Encrypted Database

Bianca Pricopi

Facultatea de Informatica, Universitatea "Alexandru Ioan Cuza", Iasi
biancapricopi1@gmail.com
https://www.info.uaic.ro/

Introducere

Obiectivul acestui raport este de a oferi informatii despre proiectul Encrypted Database. Proiectul consta intr-un tool care va permite criptarea unui fisier, citirea continutului sau a metadatelor asociate unui fisier existent in baza de date sau stergerea acestuia.

In mod general, fisierele criptate vor fi stocate intr-o anumita locatie pe disk, iar daca utilizatorul va dori sa stearga fisierul, atunci acesta va fi sters atat din baza de date cat si de pe disk.

Tehnologiile utilizate

Pentru baza de date criptata se va folosi un algoritm de criptare asimetrica, si anume RSA. Prin criptare asimetrica se intelege faptul ca vom avea 2 perechi de chei, o cheie publica si una privata. In mod evident, criptarea se realizeaza cu cheia publica, iar decriptarea cu cea privata. Se lucreaza cu numere mari, pe 1024 de biti. Se va folosi biblioteca Crypto doar pentru generarea unor numere prime pe 1024 de biti. Pentru fiecare fisier adaugat vor fi generate chei diferite.

In momentul in care un utilizator doreste adaugarea unui fisier in baza de date, prima data se va verifica daca fisierul exista sau nu, iar in caz afirmativ acesta va fi intrebat daca doreste sa il suprascrie. Pe langa adaugarea unui fisier, utilizatorul va avea posibilitatea sa citeasca toate informatiile asociate fisierului respectiv: nume, tip, locatie, dimensiune, atribute, owner uid, owner gid, drepturi, data crearii, data modificarii, data accesarii, metoda folosita pentru criptare (in acest caz RSA), cheia publica, cheia privata care va fi afisata sub forma unui hash (functia hash folosita va fi SHA3-512), locatia unde se afla fisierul criptat. De asemenea utilizatorul va putea sa si stearga un anumit fisier.

Algoritmul RSA va fi implementat de la zero. Pentru baza de date se va folosi o baza de date relationala, MySql.

Tool-ul nu va avea interfata grafica, va fi la nivel de terminal, dar se va utiliza o paleta de culori pentru a fi mai usor de urmarit. Erorile vor aparea cu rosu, modificarile realizate cu succes vor aparea cu verde, iar informatiile importante vor aparea cu galben.

Descrierea comenzilor

Comenzile disponibile:
enc -add < **file** > : (adauga fisierul dat ca parametru in baza de date)
enc -read-file < **file** > : (citeste continutul necriptat al fisierului dat ca parametru)
enc -read-meta < **file** > : (citeste metadatele fisierului dat ca parametru)
enc -read < **file** > : (citeste atat continutul necriptat al fisierului dat ca parametru, cat si metadatele acestuia)
enc -rm < **file** > : (sterge fisierul dat ca parametru)
q : (inchide tool-ul)
help : (afiseaza sintaxa si descrierea comenzilor disponibile)

Descrierea corner case-urilor

Prima data se va verifica daca programul a fost rulat cu 2 argumente care au urmatoarea semnificatie: primul argument reprezinta calea catre directorul in care se afla fisierele care urmeaza a fi introduse in baza de date, iar al doilea argument reprezinta calea catre directorul in care vor fi stocate fisierle criptate. Se va verifica existenta acestor cai, dar si faptul ca ele sunt directoare, in caz contrar, un mesaj de eroare va fi afisat, iar utilizatorul va proceda corespunzator.

De fiecare data va fi validat inputul introdus de utilizator. Un input valid consta in prefixul 'enc' urmat de comanda care se vrea a fi executata si de fisierul ce urmeaza a fi prelucrat corespunzator. Daca utilizatorul nu respecta constrangerile si conventiile, vor fi afisate mesaje de eroare descriptive care il vor ajuta sa inteleaga ce anume a gresit.

In cazul tuturor comenzilor se va verifica daca fisierul exista la locatia data in argument, in caz contrar va fi afisat un mesaj de eroare. Daca utilizatorul doreste sa introduca un fisier deja existent, va fi intrebat daca doreste sa il suprascrie.

In cazul in care incearca sa citeasca un fisier care nu exista in baza de date, un mesaj de eroare va fi afisat

In cazul in care vrea sa stearga un fisier care nu exista in baza de date, un mesaj de eroare va fi afisat.
In cazul in care modificarile au fost realizate cu success, va fi afisat un mesaj de succes.

Biblioteci folosite

1) **sys**: pentru a manipula argumentele de la linia de comanda
2) **os.path**: pentru a lucra cu fisiere si path-uri
3) **os.stat**: pentru a extrage metadatele asociate unui fisier
4) **dotenv**: pentru a avea acces la variabilele din fisierul .env
5) **mysql.connector**: pentru a lucra cu baza de date MySql
6) **colorama**: pentru a schimba culoarea textului in terminal
7) **random**: pentru a genera numere random

8) Crypto.Util: doar pentru a genera numere prime de lungime foarte mare (1024 biti)
9) hashlib: pentru a genera un hash pentru cheia privata. Aici folosesc SHA3-512.
10) datetime: pentru a converti secundele in datetime. Folosesc asta pentru metadatele asociate fisierului (date created, date modified, date accessed)

```
[Error]: Not enough arguments. Please enter one directory for Plain files and one directory for Encrypted Files in this order
```

```
>>:enc -no file
[Error]: -no not recognized as an internal command
Type "help" to list available commands
```

```
>>:eb f g
[Error]: You have a syntax error. "eb" is not recognized
Type "help" to list available commands
>>:
```

```
Type "help" to list available commands
>>:enc -add dummy.txt
[Error]: File not found. Please try again
>>:
```

```
>>:enc -rm secret.sql
File secret.sql successfully deleted from database
>>:enc -read secret.sql
[Error]: File not found. Please try again
>>:enc -reaad-content secret.sql
[Error]: -reaad-content not recognized as an internal command
Type "help" to list available commands
>>:enc -read-file secret.sql
[Error]: File not found. Please try again
>>:enc -read-meta secret.sql
[Error]: File not found. Please try again
>>:
```

```
>>:enc -add secret.sql
There is a file with the same name, do you want to override it? (y/n): y
File successfully encrypted.
>>:enc -rm secret.sql
File secret.sql successfully deleted from database
>>:enc -add secret.sql
File successfully encrypted.
>>:enc -read-file secret.sql
Content of the file:
CREATE TABLE SECRET_TABLE
>>:enc -read-meta secret.sql
Metadata:
Name: secret.sql
Type: .sql
File location: D:\facultate\III-Sem1\python\project\encrypted-database\PlainFiles\secret.sql
Size: 25.0
Attributes: 32
Owner uid: 0
Owner gid: 0
Mode: -rw-rw-rw-
Date created: 2022-11-28 01:04:58
Date modified: 2022-11-28 01:04:58
Date accessed: 2022-12-01 00:53:04
Method used for encryption: RSA
Encryption type: asymmetric
Public key modulus: 249965957131941876217802507805204028038133692920981194321169439587653657803357727103595976730593783384284512875632511185005700091790006735382616352014701831201415797790673316132734904808421573750566972296906461323142970466366773498140302298699681359465978351995670162995683365933477495419654624096613752200610477556821041653780891280702420769124554798382356966432339192108759868911774857486560564399876507793073352243879475870351889473135843333378628248174572767759403166025053985985985962263077629723282714845251718741117843626895910484089641370019483872147776761610058705057216393158611330973475462544324257033486693
Public key exponent: 12868206762550082051853007880381023927207192099680019372395854712670875640278218518478146383428817550414216099851939368308786871309386399654080540957120792761182434287220948823526718494203025384039506122147427287425124619534136520552970082716319179622439195560215442235015573423786998681630548324457523558059523980228123445941003649173521862705548101710095806166829921429628451920839038102834623823984433744504078959374011773788296391723506305782250890901814457551021219560698033894508342325048644648241091704116295127951327227740821769086297285365435771430502097927863670841757998742422241694641322095530023870174699 5
Private key: 5ac3591772281a8592124d5c9b74abe861f9e714aaa003e39c08e33177f615614a418219530ce91a292c27b75d1ba6a661cd1682ee5c7055e2408320946cfdf8
Encrypted file location: D:\facultate\III-Sem1\python\project\encrypted-database\EncryptedFiles\secret.sql
>>:
```

```
>>:help
Encrypted database (c)
enc -add <file> : add the file to encrypted database
enc -read-file <file> : display content of the encrypted file
enc -read-meta <file> : display metadata (properties) of the file
enc -read <file> : display content and metadata of the file
enc -rm <file> : delete the file from encrypted database
help : display a list of the available commands
q : to exit
>>:q
Bye
```