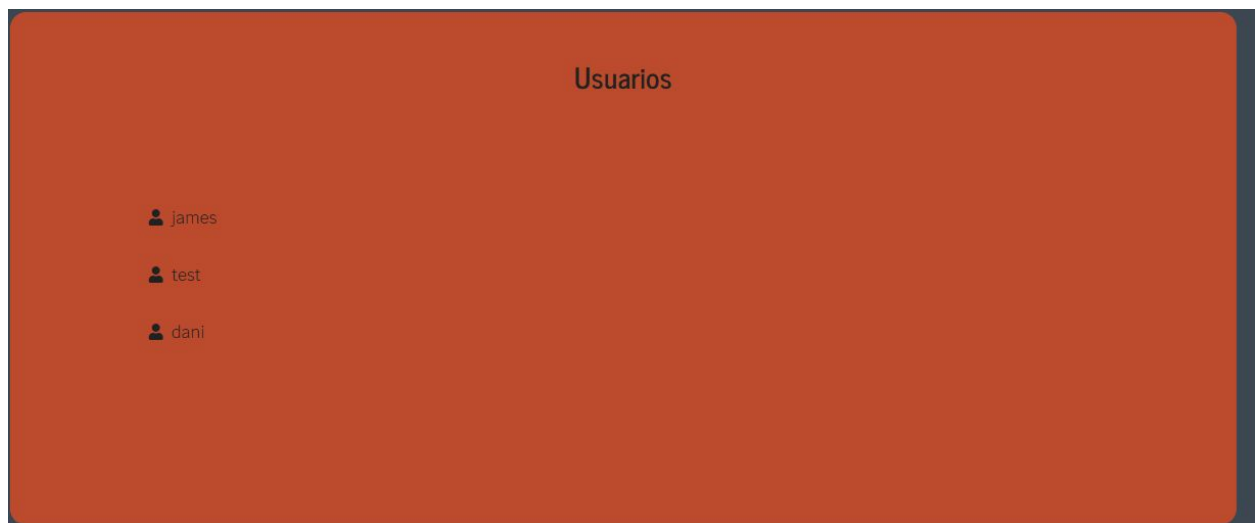
 deciphermemo.txt - NotepadFile Edit Format View Help

This is a test

## Authenticated Memo



When you visit the "Memorandum" section you will be able to send a Memo to a specific user of the system.

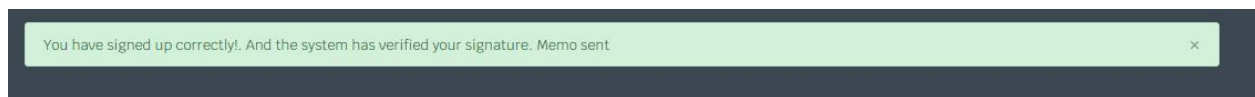


You have to select the user that will receive the memo.

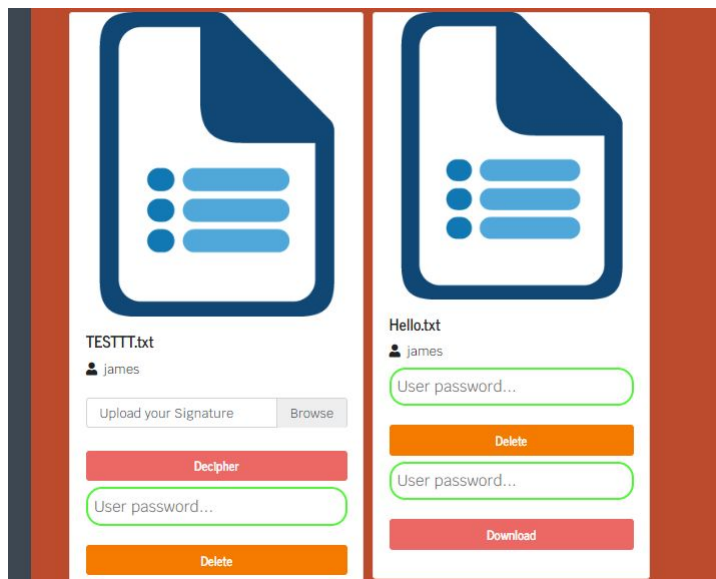
Now you can fill all the fields needed and you must provide your signature "Private Key" file, this is to ensure that you have been the one who sent the Memo:



A message like this will appear:



Now you can see the Memo in the “View Memos” page. Note that this type of Memo cannot be deciphered since it is just an Authenticated Memo:



You can download it or delete it.