Documentație

# Programul ”Criptare & Decriptare”:

Programul "Criptare & Decriptare" utilizează o serie de funcții pentru a cripta si decripta o imagine bitmap.

Programul este împărțit in trei fișiere, două sursă (main.c si cod.c) și un header (cod.h). "main.c" conține funcția main, necesară pentru rularea programului, în care apelăm funcția "criptare" și "decriptare". "cod.c" conține definițiile funcțiilor necesare algoritmului de criptare și decriptare. "cod.h" conține predefinițiile pentru funcțiile cu care lucrează programul nostru.

# Criptarea :

În partea de criptare, programul stochează în memorie o imagine bitmap în formă liniarizată (pixelii sunt reținuți într-un vector alocat dinamic), ținând cont de octeții de padding pe care nu îi ia în considerare la liniarizare. Se creează apoi un șir de numere pseudo – aleatoare folosind generatorul XorShift 32 si o cheie secreta R0, pe care le vom folosi în obținerea unei permutări. Se aplică permutarea asupra pixelilor din imaginea inițială și rezultatul este reținut prin intermediul unui vector alocat dinamic. Asupra pixelilor permutați se aplică o relație de substituție, utilizând o altă cheie secretă SV, care ne va da ca rezultat pixelii imaginii criptate.

**Funcțiile utilizate:**

* *criptare*:
* funcția de bază pentru criptarea de imaginii, în care se apelează funcțiile și se inițializează variabilele necesare algoritmului
* *calc\_W\_H:*
  + functia calculeaza dimensiunile pentru o imagine trimisă ca parametru
* *calc\_padding:*
* calculează octeții de padding de pe o linie dintr-o imagine
* *liniarizare:*
  + returnează un vector care conține pixelii dintr-o imagine (imagine liniarizată)
* *citire\_chei:*
  + citește cheile necesare algoritmului de criptare, dintr-un fișier text
* *XorShift32:*
  + Generator de numere pseudo-aleatoare
* *create\_rand\_sequence:*
  + creează un șir de numere aleatoare folosindu-se de funcția *XorShift32* și de cheia secretă R0
* *create\_array:*
  + returnează un vector în care fiecare element este inițializat cu poziția pe care se află în vector (necesar pentru permutare)
* *Durstenfeld:*
  + Creează permutarea aleatoare
* *Xor0 & XorK:*
  + Calculează valorile pentru pixelii criptați ( Xor0 folosește cheia secretă SV)
* *Substituție:*
  + Relația de substituție care se aplică pentru fiecare pixel permutat din imagine
* *create\_new\_img:*
  + creează imaginea criptată cu pixelii rezultați dupa substituție

# Decriptarea:

Partea de decriptare a programului reprezintă inversa părții de criptare și utilizează majoritatea funcțiilor folosite în algoritmul de criptare. Imaginea criptată la pasul anterior este liniarizată într-un vector alocat dinamic. Sunt reconstruite secvența de numere aleatoare, cu ajutorul cheii R0, și permutarea folosită la criptare. Se aplică inversa procesului de substituție asupra pixelilor din imaginea criptată, folosind cheia secretă SV. Pixelii rezultați sunt permutați invers ( folosind inversa permutării inițiale) și se obține imaginea originală.

**Funcțiile utilizate:**

* *inv\_substituție:*
  + reprezintă inversa procesului de substituție utilizat în algoritmul de criptare
* *inv\_permutare:*
  + inversa permutării utilizate în algoritmul de criptare

(Restul de funcții le-am explicat mai sus).