**SCRIPT**

**Common User**

The common user will have three main possibilities:

* To sign a Minute.
* To send a Confidential Memo.
* To send a Memo.

**Singing a Minute:**

This action will be validated as follows:

The system will make a hash of the Minute using SHA-2.

Then it will take the user password as if it is a key and perform an HMAC with the Minute hashed and that key. This will result in a label “t”.

Then the system will take the stored password of that user in the database and perform the HMAC with the stored password and the Minute hashed. This will result in a second label lets say “t´”.

The system will check if t == t’. If this is not the case then something is wrong, like the person who is submitting the signature over the Minute is not the one who says it is or the Minute has been corrupted.

**Sending a Confidential Memo**

Any user will have the possibility to send a Confidential Memorandum. The process will be as follows:

The user will select the option of send a Memorandum

Then the system will ask for the user to type the Memo and select who to send it

Then it will ask for a particular key to be inserted, this key must be base 64 so we will restrict the user to type only the following characters: A-Za-z-0-9, +/. The user will be ask to insert a key with at least 8 and at most 16 characters.

Once the system has this key it will automatically pad it (in case user typed less than 16 characters), using PKCS to the length of the 128 bits needed for AES to encipher.

This key will be enciphered using the public key of the receiver (RSA) and sent along with the Memo.

The receiver will then use its private key to decipher the key so he can decipher the Memo.

**Sending a Memo**

This action will be as follows:

The user will select who to send the Memo, and then write the Memo, then clicking the button.

**CEO**

The CEO of the company will have two possibilities:

* To upload a Minute
* To encipher a Minute

The mechanisms that will be applied to this options are commented below

**Uploading and Enciphering a Minute**

After finishing a meeting held in the company the CEO will be the only one who can upload a Minute to the system. Once he has upload this Minute the other users of the system and assistants of the meeting will be capable of sign this Minute.

When all the the assistants of the meeting have sign the Minute then it will be stored in the DB, enciphering it using AES with a key of size 128 bits that was previously given to the CEO.