Национальный исследовательский университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники

Вариант №4а
Лабораторная работа №1.2
«Блочное симметричное шифрование»
По дисциплине:
«Информационная безопасность»

Работу выполнила:
Студентка группы P34102
Никонова Наталья Игоревна
Преподаватель:
Рыбаков Степан Дмитриевич

Санкт-Петербург
2023

Цель работы

Изучение структуры и основных принципов работы современных алгоритмов блочного симметричного шифрования, приобретение навыков программной реализации блочных симметричных шифров.

Задание

Реализовать систему симметричного блочного шифрования, позволяющую шифровать и дешифровать файл на диске с использованием заданного блочного шифра в заданном режиме шифрования.

По варианту: шифр - ГОСТ 28147-8, режим – ECB (электронной кодовой таблицы)

Листинг программы

```
val k = Array(8) { longToBinary(Random.nextLong(0, maxValueInt + 1), 32) }.asList()
val kR = k.asReversed()
val x = listOf(k, k, k, kR).flatten()
val xR = listOf(k, kR, kR, kR).flatten()
val s = Array (16) { (0..15).map { longToBinary(it.toLong(), 4) }.shuffled() }

fun main() {
    println("Введите название файла для зашифровки в одну строку (внутри можно использовать только символы ascii):")
    var fileName = readlnOrNull()
    while (fileName == null) {
        println("Вы ничего не ввели")
        fileName = readlnOrNull()
    }
    val fileName1 = fileName.split(".")[0]
    val fileFormat = fileName.split(".").getOrNull(1) ?: ""

    var input = File(fileName).inputStream().readBytes().joinToString("") { longToBinary(it.toLong(), 8) }
    println("Считано из файла (в битах)")
    println(input)
    val mod64 = input.length % 64
    if (mod64 != 0) {
        val padding = Array((64 - mod64) / 8) { charToBinary(' ') }.joinToString("")
        input += padding
    }
    val blocks = input.chunked(64)
    var result = ""

    for (block in blocks) {
        result += encoding(block)
    }

    val textResult = toTextResult(result)
```

```
    println("Кодировка")
    println(textResult)
    println(result)
    File(fileName1 + "_encod." +
fileFormat).writeText(textResult)

    val blocksDe = result.chunked(64)
    var resultDe = ""

    for (block in blocksDe) {
        resultDe += decoding(block)
    }
    val textResultDe = toTextResult(resultDe)
    println("Декодировка")
    println(textResultDe)
    println(resultDe)
    File(fileName1 + "_decod." +
fileFormat).writeText(textResultDe)
}

fun encoding(input: String): String = coding(input, x)
fun decoding(input: String): String = coding(input, xR)
fun coding(input: String, xUse: List<String>): String {
    val midList = input.chunked(32)
    var a = midList[1]
    var b = midList[0]
    for (i in xUse.indices) {
        val tmp = a
        a = xor(b, f(a, xUse[i]))
        b = tmp
    }
    return a + b
}

fun f(a: String, xI: String): String {
    val a1 = a.toLong(2)
    val xI1 = xI.toLong(2)
    val result1 = (a1 + xI1) % maxValueInt
    val result = longToBinary(result1, 32)
        .chunked(4)
        .mapIndexed { index, str -> s[index][str.toInt(2)] }
        .joinToString("")
        .toCharArray()
        .toList()
    Collections.rotate(result, -11)
    return result.joinToString("")
}

fun xor(a: String, b: String): String {
    val a1 = a.toLong(2)
    val b1 = b.toLong(2)
    val result = a1 xor b1
```

```
    return longToBinary(result,32)
}
```

Вывод программы

```
Введите название файла для зашифровки в одну строку (внутри можно использовать только символы ascii):
ex.txt
Считано из файла (в битах)
01100001
Кодировка
q␇h9-␣<.
011100011001001001101000001110010010110100010010001111000010111 0
Декодировка
a
0110000100100000001000000010000000100000001000000010000000100000
```

Выводы

В ходе выполнения данной лабораторной работы я познакомилась с одним из методов блочного симметричного шифрования, который в прошлом был государственным стандартом (с 1990 по 2019). Но зато стал прародителем одним из рекомендованных новых шифров – «Магма» - от которого по сути отличается обратным порядком байт вместо прямого и отсутствием фиксированной таблицы перестановок.