МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №1

“Основы шифрования данных”

по дисциплине

“Информационная безопасность”

Вариант №5

Студент:

Миху Вадим Дмитриевич

Группа P34301

Преподаватель:

Рыбаков Степан Дмитриевич

г. Санкт-Петербург

2024

Цель работы:

Изучение основных принципов шифрования информации, знакомство с широко известными алгоритмами шифрования, приобретение навыков их программной реализации.

Вариант 5:

Реализовать в программе шифрование и дешифрацию файла с использованием квадрата Кардано размером 4x4.

Выполнение:

Для реализации данного алгоритма был разработан скрипт на Scala, который принимает от пользователя данны), выполняет шифрование/дешифрацию текста из указанного входного файла с использованием введенного ключевого слова, а затем сохраняет результат в выходной файл.

Листинг разработанной программы с комментариями:

```
@main
def main(): Unit = {
  // ENCODE
  val inputFile = "resources/input.txt"
  val encodeFile = "encode.txt"
  val partSize = GridSize * GridSize
  val pattern = generatePattern()
  println("\tGenerated pattern")
  printGrid(pattern)

  println(s"\tReading from file - $inputFile")
  val input = fromFile(inputFile)
  val inputText = input.mkString
  println(inputText.take(100) + "...")
  val splitText = splitStringIntoBlocks(inputText, partSize)
  input.close()
  var charGrids: Array[CharGrid] = Array()
  splitText.foreach(stringPart => {
    charGrids :+= encodeStringByPattern(pattern, stringPart)
  })

  println("\tConvert to encoded string")
  val encodedStr = charGrids.map(grid => charGridToString(grid)).mkString
  println(encodedStr.take(100) + "...")

  //DECODE
  println("\tDecode")
  val writer = new PrintWriter(encodeFile)
  writer.write(encodedStr)
  writer.close()
  val decodedString = encodedStr.grouped(partSize).map(
    stringPart => decodeGridByPattern(pattern, stringToCharGrid(stringPart))
  ).mkString
  println(decodedString.take(100) + "...")
}
```

```
def charGridToString(charGrid: CharGrid): String = {
  charGrid.map(row => row.mkString("")).mkString("")
}

def stringToCharGrid(string: String): CharGrid = {
  val charGrid = generateCharGrid()
  val stringIter = string.iterator
  for (x <- charGrid.indices) {
    for (y <- charGrid.indices) {
      charGrid(x)(y) = stringIter.next()
    }
  }
  charGrid
}

def encodeStringByPattern(pattern: Grid, string: String): CharGrid = {
  val stringIter = string.iterator
  val charGrid = generateCharGrid()
  for (part <- 1 to 4) {
    val transform = partToTransform(part)
    val transformedPattern: Grid = transformGrid(pattern, transform)
    for (x <- pattern.indices) {
      for (y <- pattern.indices) {
        if (pattern(x)(y)) {
          val newCords = transform(Cord(x, y))
          charGrid(newCords.x)(newCords.y) = stringIter.next()
        }
      }
    }
  }
  charGrid
}

def decodeGridByPattern(pattern: Grid, charGrid: CharGrid): String = {
  val stringBuilder = StringBuilder()
  for (part <- 1 to 4) {
    val transform = partToTransform(part)
    val transformedPattern: Grid = transformGrid(pattern, transform)
    for (x <- pattern.indices) {
      for (y <- pattern.indices) {
        if (pattern(x)(y)) {
          val newCords = transform(Cord(x, y))
          stringBuilder.append(charGrid(newCords.x)(newCords.y))
        }
      }
    }
  }
  stringBuilder.mkString
}
```

```
import Const.GridSize

private type CordTransformer = Cord => Cord

def transform0(cord: Cord): Cord = cord
def transform90(cord: Cord): Cord = Cord(GridSize - cord.y - 1, cord.x)
def transform180(cord: Cord): Cord = Cord(GridSize - cord.x - 1, GridSize -
cord.y - 1)
def transform240(cord: Cord): Cord = Cord(cord.y, GridSize - cord.x - 1)

def transformGrid(grid: Grid, transformer: CordTransformer): Grid = {
  val newGrid = generateGrid()
```

```
  for (x <- grid.indices) {
    for (y <- grid.indices) {
      val newCords = transformer(Cord(x, y))
      newGrid(x)(y) = grid(newCords.x)(newCords.y)
    }
  }
  newGrid
}

def partToTransform(part: Int): CordTransformer = {
  part match
    case 1 => transform0
    case 2 => transform90
    case 3 => transform180
    case 4 => transform240
    case _ => throw NumberFormatException()
}
```

Результаты работы программы:

```
    Generated pattern
🟩⬜⬜⬜
🟩⬜🟩⬜
⬜⬜⬜⬜
⬜⬜🟩⬜
    Reading from file - resources/input.txt
Да мне всё равно на тебя, слушай. Какая у тебя там тачка, квартиры, яхты, всё. Мне всё равно, там, х...
    Convert to encoded string
Дрваа  во нёнемс лшунбаяйса те ,.еяб яК тт  каауаврамк аикт ачт,рс.ёыт,ыМв  ях ,но ,е  рантвсёвамнлт...
    Decode
Да мне всё равно на тебя, слушай. Какая у тебя там тачка, квартиры, яхты, всё. Мне всё равно, там, х...
```

Исходный текст:

```
Да мне всё равно на тебя, слушай. Какая у тебя там тачка, квартиры, яхты,
всё. Мне всё равно, там, хоть «Бэнтли», хоть «Майбах», хоть «Роллс-Ройс»,
хоть «Бугатти», хоть стометровая яхта. Мне на это всё равно, понимаешь?
Сколько ты там, кого имеешь, каких баб, каких вот этих самок шикарных или
атласных, в космос ты летишь, мне на это всё равно, понимаешь?
Я в своём познании настолько преисполнился, что я как будто бы уже сто
триллионов миллиардов лет проживаю на триллионах и триллионах таких же
планет, понимаешь, как эта Земля. Мне уже этот мир абсолютно понятен, и я
здесь ищу только одного: покоя, умиротворения и вот этой гармонии от слияния
с бесконечно вечным
```

Зашифрованный текст:

```
Дрваа  во нёнемс лшунбаяйса те ,.еяб яК тт  каауаврамк аикт ачт,рс.ёыт,ыМв
ях ,но ,е  рантвсёвамнлт,ь  »эиБотх«,бха  х«,й»атьоМ сР-х«оРйлоль
тосуаг»т,ьтБт«хо  иоем»т,ьрттсхо  оМенвхатн  . яяааанв вэс,ро о тё
кСпаоел?оьимншь окк ото,гмты а кхииьм,ба кеше ао тбк,итвэ ка хикрахо кыиншамс
хнхы аит с,аи ллв ел оксиыттсмо шэотье, в  амн нсоинён оапм авр,еомёш ьвпв с
Я? отлозин ксьанианоисл спп,нялеиро бдуч тко тк яоа  ртбеы лоитуж слираи
омолдловниввюа плрни жт еоахи  лтита нилрораикинла тх иолхжпноен ем и,лаптат
аекшаеэЗ , ькмеэ лняеожту М. тюнт амб лоор испяз онн,е дитея ск оь
тдьолщуион,у опгоиямо: окрив оетнт ояорви ниизгтаоо мй орт еб нсиксс ияляоы
мн ев н чноче
```

Расшифрованный текст:

```
Да мне всё равно на тебя, слушай. Какая у тебя там тачка, квартиры, яхты,
всё. Мне всё равно, там, хоть «Бэнтли», хоть «Майбах», хоть «Роллс-Ройс»,
хоть «Бугатти», хоть стометровая яхта. Мне на это всё равно, понимаешь?
```

```
Сколько ты там, кого имеешь, каких баб, каких вот этих самок шикарных или
атласных, в космос ты летишь, мне на это всё равно, понимаешь?
Я в своём познании настолько преисполнился, что я как будто бы уже сто
триллионов миллиардов лет проживаю на триллионах и триллионах таких же
планет, понимаешь, как эта Земля. Мне уже этот мир абсолютно понятен, и я
здесь ищу только одного: покоя, умиротворения и вот этой гармонии от слияния
с бесконечно вечным
```

Вывод:

В результате выполнения данной лабораторной работы я ознакомился с основными принципами шифрования информации, различными алгоритмами шифрования, а также реализовал алгоритм шифрования и дешифрации текста с использованием квадрата Кардано.