Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**Raport**

Disciplina: Securitatea informațională.

Lucrarea de laborator nr. 1

**Tema:** Securitatea informațională.

A efectuat: st.gr. TI-194 Ceban Vitalie

A verificat: asis. univ. Răducanu Octavian

Chişinău 2022

**Scopul lucrării:**

- Studierea algoritmului de criptare DES şi AES

- Analiza comparativă a acestora

**Obiectivele lucrării:**

Realizarea unei aplicaţi de criptare ce va utiliza algoritmul DES sau AES.

package com.aes;  
  
import javax.crypto.Cipher;  
import javax.crypto.KeyGenerator;  
import javax.crypto.SecretKey;  
import javax.crypto.spec.GCMParameterSpec;  
import java.util.Base64;  
  
public class AES\_ENCRYPTION {  
 private SecretKey key;  
 private final int KEY\_SIZE = 128;  
 private final int DATA\_LENGTH = 128;  
 private Cipher encryptionCipher;  
  
 public void init() throws Exception {  
 KeyGenerator keyGenerator = KeyGenerator.*getInstance*("AES");  
 keyGenerator.init(KEY\_SIZE);  
 key = keyGenerator.generateKey();  
 }  
  
 public String encrypt(String data) throws Exception {  
 byte[] dataInBytes = data.getBytes();  
 encryptionCipher = Cipher.*getInstance*("AES/GCM/NoPadding");  
 encryptionCipher.init(Cipher.*ENCRYPT\_MODE*, key);  
 byte[] encryptedBytes = encryptionCipher.doFinal(dataInBytes);  
 return encode(encryptedBytes);  
 }  
  
 public String decrypt(String encryptedData) throws Exception {  
 byte[] dataInBytes = decode(encryptedData);  
 Cipher decryptionCipher = Cipher.*getInstance*("AES/GCM/NoPadding");  
 GCMParameterSpec spec = new GCMParameterSpec(DATA\_LENGTH, encryptionCipher.getIV());  
 decryptionCipher.init(Cipher.*DECRYPT\_MODE*, key, spec);  
 byte[] decryptedBytes = decryptionCipher.doFinal(dataInBytes);  
 return new String(decryptedBytes);  
 }  
 private String encode(byte[] data) {  
 return Base64.*getEncoder*().encodeToString(data);  
 }  
  
 private byte[] decode(String data) {  
 return Base64.*getDecoder*().decode(data);  
 }  
 public static void main(String[] args) {  
 try {  
 AES\_ENCRYPTION aes\_encryption = new AES\_ENCRYPTION();  
 aes\_encryption.init();  
 String encryptedData = aes\_encryption.encrypt("Some text to encrypt");  
 String decryptedData = aes\_encryption.decrypt(encryptedData);  
  
 System.*out*.println("Encrypted Data : " + encryptedData);  
 System.*out*.println("Decrypted Data : " + decryptedData);  
 } catch (Exception ignored) {  
 }  
 }  
}