PROIECT – CRIPTOGRAFIE ȘI SECURITATEA DATELOR

Sistem local de management al cheilor de criptare pentru mai mulți algoritmi

Asofronie Rareș-Flavian

Hordilă Răzvan-Adrian

Hurghiș Gheorghe-Georgian

I) Introducere

În era digitală actuală, securitatea informației a devenit o componentă esențială a oricărui sistem informatic. Fie că este vorba de protejarea fișierelor personale, a datelor organizaționale sau a comunicațiilor confidențiale, criptarea joacă un rol central în asigurarea confidențialității și integrității datelor. Acest proiect propune realizarea unui sistem local de gestionare a criptării fișierelor, folosind mai mulți algoritmi de criptare (atât simetrici cât și asimetrici), într-o aplicație desktop cu interfață grafică (GUI). Scopul principal este criptarea, decriptarea și managementul cheilor într-un mod organizat și sigur, cu salvarea metadatelor în baza de date locală SQLite.

II) Obiective

Ne propunem să realizăm un manager de fișiere criptate care:

- Permite adăugarea de fișiere
- Înregistrează toate operațiile și măsoară timpul de execuție
- Suportă criptarea și decriptarea cu OpenSSL și GnuPG
- Salvează datele relevante într-o bază de date locală SQLite
- Se propune gestionarea cheilor pentru algoritmi simetrici și asimetrici

III) Explicații privind implementarea proiectului

Aplicația este dezvoltată în Python folosind:

- Tkinter - folosită pentru interfața grafică (GUI)
- SQLite – utilizat pentru gestionarea datelor
- OpenSSL și GnuPG – rolul lor este de a realiza operațiile de criptare și chei

Algoritmii aleși pentru acest proiect sunt:

★ Algoritmi simetrici:
- AES - OpenSSL
- DES - OpenSSl și GnuPG
- ChaCha20 - OpenSSL
- AES256 - GPG

★ Algoritmi asimetrici
- RSA – OpenSSL
- RSA - GnuPG

Cod semnificativ

→ Generarea cheilor RSA cu OpenSSL

```
203         subprocess.run( args: ["openssl", "genrsa", "-out", priv_path, "2048"], check=True)
204         subprocess.run( args: ["openssl", "rsa", "-in", priv_path, "-pubout", "-out", pub_path], check=True)
```

→ Utilizarea algoritmilor de criptare din funcția cripteaza_fisier

```
356     try:
357         match algoritm_id:
358             case 1:
359                 subprocess.run( args: [
360                     "openssl", "enc", "-aes-256-cbc", "-pbkdf2",
361                     "-in", cale_input, "-out", cale_output, "-pass", f"pass:{cheie}"
362                 ], check=True)
363             case 2:
364                 subprocess.run( args: [
365                     "openssl", "enc", "-des-ede-cbc", "-pbkdf2",
366                     "-in", cale_input, "-out", cale_output, "-pass", f"pass:{cheie}"
367                 ], check=True)
368             case 3:
369                 subprocess.run( args: [
370                     "openssl", "enc", "-chacha20", "-pbkdf2",
371                     "-in", cale_input, "-out", cale_output, "-pass", f"pass:{cheie}"
372                 ], check=True)
373             case 4:
374                 with open("public.pem", "w") as file:
375                     file.write(cheie)
376                 subprocess.run( args: [
```

```
def cripteaza_fisier():  1 usage
                    "openssl", "pkeyutl", "-encrypt",
                    "-in", cale_input, "-pubin", "-inkey", "public.pem",
                    "-out", cale_output
                ], check=True)

            case 6:
                subprocess.run([
                    "gpg", "--symmetric", "--cipher-algo", "AES256",
                    "--batch", "--yes", "--pinentry-mode", "loopback", "--passphrase", cheie,
                    "--output", cale_output, cale_input
                ])

            case 7:
                subprocess.run([
                    "gpg", "--symmetric", "--cipher-algo", "3DES","--allow-old-cipher-algos",
                    "--batch", "--yes", "--pinentry-mode", "loopback",
                    "--passphrase", cheie, "--output", cale_output, cale_input
                ])
            case 8:
                with open("public.asc", "w") as file:
                    file.write(cheie)

                subprocess.run( args: ["gpg", "--import", "public.asc"], check=True)

                subprocess.run( args: [
                    "gpg", "--encrypt", "--recipient", "razvan-adrian.hordila@student.tuiasi.ro",
                    "--output", cale_output, cale_input
                ], check=True)
```

→ Înregistrarea operațiilor

```
        cursor.execute( sql: """
            INSERT INTO Operatii (fisier_id, algoritm_id, cheie_id, tip_operatie, durata, rezultat)
            VALUES (?, ?, ?, 'criptare', ?, 1)
        """,  parameters: (fisier_id, algoritm_id, cheie_id, durata))

```

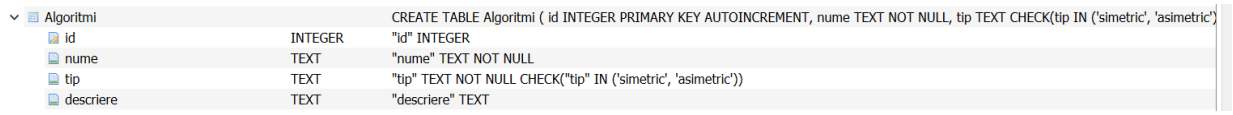
→ Afișarea logului în GUI

```
    for op in operatii:
        status = "SUCCES" if op[5] == 1 else "ESEC"
        text_loguri.insert(tk.END,
                           chars: f"[#{op[0]}] {op[3].upper()} | {op[1]} | Alg: {op[2]} | {op[4]} ms | {status} | Msg: {op[6]}\n"
                           )

```

IV) Baza de date

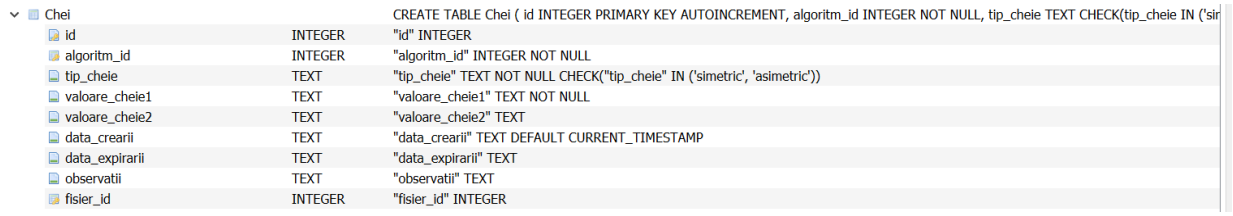
Aplicația utilizează o bază de date SQLite numită csd.db, care conține tabelele afișate in pozele de mai jos:



Algoritmi		CREATE TABLE Algoritmi (id INTEGER PRIMARY KEY AUTOINCREMENT, nume TEXT NOT NULL, tip TEXT CHECK(tip IN ('simetric', 'asimetric')
id	INTEGER	"id" INTEGER
nume	TEXT	"nume" TEXT NOT NULL
tip	TEXT	"tip" TEXT NOT NULL CHECK("tip" IN ('simetric', 'asimetric'))
descriere	TEXT	"descriere" TEXT

Algoritmi

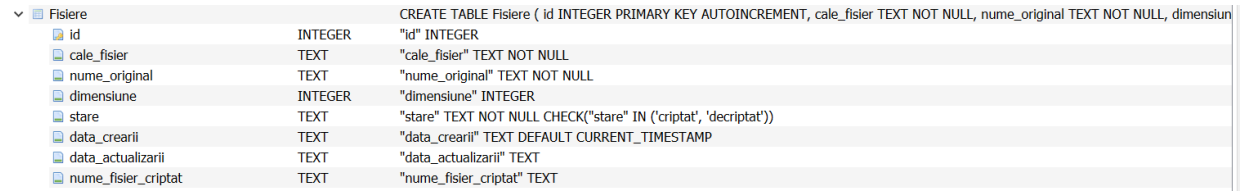
- id: identificator
- nume: ex: AES, RSA
- tip: simetric / asimetric
- descriere: scurtă explicație



Chei		CREATE TABLE Chei (id INTEGER PRIMARY KEY AUTOINCREMENT, algoritm_id INTEGER NOT NULL, tip_cheie TEXT CHECK(tip_cheie IN ('sir
id	INTEGER	"id" INTEGER
algoritm_id	INTEGER	"algoritm_id" INTEGER NOT NULL
tip_cheie	TEXT	"tip_cheie" TEXT NOT NULL CHECK("tip_cheie" IN ('simetric', 'asimetric'))
valoare_cheie1	TEXT	"valoare_cheie1" TEXT NOT NULL
valoare_cheie2	TEXT	"valoare_cheie2" TEXT
data_crearii	TEXT	"data_crearii" TEXT DEFAULT CURRENT_TIMESTAMP
data_expirarii	TEXT	"data_expirarii" TEXT
observatii	TEXT	"observatii" TEXT
fisier_id	INTEGER	"fisier_id" INTEGER

Chei

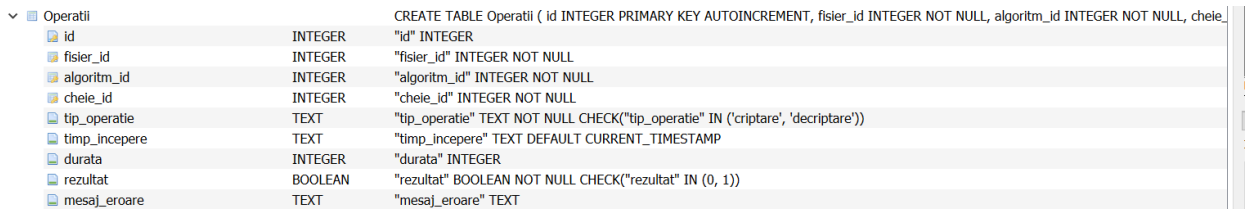
- id: INTEGER – cheie primară
- algoritm_id: INTEGER – legătură către Algoritmi
- fisier_id: INTEGER – legătură către Fisiere
- tip_cheie: TEXT – simetric / asimetric
- valoare_cheie1: TEXT – cheie publică sau parolă
- valoare_cheie2: TEXT – cheie privată (dacă există)
- data_crearii: TEXT – timestamp implicit
- data_expirarii: TEXT – rezervat pentru viitor
- observatii: TEXT – detalii (ex: „generată cu OpenSSL”)



Fisiere		CREATE TABLE Fisiere (id INTEGER PRIMARY KEY AUTOINCREMENT, cale_fisier TEXT NOT NULL, nume_original TEXT NOT NULL, dimensiun
id	INTEGER	"id" INTEGER
cale_fisier	TEXT	"cale_fisier" TEXT NOT NULL
nume_original	TEXT	"nume_original" TEXT NOT NULL
dimensiune	INTEGER	"dimensiune" INTEGER
stare	TEXT	"stare" TEXT NOT NULL CHECK("stare" IN ('criptat', 'decriptat'))
data_crearii	TEXT	"data_crearii" TEXT DEFAULT CURRENT_TIMESTAMP
data_actualizarii	TEXT	"data_actualizarii" TEXT
nume_fisier_criptat	TEXT	"nume_fisier_criptat" TEXT

Fisiere

- id: INTEGER – cheie primară
- cale_fisier: TEXT – locația pe disc
- nume_original: TEXT – nume fișier original
- dimensiune: INTEGER – mărimea în octeți
- stare: TEXT – criptat / decriptat
- data_crearii: TEXT – timestamp
- data_actualizarii: TEXT – ultimă modificare
- nume_fisier_criptat: TEXT – cale fișier criptat (dacă există)



Operatii		CREATE TABLE Operatii (id INTEGER PRIMARY KEY AUTOINCREMENT, fisier_id INTEGER NOT NULL, algoritm_id INTEGER NOT NULL, cheie_
id	INTEGER	"id" INTEGER
fisier_id	INTEGER	"fisier_id" INTEGER NOT NULL
algoritm_id	INTEGER	"algoritm_id" INTEGER NOT NULL
cheie_id	INTEGER	"cheie_id" INTEGER NOT NULL
tip_operatie	TEXT	"tip_operatie" TEXT NOT NULL CHECK("tip_operatie" IN ('criptare', 'decriptare'))
timp_incepere	TEXT	"timp_incepere" TEXT DEFAULT CURRENT_TIMESTAMP
durata	INTEGER	"durata" INTEGER
rezultat	BOOLEAN	"rezultat" BOOLEAN NOT NULL CHECK("rezultat" IN (0, 1))
mesaj_eroare	TEXT	"mesaj_eroare" TEXT

Operatii

- id: INTEGER – cheie primară
- fisier_id, algoritm_id, cheie_id: INTEGER – referințe
- tip_operatie: TEXT – criptare / decriptare
- timp_incepere: TEXT – data/timpul începerii
- durata: INTEGER – milisecunde
- rezultat: BOOLEAN – 1 = succes, 0 = eșec
- mesaj_eroare: TEXT – mesaj dacă a eșuat

V) Dificultăți întâmpinate

- Erori criptografice: un exemplu concret îl reprezintă criptarea RSA ce eșuează dacă cheia publică nu este corect salvată temporar.
- Gestionarea fișierelor temporare : Pentru RSA și GPG, cheile se scriu temporar în fișiere .pem sau .asc, iar apoi sunt importate de openssl sau gpg.
- Incompabilitatea dintre framework-uri: Cheile generate cu GPG nu pot fi folosite în OpenSSL și invers, fiecare framework are formate proprii.
- Sincronizarea în GUI : Funcțiile update_dropdown , update_algoritmi_dropdown și update_chei_dropdown au sincronizat dropdown-urile pentru algoritmi, chei și fișiere.

VI) Concluzie

Acest proiect este complet funcțional, fiind capabil să demonstreze o serie de servicii extrem de complexe, cum ar fi:

- Implementarea unui manager de criptare
- Asigurarea unui suport pentru algoritmi simetrici și asimetrici
- Realizarea unei interfețe intuitive cu SQLite
- Integrarea cu succes a 2 framework-uri criptografice: OpenSSL și GnuPG(GNU Privacy Guard)