CryptoPerf

Criptare rapidă și analiză eficientă a performanței

Realizat de:

Dancău Rareș-Andrei, 1410A

Cojocaru Rareș, 1410A

# Cuprins

[**Cuprins 1**](#_821b6axyhl1e)

[**Introducere 1**](#_p6tmoam3jle8)

[**Baza de date 2**](#_cb39g8am7iro)

[**Implementarea proiectului 3**](#_ccgxef3vhauf)

[**Analiza performanțelor 5**](#_l1us7t2x3sl)

[**Concluzii 6**](#_lwjliwa8cg6i)

# 

# Introducere

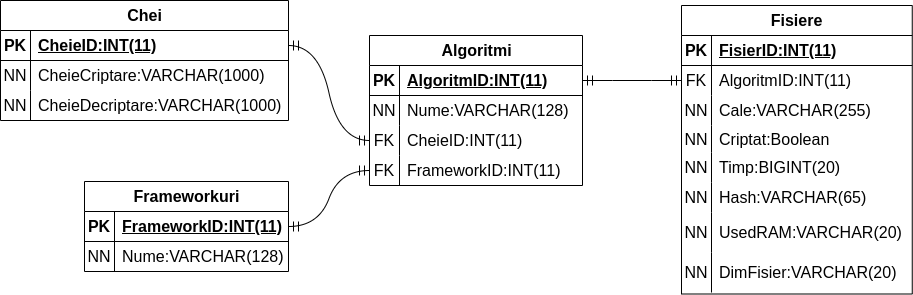
CryptoPerf este un proiect ce vizează managementul de fișiere și perechi de chei ai algoritmilor de criptare, dar are ca scop și evaluarea performanței a 3 framework-uri ce suportă operații de criptare și decriptare a fișierelor: OpenSSL, Ccrypt și Mcrypt.

Scopul principal al proiectului CryptoPerf este de a permite criptarea și decriptarea fișierelor folosind perechi de chei definite de utilizator, dar mai are și rolul de a furniza un instrument de evaluare a performanței framework-urilor de criptare, în vederea facilitării alegerii celui mai adecvat pentru o anumită aplicație sau scenariu de utilizare. Acest instrument permite utilizatorilor să compare timpul de criptare și decriptare, consumul de memorie și alte metrici relevante pentru diferitele framework-uri disponibile.

Programul suportă atât algoritmi de criptare cu chei simetrice, cât și cu chei asimetrice, oferind utilizatorilor flexibilitate în alegerea și evaluarea framework-urilor în funcție de nevoile și cerințele lor specifice.

# Baza de date

Proiectul CryptoPerf folosește o bază de date MySQL a cărei structură poate fi vizualizată mai jos:



Baza de date este formată din următoarele tabele:

* Tabelul Chei care stochează informații despre cheile folosite în procesul de criptare/decriptare.
* Tabelul Frameworkuri conține informații despre framework-ul de criptare.
* Tabelul Algoritmi conține informații despre algoritmii de criptare/decriptare utilizați.
* Tabelul FIșiere conține informații despre fișierele gestionate de proiect.

În continuare, vor fi descrise pe larg cele cinci tabele.

Tabelul Chei este format din:

* CheieID care reprezintă un identificator unic pentru perechea de chei.
* CheieCriptare conține cheia care va fi folosită în procesul de criptare.
* CheieDecriptare conține cheia folosită în procesul de decriptare. Pentru algoritmii simetrici, are aceeași valoare cu cea din câmpul CheieCriptare.

Tabelul Frameworkuri este alcătuit din:

* FrameworkID care reprezintă un identificator unic pentru fiecare framework.
* Nume care stochează numele framework-ului de criptare/decriptare.

Tabelul Algoritmi este format din:

* AlgoritmID care reprezintă un identificator unic pentru un algoritm de criptare. Algoritmul de criptare este definit în acest caz de un nume, o pereche de chei și un framework.
* Nume stochează numele algoritmului de criptare/decriptare.
* CheieID stochează o legătură la tabela Chei care descrie identificatorul perechii de chei folosite în cadrul algoritmului.
* FrameworkID stochează o legătură către tabela Frameworkuri care descrie identificatorul framework-ului din care face parte algoritmul curent.

Tabelul Fisiere este alcătuit din:

* FisierID care reprezintă un identificator unic pentru fișier care reprezintă o cale și un algoritm de criptare(definit anterior ca fiind format dintr-un nume, un framework și o pereche de chei).
* Cale stochează calea la care este stocat fișierul curent.
* Criptat este o valoare booleană care este adevărat dacă fișierul curent este criptat și fals dacă fișierul curent este decriptat.
* Timp stochează în milisecunde timpul de criptare/decriptare.
* Hash stochează valoarea hash-ului MD5 a fișierului curent. Acesta este calculat înainte de criptarea fișierului și rămâne constant în baza de date.
* UsedRAM stochează în MB cantitatea de memorie RAM utilizată în timpul criptării/decriptării.
* DimFisier stochează în kB dimensiunea fișierului. Este calculată înainte de criptare/decriptare și este folosită împreună cu Timp și UsedRAM în cadrul evaluării performanțelor framework-urilor.

# Implementarea proiectului

Proiectul a fost implementat folosind Python ca limbaj principal de programare și o serie de tehnologii și framework-uri suplimentare pentru diferite aspecte ale aplicației.

Pentru stocarea datelor, am folosit o bază de date MySQL, a cărei structură a fost descrisă anterior, iar interacțiunea cu aceasta este realizată folosind peewee, un ORM (Object Relational Mapper) minimalist și ușor de utilizat.

Dezvoltarea interfeței grafice a fost realizată folosind framework-ul PyQt împreună cu Qt Designer oferind astfel o interfață modernă și ușor de utilizat. În plus, am integrat funcționalități de generare și afișare a graficelor direct în interfață, folosind biblioteca Matplotlib, pentru a oferi utilizatorilor o experiență completă și informativă.

Proiectul CryptoPerf utilizează trei framework-uri diferite pentru criptarea fișierelor: OpenSSL, Mcrypt și Ccrypt. Criptarea fișierelor este realizată prin lansarea comenzilor corespunzătoare în linia de comandă, folosind librăria subprocess.

[**OpenSSL**](https://www.openssl.org/docs/) reprezintă un set de instrumente de criptografie care implementează protocoalele Secure Sockets Layer (SSL v2/v3) și Transport Layer Security (TLS v1), împreună cu standardele de criptografie asociate acestora. A fost lansat în anul 1998 și este disponibil pentru sistemele de operare: Linux, Windows și MacOS. OpenSSL include un set extins de instrumente de linie de comandă care permit utilizatorilor să efectueze o varietate de operațiuni criptografice direct din terminal, cum ar fi generarea de chei, criptarea și decriptarea fișierelor, semnături digitale și multe altele. Algoritmi suportați: AES128, AES192, AES256.

[**Ccrypt**](https://ccrypt.sourceforge.net/) este o unealtă pentru criptarea și decriptarea fișierelor și fluxurilor de date. A fost concepută pentru a înlocui utilitarul standard unix crypt, care este cunoscut pentru folosirea unui algoritm de criptare foarte slab. ccrypt se bazează pe cifrul bloc Rijndael, care a fost, de asemenea, ales de guvernul american ca Standardul Avansat de Criptare. Spre deosebire de unix crypt, trebuie să se specifice dacă se dorește criptarea sau decriptarea. Cea mai comună modalitate de a folosi ccrypt este prin comenzile ccencrypt și ccdecrypt. Algoritmi suportați: AES256.

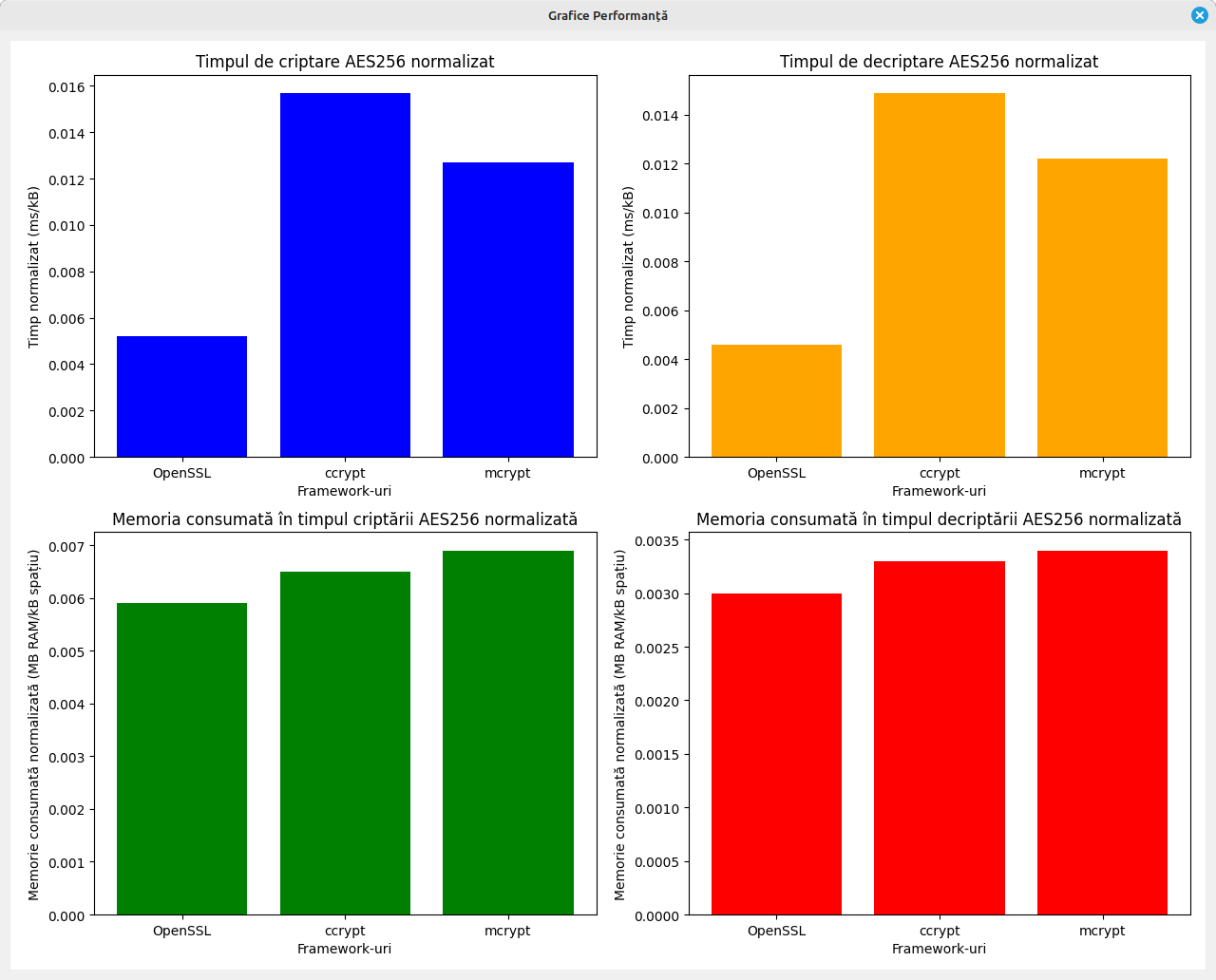
[**Mcrypt**](https://mcrypt.sourceforge.net/), similar cu Ccrypt, reprezintă un înlocuitor pentru vechiul crypt din Unix. Atunci când se criptează sau decriptează un fișier, este creat un fișier nou cu extensia .nc și modul 0600. Fișierul nou păstrează data de modificare a originalului. Fișierul original poate fi șters specificând parametrul -u. Dacă nu sunt specificate fișiere, intrarea standard este criptată în ieșirea standard. Mcrypt folosește toți algoritmii simetrici incluși în libmcrypt. Algoritmi suportați: AES256, CAST128, CAST256.

Pentru o funcționare optimă a programului, se recomandă utilizarea sistemului de operare Linux. Deoarece proiectul utilizează comenzi specifice terminalului pentru criptare și decriptare, Linux oferă un mediu mai stabil și mai potrivit pentru execuția acestor comenzi.

# Analiza performanțelor

În cadrul proiectului, analiza diferitelor framework-uri s-a făcut comparând timpi de criptare/decriptare normalizați în funcție de dimensiunea fișierelor, dar și după cantitatea de memorie RAM utilizată în cadrul procesului de criptare/decriptare, normalizată la dimensiunea fișierelor procesate.

Pentru a compara performanțele, s-au folosit aceleași fișiere între framework-uri și s-a făcut o medie între timpul de procesare și dimensiunea fișierelor pentru a afla timpul normalizat, iar pentru memoria RAM consumată normalizată s-a făcut o medie între memoria RAM consumată la procesarea fișierelor și dimensiunea fișierelor. Compararea s-a făcut pe algoritmul AES256 deoarece acesta este suportat de toate cele trei framework-uri. În urma acestei comparații, au rezultat graficele de mai jos:



| Framework | Timp criptare normalizat(ms/kB) | Timp decriptare normalizat(ms/kB) | Timp criptare+decriptare |
| --- | --- | --- | --- |
| OpenSSL AES256 | 0.0052 | 0.0046 | 0.0098 |
| Ccrypt AES256 | 0.0157 | 0.0149 | 0.0306 |
| Mcrypt AES256 | 0.0127 | 0.0122 | 0.0249 |

| Framework | RAM criptare normalizat(MB/kB) | RAM decriptare normalizat(MB/kB) |
| --- | --- | --- |
| OpenSSL AES256 | 0.0059 | 0.003 |
| Ccrypt AES256 | 0.0065 | 0.0033 |
| Mcrypt AES256 | 0.0069 | 0.0034 |

# Concluzii

Din rezultatele obținute, se constată că framework-ul OpenSSL prezintă cele mai bune rezultate în materie de timp de criptare și decriptare, urmat de mcrypt pe locul 2 și de ccrypt pe locul 3. Din punct de vedere al consumului de memorie, se remarcă performanțe similare între cele trei framework-uri.