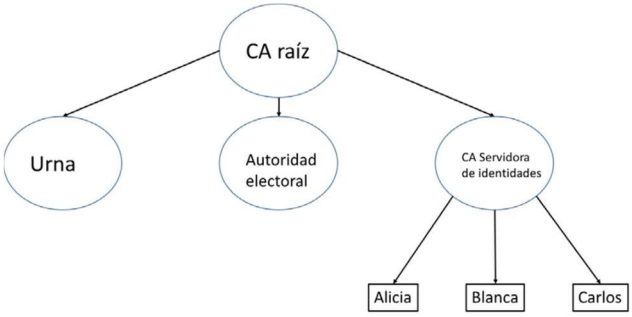
PRÁCTICA 4. CIFRADO Y DESCIFRADO RSA

Para la realización de la práctica 4 se debe tener en cuenta el diagrama de voto de la práctica 3 que se muestra debajo. En este caso, se ha realizado la práctica suponiendo que Alicia envía su voto a la urna, por lo tanto, necesitaremos tanto el certificado como la clave privada de la urna generada en la práctica anterior.



Las funciones de cifrado y descifrado RSA se pueden realizar de dos maneras, o bien obteniendo tanto los números públicos como privados de las claves para cada una de las acciones o bien mediante las funciones encrypt y decrypt con los parámetros de entrada necesarios. En el caso de la utilización de los números públicos (e,n) y privados (d,p,q) el procedimiento sería:

CIFRADO →((clave en int)^e)mod n

DESCIFRADO → ((cifrado en int)^d(mod(p*q)

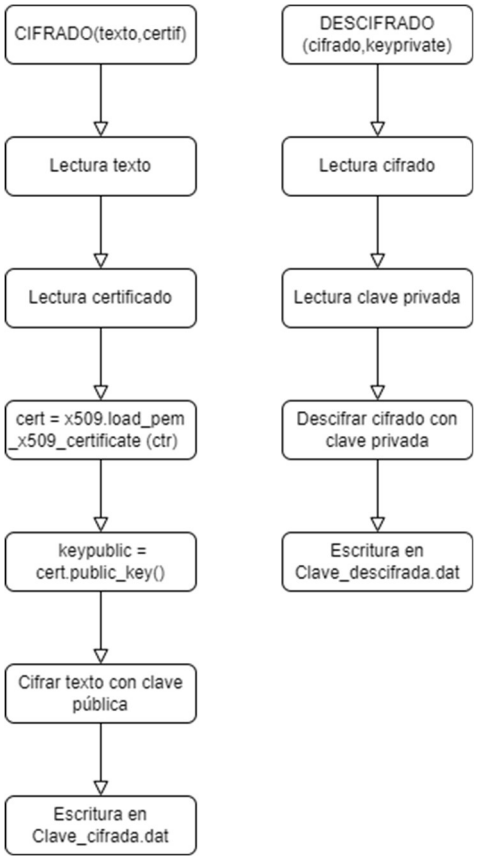
Y en el caso de las funciones encrypt y decrypt el procedimiento sería:

```
ciphertext = keypublic.encrypt(
        clavesesion,
        padding.OAEP(
                mgf=padding.MGF1(
                        algorithm=hashes.SHA256()),
                algorithm=hashes.SHA256(),
                label=None
        )
)
```

```
plaintext =  private_key.decrypt(
        clavecif,
        padding.OAEP(
                mgf=padding.MGF1 (
                        algorithm=hashes.SHA256()),
                algorithm=hashes.SHA256(),
                label=None
        )
)
```

En mi caso, el método utilizado ha sido el segundo dado que el primero conllevaba mucho tiempo de ejecución.

Para la definición de las funciones de cifrado y descifrado he seguido los siguientes diagramas:



Es preciso mencionar que para que las funciones de cifrado y descifrado tanto con clave pública como privada los ficheros se leen y escriben mediante “rb” y “wb”, es decir, en bytes.

Teniendo todo esto en cuenta el programa principal quedará de la siguiente manera:

```
def main():
    key = b"HOLA"
    with open("Clave_AES.dat", "wb") as key_file:
        key_file.write(key)
    key_file.close()

    print(cifrado("Clave_AES.dat","Urna.crt"))
    print(descifrado("Clave_cifrada.dat","Urna.pem"))

if __name__ == '__main__':
        main()
```

Por lo tanto, para ejecutar el programa únicamente son necesarios los ficheros “Clave_AES.dat”, “Urna.crt” y “Urna.pem” en el mismo directorio que el fichero fuente y una vez ejecutado se crearán los ficheros “Clave_cifrada.dat” y “Clave_descifrada.dat”. Cabe mencionar que el fichero original de la práctica contiene la clave aleatoria generada en la práctica 2, es decir, la clave para el cifrado AES.