MysteryTwister C3
THE CRYPTO CHALLENGE CONTEST

Kaskade-S/T - Part 1

Author: Bernhard Esslinger

February 2011

Kaskade

One evening an old friend gives you a call. As he knows, that you are interested in ciphers, he asks you to “test” a cipher designed by him. For this purpose he sends you the implementation of the cipher in Python, a plaintext and the corresponding ciphertext.

The cipher, designed by your friend, is a so called Substitution-/Transposition cipher (S/T cipher). Of course, you know that modern ciphers like AES and 3DES also use substitutions and transpositions and are much safer.

Kaskade

The given cipher only consists of one (simple) substitution and one (simple) transposition, and it processes on bytes, not on bits. As first challenge your friend made it pretty easy for you: For the substitution and the transposition the two parts of the key are identical (each with a length of 26 byte).
The complete key has a length of 2*26 byte. You find a plaintext and a corresponding ciphertext within the zip archive of this challenge. So you have everything you need to perfom a known-plaintext attack. The text only consists of the 26 capital letters.

Challenge

The cipher processes the plaintext and the ciphertext in blocks of the length of one key part (26 byte) and pads the last block, if necessary, with X.

The solution of this challenge is the key, which has been used to encrypt the given plaintext (you have to give it in hex format without delimiter, also without the commonly used “0x” prefix in front of hex values, for example 120a...). Please notice that the solution consists of both key parts, even if they are identical.

Challenge

If you write the key you found into the file “26key”, you are able to test with the given Python v3.1 program (Kaskade.py), whether the given plaintext can be encrypted correctly into the given ciphertext.

The call for encryption is:

```
python3.1 Kaskade.py -e plaintext ciphertext 26key -l 26 -f A
```

The call for decryption is:

```
python3.1 Kaskade.py -d ciphertext plaintext 26key -l 26 -f A
```