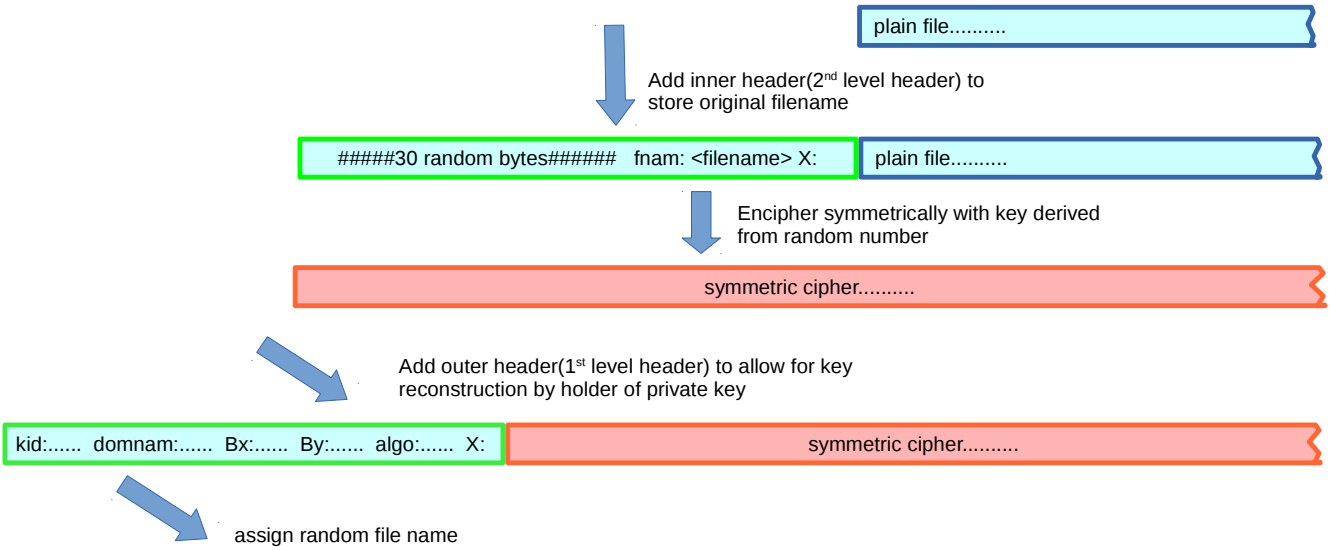
B.) Specifications of "Academic Signature – ECC-ciphers"

In contrast with the signature case, the ciphers do not adhere to any scheme standardized somewhere else. Highest priority in designing the structure of the ECC-cipher was fitness for purpose and security. Existing standards were disregarded.

Overview:

Prior to symmetric enciphering, the plainfile is prepended with an internal header file containing a reference to the original filename and head padding. Then the plainfile is enciphered symmetrically using a key derived from a random number "ke". The resulting cipher is then prepended with a plain, human readable header. This header allows to reconstruct the key to the holder of the corresponding private key.

Pattern for the Generation of an Elliptic Curve Cipher in Academic Signature



Example for first level plain header:

kid: anders_256_k1 domnam: p256r1

Bx: 8758eb791ad2b529f78c4fc4408244c45bea2c696804d62bb3098ed7c6bab0fb

By: 4f5a2ad632219595bb74d1eb6a62ae9df76590440498da8da8da538fcbdc7ab1

algo: F_cnt_1b X:

Description of first level plain header

The header contains five items which need to be given in the order shown in the example header. As in Academic Signatures ECDSA signature, items and identifiers are separated by one or more whitespaces as defined for the C procedures scanf or printf.

The first item is preceded by “kid: “ and is the ascii representation of the id of the public key used by the encipherer.

The second item is preceded by the string “domnam: “ and is the ascii representation of the identifier of the elliptic curve.

Third and fourth items are the coordinates of a point “B” on the elliptic curve. Note that By is redundant since it could be calculated from Bx up to the sign, so one bit would suffice. The redundancy is intended to allow for point/ellipse verification.

The characters “X:” are the tail marker of the header file, marking the subsequent start of the cipher file.

Usage of the information given in the first level plain header

B is the result of the point multiplication of the public key of the intended recipient with “ke”, whereas the key used for the symmetric cipher is the x- coordinate of the product of “ke” with the generator G of the group(low digit at low address in key block).

Heading zero bytes are discarded. Note that the Fleas algorithms operate with arbitrary key size. There will be no key padding with zeros to a defined key length. In the special case of using aes the first 16 bytes of a derivative(not in the mathematical sense) of the key are used for key whitening of the first block and the following 32 bytes of the key derivative are used as 256 bit aes-key.

The holder of the private key is able to invert the multiplication of his/her private key with ke (which lead to point B) and regain ke*G and thus its x coordinate which is the key used for the symmetric cipher.

Note that this pattern deviates from classical ECC elGamal in that the public key of the recipient has absolutely no influence on the key used for the symmetric cipher here.

The coordinates of B are given as plain hex numbers (high digits left).

The last item is preceded by “algo: “ and is an identifier of the symmetric algorithm used. In the above example, F_cnt_1b stands for 2 path Fleas in counter mode. Presently using the following algorithms for symmetric enciphering is possible in Academic Signature: aes, Fleas_3, Fleas_4, Fleas_5, Fleas_x2, Fleas_x5, Fleas_o2, Fleas_o5, Fleas_l, Fleas_ls, Fleas_l3, Fleas_lc, Fleas_ld, F_cnt_1b, F_cnt_1c, F_cnt_1d.

The legitimate receiver of the cipher can decipher the remaining part of the ciphertext using the key regained from this header.

The deciphered file exposes the encapsuled 2nd level header containing a random number padding and the original filename.

The 2nd level header is read, then cut after the tail marker “X:” and discarded. In a last step the resulting plaintext is renamed to the original filename and all intermediate file fragments are deleted.

Structure and usage of the 2nd level encapsulated header

The first 30 bytes are random numbers and are to be discarded. Upon setting these random numbers prior to enciphering, it is checked that accidental occurrence of the tail marker “X:” is avoided.

What follows is the string “ fnam: “ followed by the original filename in ascii representation. The filename will be truncated if longer than 400 bytes. Should the filename contain the tail marker sequence “X:”, this will be replaced by the sequence “X_” prior to enciphering. Filenames containing blanks will have the blanks replaced by underscores. Should any of these replacements occur, the deciphered plainfile will retain these name changes on the side of the recipient.