Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2
по курсу «СиМЗИИС»
на тему:
«Простейшие криптографические преобразования»

Выполнил студент группы 121703: Якимович И.В.

Проверил: Захаров В.В.

МИНСК, 2023

Задание

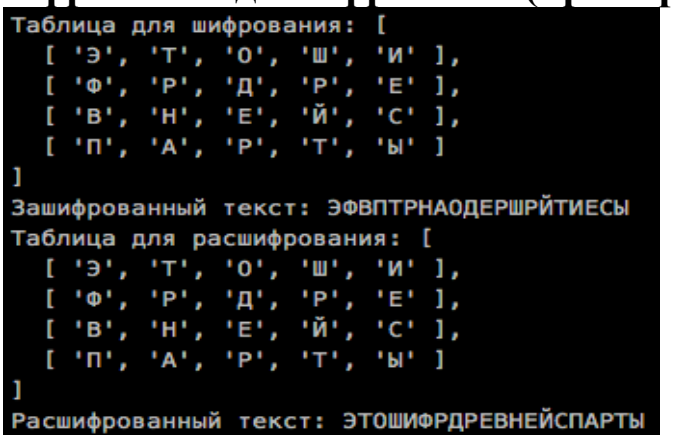
1) Реализовать в виде программы шифр (зашифрование и расшифрование) в соответствии с вариантом. Язык исходного текста русский или английский по выбору исполнителя.
2) Реализовать в виде программы атаку полным перебором ключа, используя для оценки правильности выбора ключа визуальный метод или исходный текст для автоматического сравнения результата дешифрования.
3) Оценить криптографическую стойкость реализованного шифра.
4) Предложить варианты усложнения шифра. Предложенные варианты оформить в виде алгоритма.
Варианты для реализации.
4) Шифр перестановки, использующий простые (прямоугольные) таблицы.

ЛИСТИНГ

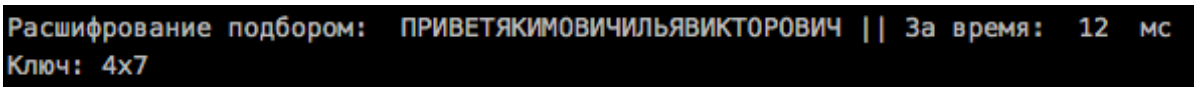
```
function encrypt(text, rows, columns) {
  text = text.replace(/\s/g, '')
  let encryptedText = ''

  // Вписывание текста в таблицу по строкам
  const table = []
  for (let i = 0; i < rows; i++) {
    const row = []
    for (let j = i * columns; j < i * columns + columns; j++) {
      row.push(text[j])
    }
    table.push(row)
  }

  // Вписывание текста в таблицу по строкам
  for (let i = 0; i < columns; i++) {
    for (let j = 0; j < rows; j++) {
      encryptedText += table[j][i]
    }
  }

  return {
    encryptedText,
    table,
  }
}

function decrypt(encryptedText, rows, columns) {
  let decryptedText = ''

  const table = []
  for (let i = 0; i < rows; i++) {
    const row = []
    for (let j = i; j < encryptedText.length; j += rows) {
      row.push(encryptedText[j])
    }
    table.push(row)
  }

  for (let i = 0; i < rows; i++) {
    for (let j = 0; j < columns; j++) {
      decryptedText += table[i][j]
    }
  }

  return {
    decryptedText: decryptedText.trim(),
    table,
  }
}
```

```
function bruteForceAttack(ciphertext, plaintext) {
  const maxRows = Math.ceil(ciphertext.length / 2) // Максимальное количество строк
  const maxColumns = Math.ceil(ciphertext.length / 2) // Максимальное количество столбцов

  let bestDecryption = '' // Переменная для хранения наилучшего расшифрованного текста
  let bestKey = '' // Переменная для хранения наилучшего ключа
  for (let rows = 1; rows <= maxRows; rows++) {
    for (let columns = 1; columns <= maxColumns; columns++) {
      const { decryptedText } = decrypt(ciphertext, rows, columns)

      // Сравниваем расшифрованный текст с исходным текстом
      if (decryptedText.toLowerCase() === plaintext.toLowerCase()) {
        // Если совпадение найдено, сохраняем результат и выходим из цикла
        bestDecryption = decryptedText
        bestKey = `${rows}x${columns}`
        break
      }
    }
  }

  return {
    bestDecryption,
    bestKey,
  }
}

const plaintext = 'ЭТО ШИФР ДРЕВНЕЙ СПАРТЫ'
const rows = 4
const columns = 5

const { encryptedText, table: encryptedTable } = encrypt(
  plaintext,
  rows,
  columns
)
const { decryptedText, table: decryptedTable } = decrypt(
  encryptedText,
  rows,
  columns
)
console.log('Таблица для шифрования:', encryptedTable)
console.log('Зашифрованный текст:', encryptedText)

console.log('Таблица для расшифрования:', decryptedTable)
console.log('Расшифрованный текст:', decryptedText)

const startTime = new Date().getTime()
const attackResult = bruteForceAttack(
  'ПЕАООРТНКВИИПОИВВРПЧ',
  'ПРИВЕТИВАНПРОКОПОВИЧ'
)
const endTime = new Date().getTime()
const time = endTime - startTime
console.log(
  'Расшифрование подбором: ',
  attackResult.bestDecryption,
  'll За время: ',
  time,
  ' мс'
)
console.log('Ключ:', attackResult.bestKey)
```

Шифрование/дешифрование(пример)



Пример подбора



Результаты подбора

	Среднее время
2x2 'ИРГА' 'ИГРА'	<1мс
2x4 'БРЕУЛСАЬ' 'БЕЛАРУСЬ'	1мс
4x5 'ЭФВПТРНАОДЕРШРЙТИЕСЫ' 'ЭТОШИФРДРЕВНЕЙСПАРТЫ'	7мс
4x7 'ПКИТРИЛОИМЬРВОЯОЕВВВТИИИЯЧКЧ' 'ПРИВЕТЯКИМОВИЧИЛЬЯВИКТОРОВИЧ'	12мс

Моя идея для усложнения шифра

После того, как мы построили таблицу, мы можем начать шифровать пароль не с первого столбца, а с какого-нибудь n-ого. Для этого нам понадобится ключ, который будет указывать номер столбца, с которого нужно начинать шифрование.

Пример: Беларусь

Матрица 2x4

Б	**Е**	**Л**	**А**
Р	**У**	**С**	**Ь**

А ключ будет, к примеру, 3

В зашифрованном виде: **ЛСАЬБРЕУ**

А при обычной шифровании было бы: **БРЕУЛСАЬ**

АЛГОРИТМ

1. Открытый текст построчно, начиная с верхней строки, впишите в таблицу, состоящую из m строк и n столбцов.
2. Задайте ключ, который будет указывать номер столбца, с которого нужно начинать шифрование. Номер столбца должен быть меньше или равен общему количеству столбцов в таблице.
3. Запишите символы из таблицы, начиная с выбранного столбца согласно ключу.

ВЫВОД

При проведении атаки полным перебором ключа было обнаружено, что шифр перестановки с простыми таблицами имеет низкую криптографическую стойкость. Это связано с тем, что при использовании простых таблиц возможностей для перебора ключа относительно невелико, и исходный текст может быть восстановлен сравнительно легко. Однако, при увеличении размеров матрицы время дешифрования увеличивается. Таким образом, для обеспечения более высокой стойкости шифра необходимо внести дополнительные модификации.