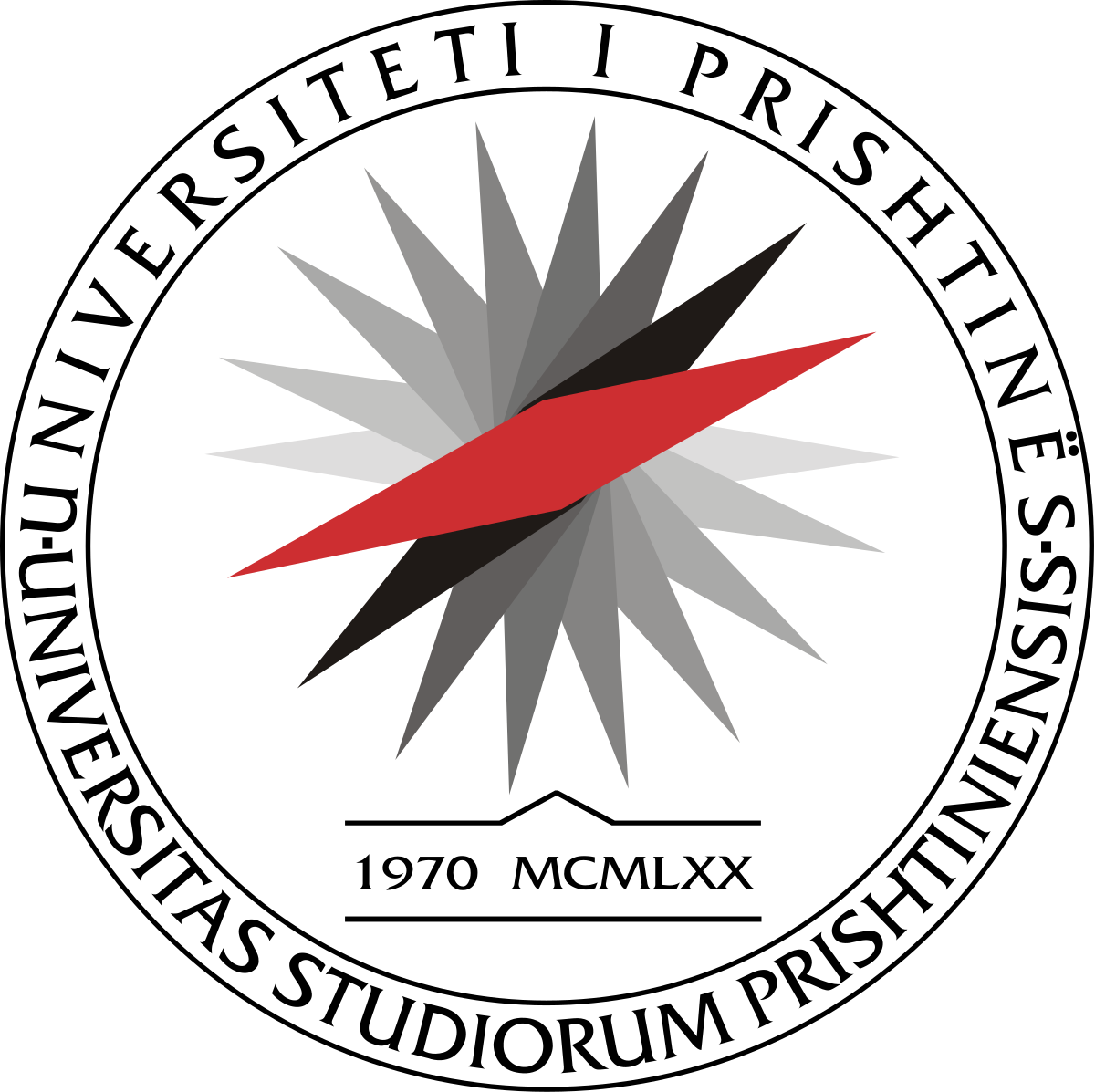
UNIVERSITETI I PRISHTINES “HASAN PRISHTINA”

FAKULTETI I SHKENCAVE MATEMATIKE-NATYRORE



Drejtimi:Shkenca Kompjuterike

**Detyra 5**

Punoi: Alketa Hajdari dhe Elda Meshi

# Paraqitja e problemit

5) Implementoni në Java skemën e enkriptimit dhe dekriptimit bazuar në skemën e algoritmit AES i thjeshtuar. Aplikacioni duhet të marrë si hyrje çelësin sekret dhe nje tekst dokument (text file) dhe të bëjë enkriptimin/dekriptimin e përmbajtjes së atij tekst dokumenti. Pastaj implementoni algoritmin AES i thjeshtuar në mënyrën e enkriptimit/dekriptimit GCM?

Sqarim për detyrat 1) deri 5): Te secila mënyrë e enkriptimit/dekriptimit duhet të kemi parasysh parametrat hyrës përkatës. Te mënyra e enkriptimit/dekriptimit GCM veprimet aritmetikore do të kryhen në fushën GF(2^16)/P(x). Ku si P(x) duhet të zgjedhni njërin polinom ireducibil të dhënë në formën oktale: 210013, 234313, 233303, 307107, 307527, 306357, 201735, 272201, 242413, 270155, 302157, 210205, 305667, 236107.

# Hyrja

Modaliteti Galois/Counter (GCM) është një mënyrë tipike e funksionimit të shifrimit të bllokut duke përdorur algoritmin e shifrimit të bllokut. Në këtë version, ne ofrojmë aftësinë e përpunimit të standardit të përparuar të enkriptimit (AES), gjatësia e çelësit të shifrimit për AES duhet të jetë 128/192/256 bit. Implementimi ynë merr një ngarkesë me madhësi fikse (128 bit për bllok) dhe transmetime shtesë të të dhënave të vërtetuara (AAD), por teksti mund të ketë gjatësi të ndryshme në situata të ndryshme. Kështu, ju duhet të jepni gjatësinë në bit të shoqëruar me të dhënat ose AAD. Ndërkohë, shifrat që janë prodhimi i primitivit gjithashtu përputhen me këtë model.

# Hyrjet dhe Daljet

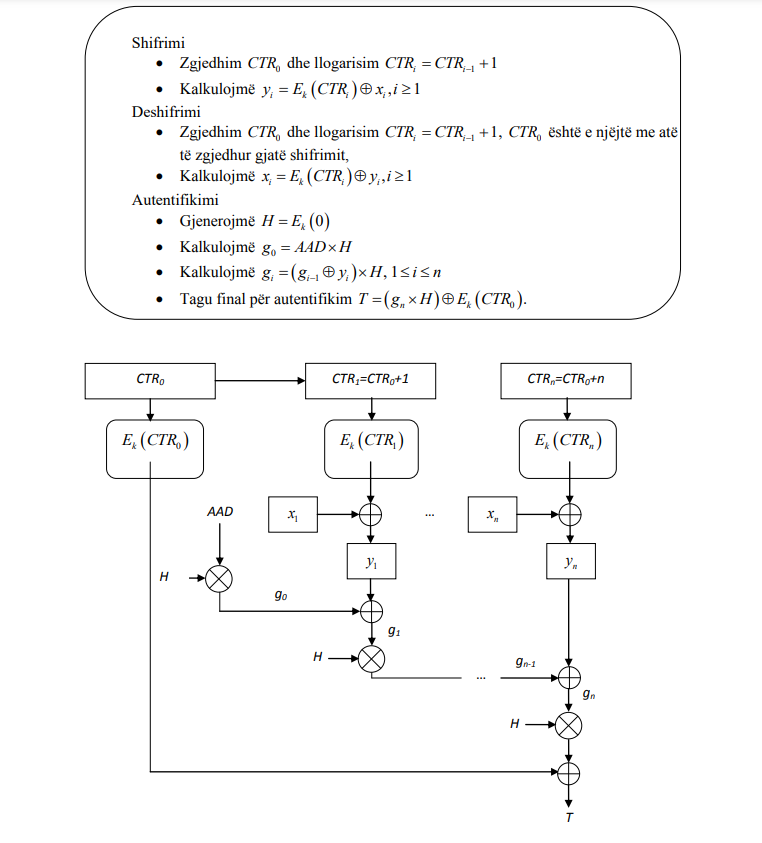
GCM ka dy operacione, enkriptim i vërtetuar dhe deshifrim i vërtetuar. Operacioni i kriptimit të vërtetuar ka katër hyrje, secila prej të cilave është një varg bit:

* Një çelës sekret K, gjatësia e të cilit është e përshtatshme për shifrën e bllokut themelor(block cipher)
* Një vektor inicializimi IV, që mund të ketë çdo numër bitësh midis 1 dhe 2^64. Për një fixed value te celesit, çdo vlerë IV duhet të jetë e dallueshme, por nuk duhet të ketë gjatësi të barabarta. 96-bit vlerat IV mund të përpunohen në mënyrë më efikase, kështu që gjatësia rekomandohet për situatat në efikasiteti i cili është kritik.
* Një tekst i thjeshtë P, i cili mund të ketë çdo numër bitësh midis 0 dhe 2^39 − 256.
* Të dhëna shtesë të vërtetuara (AAD), të cilat shënohen si A. Këto të dhëna vërtetohen, por nuk është i enkriptuar dhe mund të ketë çdo numër bitësh midis 0 dhe 2^64.

Ka dy dalje:

* Një tekst shifror C gjatësia e të cilit është pikërisht ajo e tekstit të thjeshtë P.
* Një etiketë vërtetimi T, gjatësia e të cilit mund të jetë çdo vlerë midis 0 dhe 128. \Gjatësia e etiketes shënohet si t.

# Shifrimi, deshifrimi dhe autentifikimi me anë të GCM



Për të implementuar skemën e enkriptimit dhe dekriptimit bazuar në skemën e algoritmit AES i thjeshtuar, mund të përdorni bibliotekën e Java për AES. Këtu është një shembull i kodit që mund të përdorni:

import javax.crypto.Cipher;

import javax.crypto.spec.SecretKeySpec;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.Arrays;

public class AES {

private static final String ALGORITHM = "AES";

private static final String TRANSFORMATION = "AES/ECB/PKCS5Padding";

private static final int KEY\_SIZE = 128;

public static void encrypt(String key, String inputFile, String outputFile) throws Exception {

doCrypto(Cipher.ENCRYPT\_MODE, key, inputFile, outputFile);

}

public static void decrypt(String key, String inputFile, String outputFile) throws Exception {

doCrypto(Cipher.DECRYPT\_MODE, key, inputFile, outputFile);

}

private static void doCrypto(int cipherMode, String key, String inputFile, String outputFile) throws Exception {

SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(), ALGORITHM);

Cipher cipher = Cipher.getInstance(TRANSFORMATION);

cipher.init(cipherMode, secretKeySpec);

Path inputFilePath = Paths.get(inputFile);

byte[] inputBytes = Files.readAllBytes(inputFilePath);

byte[] outputBytes = cipher.doFinal(inputBytes);

Path outputFilePath = Paths.get(outputFile);

Files.write(outputFilePath, outputBytes);

}

}

Në këtë shembull, encrypt() dhe decrypt() janë dy metoda që marrin si hyrje një çelës sekret, një file teksti si hyrje dhe një file për output. Këto metoda thjeshtë thirrjnë doCrypto() me argumente të ndryshme për modin e enkriptimit ose dekriptimit.

Metoda doCrypto() inicializon një SecretKeySpec nga çelësi i dhënë, pastaj krijon një instance të Cipher duke përdorur modin dhe transformimin e dhënë. Pastaj, metoda lexon të gjithë bytet e file-it hyrës, kryen enkriptimin/dekriptimin me anë të Cipher, dhe shkruan bytet e enkriptuar/dekriptuar në file-in për output.

Për të implementuar AES në mënyrën e enkriptimit/dekriptimit GCM, mund të përdorni klasën GCMParameterSpec për të përcaktuar parametrat e hyrjes për algoritmin GCM. Këtu është një shembull i kodit që mund të përdorni:

import javax.crypto.Cipher;

import javax.crypto.spec.GCMParameterSpec;

import javax.crypto.spec.SecretKeySpec;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.security.SecureRandom;

import java.util.Arrays;

public class AESGCM {

private static final String ALGORITHM = "AES/GCM/NoPadding";

private static final int TAG\_LENGTH = 128;

public static void encrypt(String key, String inputFile, String outputFile) throws Exception {

byte[] iv = generateIV();

Cipher

**Arkitektura MCV**

**Arkitektura MVC**

AES: Përfaqëson modelin në arkitekturën MVC.

Pamja mbledh hyrjet e përdoruesit si çelësi, shtegu i skedarit hyrës dhe shtegu i skedarit të daljes.

Kontrolluesi vepron si një ndërmjetës midis modelit dhe pamjes, duke koordinuar ndërveprimet midis tyre.

Trajton hyrjet e përdoruesit dhe aktivizon veprimet përkatëse në model.

Thërret metodat e duhura nga modeli (AES) bazuar në kërkesat e përdoruesve.

Pamja thërret metodat e duhura nga kontrolluesi për të nisur procesin e enkriptimit ose deshifrimit.

Në kontekstin e kodit të dhënë, pamja mund të përfaqësohet nga ndërfaqja e përdoruesit ose ndërfaqja e linjës së komandës (CLI) nga e cila aktivizohen operacionet e kriptimit dhe deshifrimit.

Përmban logjikën e kriptimit dhe deshifrimit duke përdorur algoritmin AES.