Ministerul Educaţiei și Cercetării

al Republicii Moldova   
Universitatea Tehnică a Moldovei

Faculatea Calculatoare, Infromatică și Microelectronică   
  
  
  
  
  
  
  
RAPORT

# LUCRARE DE LABORATOR NR. 4

# la Tehnologii ale securității informaționale

**Tema:** *Implementarea algoritmilor de criptare simetrică și asimetrică. Semnătura digitală.*

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: Octavian Răducanu

UTM, Chișinău 2023

**Tema:**

Implementarea algoritmilor de criptare simetrică și asimetrică. Semnătura digitală**.**

**Scopul lucrării:**

* **1. Algoritmul DES/AES**
  + Studierea algoritmului DES sau AES
  + Modelul matematic și funcționalul
  + Domeniile de aplicare
* **2. Algoritmul RSA**
  + Studierea modelului matematic al algoritmului RSA
  + Generarea cheilor
  + Domenii de aplicare
  + Semnătura digitală
* **3. Realizarea aplicației**
  + Realizarea unei aplicații într-un limbaj de programare la alegere
  + Interfața principală va reprezenta un meniu cu algoritmii DES/AES, RSA și semnătura digitală
  + Opțiunile la alegere sunt: Implementarea DES sau AES, la semnătura digitală fie RSA sau DSA
  + Limbaje de programare recomandate: C++, C#, Java, Python

**1. Algoritmul AES**

O schemă de criptare convențională este alcătuită din cinci elemente.

1. Textul clar – acesta este mesajul original sau infromațional de intrare pentru algoritmul de criptare.

2. Algoritmul de criptare – acest algoritm execută diferite substituții și transformări asupra textului clar.

3. Cheia secretă – această cheie este o intrare pentru algoritmul de criptare.

4. Textul cifrat – este textul rezultat din algoritmul de criptare. El depinde de textul clar și de cheia secretă. Pentru un mesaj dat, două chei secrete diferite produc două texte cifrate diferite.

5. Algoritmul de decriptare – acesta este algoritmul invers al algoritmului de criptare. Algoritmul de decriptare este aplicat textului cifrat și acelea și chei secrete pentru a obține textul clat original.

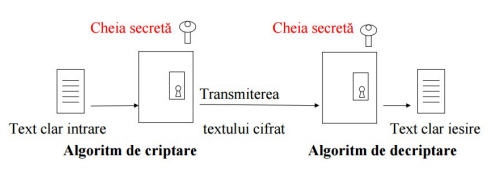


Figura 1. Modelul simplificat al criptării conveționale

*Studierea algoritmului AES*

**Algoritmul AES**

Algoritmul AES (Advanced Encryption Standard) este un algoritm de criptare simetrică, dezvoltat ca succesor al algoritmului DES. A fost adoptat ca standard de către NIST (National Institute of Standards and Technology) în anul 2001 și este utilizat pentru protejarea informațiilor sensibile într-o varietate de domenii, inclusiv finanțe, sănătate și militar.

*Modelul matematic și funcționalul*

Modelul matematic și funcțional al algoritmului AES se bazează pe un algoritm de substituție-permutare (SPN) care constă în aplicarea unui set de operații simple asupra blocurilor de date pentru a produce blocuri criptate. În cazul AES, blocurile de date au dimensiunea de 128 de biți și cheia de criptare are o dimensiune de 128, 192 sau 256 de biți.

Funcționarea algoritmului AES este următoarea:

* datele de intrare sunt împărțite în blocuri de 128 de biți;
* cheia de criptare este prelucrată pentru a genera o serie de sub-chei;
* blocurile de date sunt supuse unei serii de runde, fiecare rundă constând în patru operații: substituție, permutare, amestecare și adunare XOR cu o sub-cheie;
* la finalul rundelor, blocul de date criptat este generat.

Există trei variante ale AES: AES-128, AES-192 și AES-256, fiecare având o dimensiune diferită de cheie de criptare. AES-256 este considerat cel mai sigur dintre cele trei variante, deoarece folosește o cheie mai lungă și efectuează un număr mai mare de runde decât celelalte două variante.

*Domeniile de aplicare*

Domeniile de aplicare ale algoritmului AES includ protecția datelor în rețele de comunicații, securitatea informațiilor în sectorul financiar, sănătate și militar, precum și protejarea datelor personale în aplicațiile de securitate a informațiilor.

**2. Algoritmul RSA**

*Studierea algoritmului RSA*

**Algoritmul RSA (Rivest–Shamir–Adleman) este un algoritm de criptare asimetrică, care utilizează o pereche de chei pentru a cripta și decripta mesaje. A fost dezvoltat în anul 1977 de către Ron Rivest, Adi Shamir și Leonard Adleman și a devenit unul dintre cele mai utilizate algoritme de criptare în întreaga lume.**

*Modelul matematic*

**Modelul matematic al algoritmului RSA se bazează pe teoria numerelor și pe dificultatea factorizării numerelor mari. Mai precis, algoritmul RSA funcționează în baza faptului că este foarte ușor să se calculeze produsul a două numere prime mari, dar este foarte dificil să se afle numerele prime individuale din produs.**

*Generarea cheilor*

Generarea cheilor RSA implică trei pași:Se aleg două numere prime mari diferite, numite p și q.Se calculează produsul n = p \* q, care va fi utilizat ca modul în criptarea și decriptarea mesajelor.Se alege un număr întreg e astfel încât 1 < e < φ(n) și astfel încât e și φ(n) să fie prime între ele. Acesta va fi cheia publică.

Cheia privată este calculată folosind inversul modular al lui e, astfel încât d \* e ≡ 1 mod φ(n), unde φ(n) este funcția lui Euler pentru numărul n.

Pentru a cripta un mesaj M folosind algoritmul RSA, se utilizează cheia publică pentru a obține M^e mod n. Pentru a decripta mesajul criptat, se utilizează cheia privată pentru a calcula (M^e)^d mod n, ceea ce duce la obținerea mesajului original M.

*Domenii de aplicare*

Domeniile de aplicare ale algoritmului RSA includ securitatea informațiilor în rețele de comunicații, protejarea informațiilor sensibile în sectorul bancar și guvernamental și semnătura digitală. Semnătura digitală se bazează pe utilizarea cheilor RSA pentru a asigura integritatea și autenticitatea datelor digitale. În acest proces, mesajul original este semnat digital folosind cheia privată, iar semnătura digitală este verificată folosind cheia publică a emițătorului.

*Semnătura digitală*

Semnătura digitală este o metodă de autentificare și de verificare a integrității datelor digitale folosind algoritmi criptografici și chei asimetrice. Acest proces implică utilizarea unei chei private pentru a semna un document digital și a unei chei publice pentru a verifica semnătura digitală. În cazul algoritmului RSA, cheia privată este utilizată pentru a semna documentul digital și cheia publică este utilizată pentru a verifica semnătura.

**Pentru a semna un document digital folosind RSA, se aplică un algoritm de criptare asimetrică pe hash-ul (rezumatul) documentului digital. Hash-ul este o valoare unică și fixă, calculată prin aplicarea unui algoritm de hash (cum ar fi SHA-256 sau SHA-512) pe conținutul documentului digital. Această valoare este apoi semnată cu cheia privată a emițătorului și atașată la documentul digital.**

**Pentru a verifica semnătura digitală, destinatarul utilizează cheia publică a emițătorului pentru a decripta semnătura și pentru a obține hash-ul original al documentului digital. Acesta calculează apoi hash-ul documentului digital pe care îl primește și compară valoarea calculată cu cea obținută prin decriptarea semnăturii digitale. Dacă valorile coincid, atunci semnătura digitală este autentică și integritatea datelor digitale este confirmată.**

**Semnătura digitală este utilizată într-o varietate de domenii, inclusiv în sectorul financiar, guvernamental și în comerțul electronic, pentru a asigura autenticitatea și integritatea documentelor digitale și a altor informații sensibile.**

**3. Realizarea aplicației**

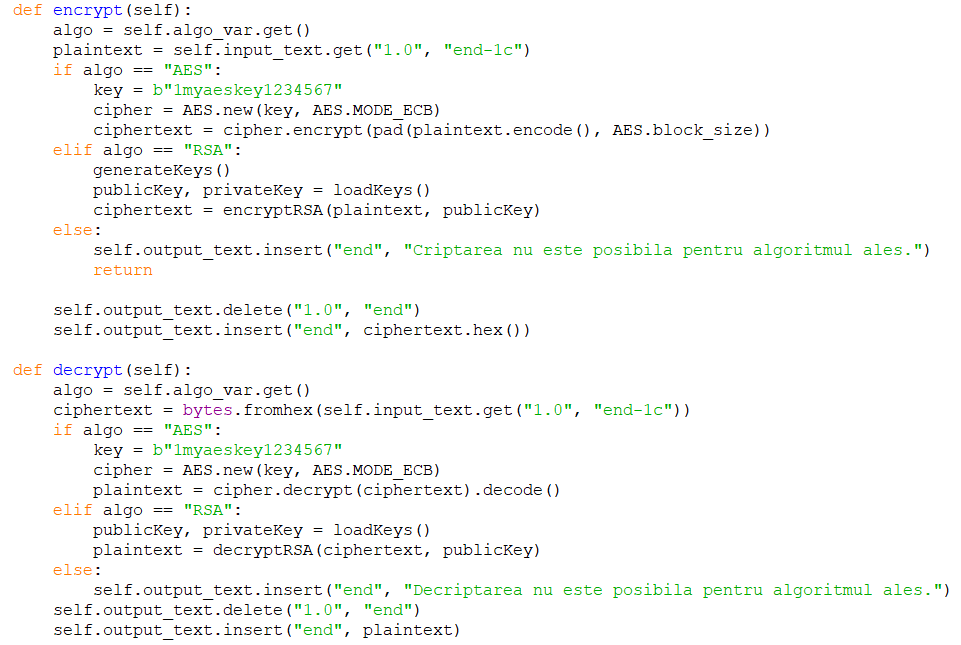
* Realizarea unei aplicații într-un limbaj de programare la alegere
* Interfața principală va reprezenta un meniu cu algoritmii DES/AES, RSA și semnătura digitală
* Opțiunile la alegere sunt: Implementarea DES sau AES, la semnătura digitală fie RSA sau DSA

Pentru implimentare am ales algoritmul AES. Pentru al implimenta am instalat libraria Crypto și pycryptodome. De asemenea acestea mi-a fost de ajutor pentru realizarea RSA și semnătura digitală.



**Figura 2.** RSA

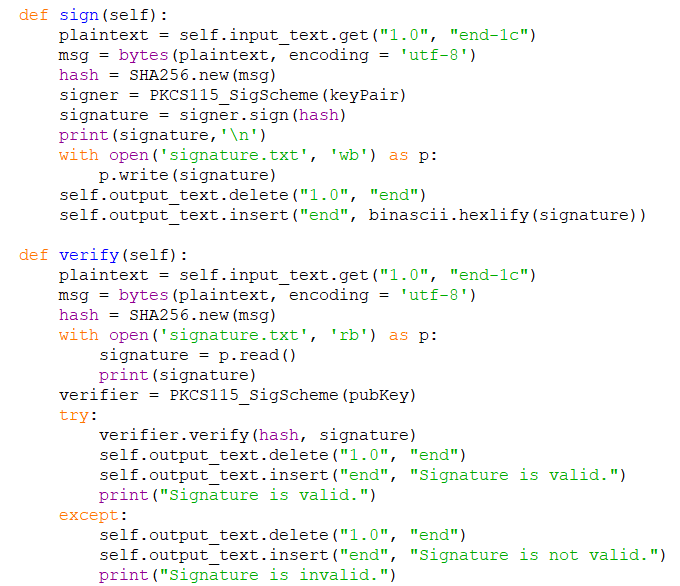
În figura 2, este reprezantat o parte din cod, unde are loc generarea cheilor private și publice. De asemenea are loc criptare și decriptarea prin RSA.



**Figura 3.** Criptarea și decriptare.

În figura 3 este reprezentată criptarea și decriptarea pentru algortimii RSA și AES, precum și o verificare daca funcția este disponibilă pentru un algoritm ales.

Mai jos, în figura 4, are loc semnare și verificare unei semnături digitale.



**Figura 4.** Semnare și verificare.

Pentru o utilizare mai efectivă, am creat o interfață simplă și clară, care este reprezentată în figura 5.



**Figura 5**. Interfața grafică.

**Concluzie**

În urma efectuării acestor puncte, se poate concluziona că am reușit să studiem și să înțelegem principiile de bază al algoritmului de criptare AES, algoritmul RSA și semnătura digitală. Am aprofundat modelul matematic și funcționalul fiecărui algoritm, precum și domeniile de aplicare ale acestora.

De asemenea, am dezvoltat o aplicație într-un limbaj de programare la alegere, care include o interfață grafică intuitivă și prietenoasă pentru utilizatori. Această interfață include un meniu cu opțiuni pentru implementarea algoritmului AES, algoritmul RSA și semnătura digitală. Am ales limbajul de programare Python pentru realizarea acestei aplicații, dar am recomandat și alte limbaje de programare, cum ar fi C++, C#, Java, în funcție de preferințele și cunoștințele fiecărui utilizator.

În general, acest raport a avut ca scop să ofere o înțelegere mai profundă a algoritmilor de criptare și a semnăturii digitale și să demonstreze modul în care aceștia pot fi implementați într-o aplicație cu interfață grafică. Sperăm că acest raport va fi util pentru cei interesați de criptarea datelor și de securitatea informațiilor în general.