ESK_o - Ethereum wallet secret key of the file owner

EPK_o - Ethereum wallet public key of the file owner

SK_* - Asymmetric encryption secret key EVM compatible

PK_* - Asymmetric encryption public key EVM compatible

F - File to be uploaded and later shared

request_id - For the file owner to track the request

ID - To uniquely identify the credentials and uploaded file

PP - Public Parameters for the Proxy Re-encryption Scheme

PSK_o - Proxy Re-encryption Secret Key generated by DPCN and shared with the file owner

PPK_o - Proxy Re-encryption Public Key generated by DPCN and shared with the file owner (generated from PSK_o and PP)

RK_o2c - Re-encryption key to modify the result of an Enc2 operation from the owner to a client's credentials

AEnc(m, Asym_Key) - Asymetric encryption, resulting in cyphertext of m

SEnc(F, Key) - Symetric encryption resulting in cyphertext of file F obtained through encryption with a symetric Key

Enc1(Key, PPK_o) - First type of Proxy Re-encryption Encryption (which can not be re-encrypted and is the output of re-encryption)

Enc2(Key, PPK_o) - Second type of Proxy Re-encryption Encryption (which allows for re-encryption to be applied using an RK_o2c)

CID(m) - Content Identifier in IPFS networks that point to where a resource m can be found (in this case the symetrically encrypted file)

ReKey(PP, PKS_o, PPK_c) - Generate Re-encryption Key from owner to client

PubCheck(PP, RK_o2c, PPK_o, PPK_c) - Public verification the Re-encryption Key is valid from owner to client