**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Raport**

Lucrarea de laborator nr.1

Disciplina: Securitatea Informațională

Tema: Algoritmul DES/AES

**Efectuat**: st.gr. TI-202 Bunescu Gabriel

**Verificat**: asist. univ. Todos Alexandru

Chișinău 2023

**Scopul lucrării:**

Elaborarea unui algorit de criptare și decriptare.

**Mersul lucrării:**

1. Studierea algoritmului de criptare DES şi AES
2. Analiza comparativă a acestora
3. Realizarea unei aplicaţi de criptare ce va utiliza algoritmul DES sau AES
4. **DES (Data Encryption Standard):**

* Istoric:

DES a fost dezvoltat în anii 1970 de către IBM și a fost adoptat ulterior ca standard federal de criptare în Statele Unite ale Americii. A fost utilizat pe scară largă în industrie și guvern pentru a securiza datele.

* Funcționare:

DES utilizează o cheie secretă de 56 de biți pentru a cripta și decripta datele. Datele de intrare sunt împărțite în blocuri de 64 de biți și sunt supuse unei serii de permutări și operații logice pentru a produce textul cifrat.

* Probleme de securitate:

Cu trecerea timpului, puterea computațională a crescut, ceea ce a făcut posibilă spargerea cheilor DES prin forță brută într-un timp rezonabil. Din acest motiv, DES a fost înlocuit treptat cu algoritmi mai puternici precum AES.

**AES (Advanced Encryption Standard):**

* Istoric:

AES a fost selectat ca înlocuitor pentru DES în 2001 prin intermediul unui proces de selecție publică.

A fost dezvoltat de către doi criptografi belgieni, Vincent Rijmen și Joan Daemen.

* Funcționare:

AES este un algoritm de bloc simetric care acceptă chei de 128, 192 sau 256 de biți. Datele de intrare sunt împărțite în blocuri de 128 de biți și sunt supuse unor runde de substituție, permutare și operații logice în funcție de cheie.

* Securitate:

AES este considerat un algoritm de criptare extrem de securizat și rezistent la atacuri de forță brută, atât timp cât cheia este aleasă și gestionată corect. Este utilizat pe scară largă în întreaga lume pentru a proteja datele sensibile, cum ar fi informațiile bancare și de securitate națională.

1. **Analiza comparativă a acestora**

* Securitate:

DES: Securitatea DES a fost compromisă în decursul timpului. Cu avansul tehnologiei, s-a demonstrat că este vulnerabil la atacuri de forță brută și alte metode de criptanaliză. Lungimea scurtă a cheii (56 de biți) este unul dintre principalele sale puncte slabe.

AES: AES este considerat mult mai securizat decât DES. Lungimea cheilor de 128, 192 și 256 de biți oferă niveluri semnificativ mai mari de securitate. În plus, nu s-au găsit vulnerabilități semnificative în algoritmul AES în ciuda unui efort considerabil pentru a-l evalua.

* Eficiență:

DES: DES este un algoritm mai vechi și, în unele cazuri, poate fi mai lent decât AES. Operațiile de permutare și substituție necesită mai multe cicluri de ceas în comparație cu AES.

AES: AES este proiectat pentru a fi eficient din punct de vedere al resurselor, ceea ce îl face potrivit pentru o gamă largă de dispozitive și aplicații. Este rapid și ușor de implementat în hardware sau software.

* Utilizare în prezent:

DES: Deși a fost odată un standard, utilizarea DES a fost în mare parte abandonată în favoarea AES. Cu toate acestea, poate fi încă întâlnit în unele sisteme mai vechi sau în contexte istorice.

AES: AES este larg utilizat în întreaga lume și este algoritmul de criptare standard în multe protocoale de securitate și aplicații, inclusiv SSL/TLS pentru securitatea internetului și dispozitivelor de stocare criptate.

* Rezistența la atacuri:

DES: DES este vulnerabil la atacuri de forță brută și criptanaliză diferențială și liniară. Cu tehnologia actuală, cheile DES pot fi sparte într-un timp rezonabil.

AES: AES este considerat rezistent la o gamă largă de atacuri cunoscute, inclusiv cele bazate pe forță brută, criptanaliză diferențială și liniară. Acesta oferă o securitate mai mare și este adaptat pentru lumea modernă a cibernetică.

1. **Realizarea unei aplicaţi de criptare ce va utiliza algoritmul AES, ulilizînd biblioteca/protocolul Fernet (criptare simetrică).**

Fernet folosește algoritmul Advanced Encryption Standard (AES) pentru a codifica și decoda mesajele. AES este un algoritm de criptare foarte sigur, utilizat pe scară largă și popular, folosit de dezvoltatori.

Textele cifrate ale Fernet sunt sigure pentru URL, ceea ce înseamnă că putem trimite textele cifrate prin World Wide Web, făcând transmisia de date mai convenabilă.

Fernet generează o cheie alfanumerică foarte sigură folosind un generator de numere aleatorii. Este o cheie lungă de 32 de octeți, ceea ce o face foarte rezistentă la atacurile de forță brută.

De asemenea, acceptă rotația cheilor, adică capacitatea de a genera chei noi și de a înlocui cheile vechi tot timpul.

Fernet acceptă marcarea timpului și serializarea datelor care urmează să fie atașate împreună cu cheia. Acest lucru se face pentru a îmbunătăți securitatea cheii, deoarece atașarea unei mărci temporale la cheie va asigura că aceasta are valabilitate limitată.

**Codul Sursă:**

**from** cryptography.fernet **import** Fernet

**def** generate\_key(key\_size):

**if** key\_size == 128:

key = Fernet.generate\_key()

key\_size\_in\_bits = 128

**elif** key\_size == 192:

key = Fernet.generate\_key()

key\_size\_in\_bits = 192

**elif** key\_size == 256:

key = Fernet.generate\_key()

key\_size\_in\_bits = 256

**else**:

**raise** ValueError("Dimensiunea cheii nu este validă. Alegeți una dintre: 128, 192 sau 256.")

**return** key, key\_size\_in\_bits

**def** encrypt\_message(key, plaintext):

f = Fernet(key)

encrypted\_message = f.encrypt(plaintext.encode())

**return** encrypted\_message

**def** decrypt\_message(key, encrypted\_message):

f = Fernet(key)

decrypted\_message = f.decrypt(encrypted\_message).decode()

**return** decrypted\_message

**try**:

key\_size = int(input("Introduceți dimensiunea cheii (128, 192 sau 256): "))

**if** key\_size **not** **in** [128, 192, 256]:

**raise** ValueError("Dimensiunea cheii nu este validă. Alegeți una dintre: 128, 192 sau 256.")

key, key\_size\_in\_bits = generate\_key(key\_size)

**print**(f"Cheia secretă generată ({key\_size\_in\_bits} biți):", key.decode())

plaintext = input("Introduceți textul pe care doriți să-l criptați: ")

encrypted\_message = encrypt\_message(key, plaintext)

**print**("Mesaj criptat:", encrypted\_message)

decrypted\_message = decrypt\_message(key, encrypted\_message)

**print**("Mesaj decriptat:", decrypted\_message)

**except** ValueError **as** e:

**print**(e)

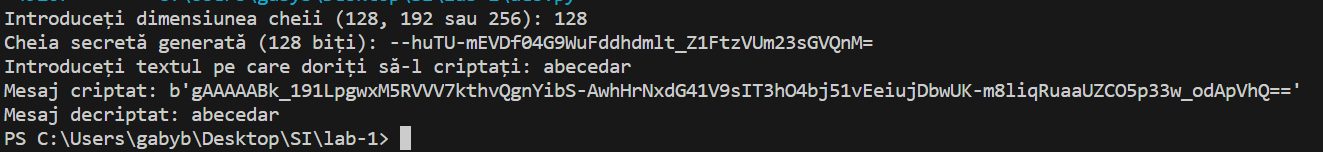


Figura 1 – Executarea pe 128 de biți.

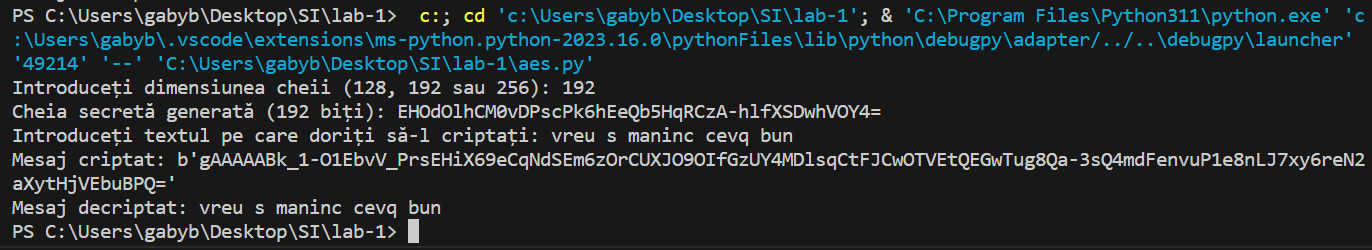


Figura 2 – Executarea pe 192 de biți.

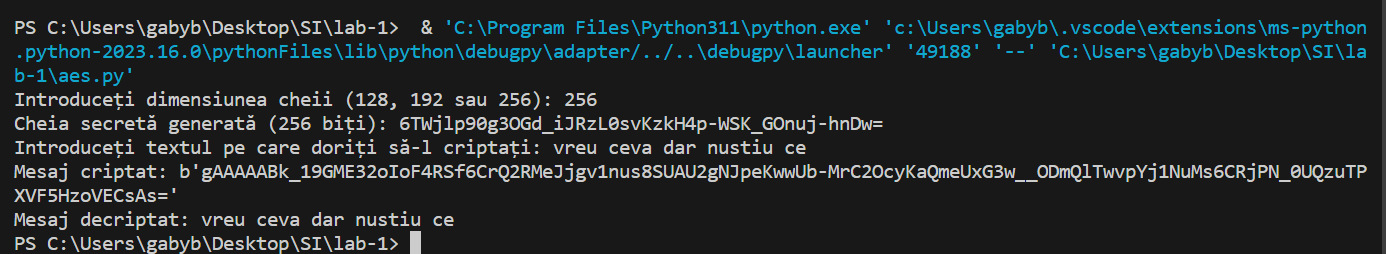


Figura 3 – Executarea pe 256 de biți.

În cazul în care alegerea este între DES și AES, AES este întotdeauna alegerea mai bună din punct de vedere al securității, eficienței, lungimea cheilor și standard actual. DES este acum considerat învechit și nu ar trebui folosit în aplicații sau sisteme care necesită un nivel ridicat de securitate.

**Concluzie:**

În concluzie, DES a fost un standard important în trecut, dar a fost înlocuit de AES datorită securității mai bune. AES este acum cel mai utilizat algoritm de criptare simetric și este folosit în numeroase aplicații pentru a proteja datele împotriva atacurilor cibernetice.

AES este preferat în mod clar în comparație cu DES în majoritatea contextelor datorită nivelului său superior de securitate și eficiență. DES este acum considerat învechit și nerecomandat pentru utilizare în aplicații critice de securitate. AES reprezintă standardul actual pentru criptarea datelor și este larg utilizat în industrie și în sistemele de securitate moderne

Aplicația data este un codul de implementare tipică a utilizării algoritmului AES într-o aplicație de criptare simetrică cu ajutorul bibliotecii Fernet din modulul cryptography în Python. Acesta servește drept punct de plecare pentru crearea unei aplicații de criptare și decriptare a datelor, oferind funcționalități de bază pentru generarea cheilor, criptarea și decriptarea mesajelor.

**Bibliografie:**

Codul sursă [17.09.2023] Disponibil:

https://8gwifi.org/fernet.jsp