Python II

Kryptografi:

Indhold:

Faglige mål:

Øvelsen dækker følgende målpinde:

1. *Eleven kan anvende Python til opbygning af en datapipeline baseret på ETL-programmeringsmønsteret.*
2. *Eleven kan oprette og bruge egne Python-moduler i komplekse programmeringsprojekter, så funktionalitet kan genbruges og vedligeholdes nemt.*
3. *Eleven kan anvende Python-moduler til at hente (extract) data fra internet- og netværkskilder og integrere dem i en programmeringspipeline på en sikker og robust måde, der demonstrerer beskyttelse af datatransport mod angreb samt håndtering af ustabile netværksforbindelser.*
4. *Eleven kan implementere databehandling ved hjælp af Python-moduler til databehandling, såsom pandas, dask eller PySpark.*
5. **Eleven kan anvende kryptografiske metoder til beskyttelse af data i en programmeringspipeline.**
6. *Eleven kan programmere mod databaser og anvende SQL i forbindelse med Python-programmering og gøre rede for, hvordan det anvendes i dataanalysen.*
7. *Eleven kan anvende Python-moduler til grafisk visualisering af data og gøre rede for, hvordan det anvendes i dataanalysen.*

Øvelse (Case) :

Baggrund

Du er ansat i en softwareudvikling organisation. Du får et naturcenter som kunde som ønsker at implementer et system på en Linux setup til analysering af blomster data til overvågning og dokumentation, med henblik på forskning. Du har tidliger udviklede et Python datapipeline til kunden, hvor kunden kan fortag dataanalyse via SQL samt datavisualisering.

Det viser sig, at det transformeret data du har implementeret under ETL-trasform er vigtige og skal beskyttes med kryptografi.

Problemformulering

- Hvordan kan du, implementer kryptografi på det data som er transformeret og gemt, både data gemt i csv filen og data i databasen.
- Hvordan kan du implementer dekryptering når de data genindlæs til generering af de 3 diagrammer du implementeret under datavisualisering.

Baggrundsteori

Egenskab	AES-GCM	AES-CBC
Krypteringstype	Authenticated Encryption (AEAD)	Symmetrisk blokchiffer
Giver integritet & autenticitet	✅ Ja (indbygget authentication tag)	❌ Nej (kræver HMAC separat)
Typisk brug i Python	AESGCM	Cipher(algorithms.AES, modes.CBC)
IV / Nonce	Nonce (typisk 12 bytes)	IV (16 bytes, blokstørrelse)
IV/Nonce-krav	Skal være unik, ikke hemmelig	Skal være uforudsigelig
Padding nødvendig	❌ Nej	✅ Ja (fx PKCS7)
Risiko ved forkert brug	Lavere (sværere at misbruge korrekt)	Højere (padding oracle attacks)
Performance	Hurtig (kan udnytte hardware-acceleration)	Langsommere
Modstandsdygtig mod manipulation	✅ Ja	❌ Nej
Anbefalet til nye systemer	⭐ ⭐ ⭐ ⭐ ⭐	⭐
Almindelige anvendelser	API'er, tokens, filer, netværk	Legacy-systemer

Krav

- Du skal lave en kopi af koden fra projektets første itiration.
- Du skal i kopien opdater din datapipeline således, at data gemmes (ETL-Load) krypteret med AES symetrisk kryptering både i den transformeret csv fil, samt i databasen.
    - Du skal implementer en **security modul** som indeholder følgende 3 kryptering metoder:
        - Krypter med AES-GCM operation-mode.
        - Krypter med AES-CBC operation-mode.
        - Krypter med AES-CBC operation-mode, men med Fernet teknologi.

          Deu skal undersøg og vælge hvilket af de 3 metoder du vil anvende til kryptering hvor du forklare med kode-kommentar hvorfor du har valgt metoden frem for de andre, lige præcis for din datatype.

    - Du skal herefter opdater din kode, så data indlæst til diagrammerne bliver dekrypteret ind brug til generering af graferne.

Hvordan bliver du bedømmet

- Fremvis kørsel af din Python script.