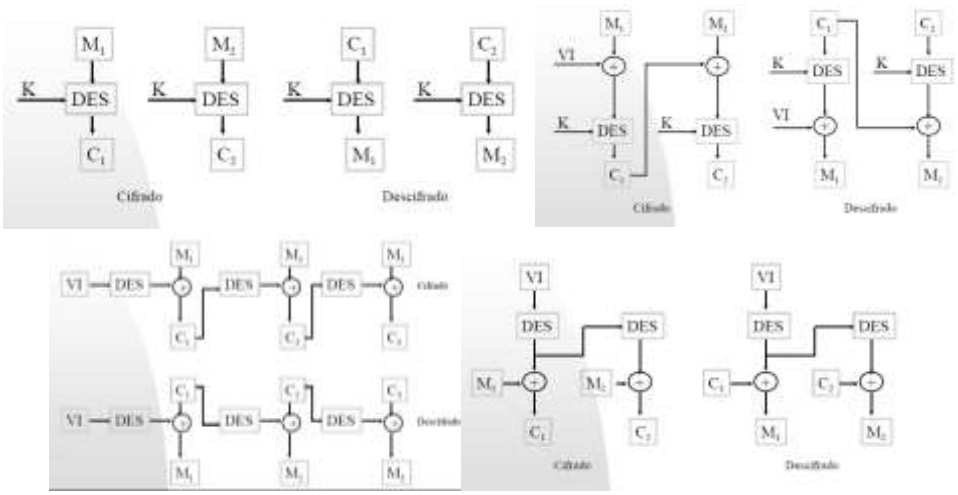
PRÁCTICA 2. CIFRADO Y DESCIFRADO AES Y FUNCIONES HASH

Para la realización de la práctica 2 se han tenido que tomar dos decisiones, una de ellas el método de cifrado y descifrado AES y la función HASH a utilizar.

En cuanto al cifrado y descifrado AES, se han tenido en cuenta los siguientes métodos, con sus respectivas ventajas y desventajas:

METODO	VENTAJA	DESVENTAJA
ECB	Cálculo sencillo en paralelo y sin transmisión de errores	El texto plano puede ser atacado
CBC	No es fácil atacar activamente y la seguridad es mejor que ECB	Sin paralelo, transmisión de errores, necesita vector de inicialización
CFB	Evita ataques por comienzos y finales iguales y por el reenvío de bloques	Sin paralelo en cifrado, transmisión de errores
OFB	Evita ataques por comienzos y finales iguales y por el reenvío de bloques.	Sin paralelo en cifrado y descifrado y transmisión de errores

Además de la tabla analizamos también sus diagramas para tomar la elección más conveniente.



En primer lugar se descarta el modo ECB, ya que el cálculo en paralelo de cada cifrado ofrece muy poca seguridad.

En cuanto al resto, analizando las ventajas y desventajas de la tabla y los diagramas de cada uno, se ha elegido el modo CBC, ya que como sus ventajas indican no es fácil de atacar y en el caso de una votación es un requisito indispensable. Además, sabiendo que su código es el siguiente:

```
Alg E_K(M)                          | Alg D_K(C)
C[0] <-$ {0,1}^n                    | for i = 1,...,m do
for i = 1,...,m do                  |     M[i] <- E_K^{-1}(C[i]) ⊕ C[i − 1]
    C[i] <- E_K(M[i] ⊕ C[i − 1])    | return M
return C
```

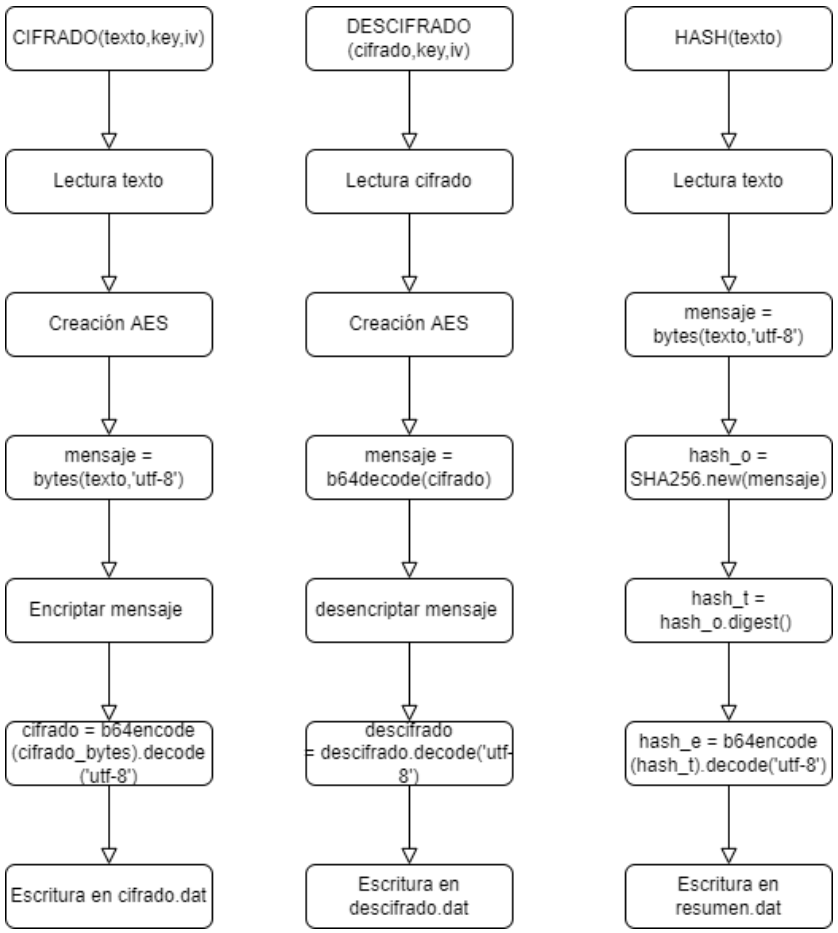
Si ciframos dos mensajes y un adversario conoce uno de ellos, no podrá determinar el restante. Por lo tanto, esto proporciona cierta seguridad, ya que aunque un adversario conozca uno de los votos, no conocerá el resto.

En cuanto a la elección del HASH, se han valorado las siguientes posibilidades, analizando como en el caso anterior sus ventajas y desventajas.

METODO	VENTAJA	DESVENTAJA
MD5	Fácil generar un resumen de mensaje del mensaje original	Propenso a la debilidad de colisión de hash, lento
SHA1	Más seguro que MD5	Difícil desencriptar, vulnerabilidad ante ataques
SHA256	Más pequeño, menos ancho de banda para almacenar y transmitir	Menos seguro que SHA512, SHA384

Teniendo en cuenta las ventajas y desventajas de las tres propuestas me he decido por usar SHA256 ya que es uno de los HASH más usados, dado que tiene un perfecto equilibrio entre seguridad y coste computacional y además tiene gran eficiencia. Además este algoritmo hash tiene la particularidad de que la longitud resultante es siempre igual dando igual el contenido utilizado para generar el hash.

A continuación se deben definir tanto la función de cifrado como la de descifrado como la del HASH. En mi caso para definir dichas funciones he seguido los siguientes diagramas:



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

```
def main():
    key = get_random_bytes(16)
    iv = get_random_bytes(16)
    print(cifrado("voto.dat",key,iv))
    print(descifrado("cifrado.dat",key,iv))
    print(hash_f("voto.dat"))

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

Por lo tanto, para ejecutar el programa únicamente es necesario un fichero “voto.dat” en el mismo directorio que el fichero fuente y una vez ejecutado se crearán los ficheros “cifrado.dat”, “descifrado.dat” y “resumen.dat”.