Front matter

title: "Отчёт по лабораторной работе № 8" subtitle: "Информационная безопасность" author: "Надиа Эззакат"

Generic otions

lang: ru-RU toc-title: "Содержание"

Bibliography

bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt

l18n polyglossia

polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true polyglossia-otherlangs: name: english

l18n babel

babel-lang: russian babel-otherlangs: english

Fonts

mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9

Biblatex

biblatex: true biblio-style: "gost-numeric" biblatexoptions:

- parentracker=true
- backend=biber
- hyperref=auto
- language=auto
- autolang=other*
- citestyle=gost-numeric

Pandoc-crossref LaTeX customization

figureTitle: "Рис." tableTitle: "Таблица" listingTitle: "Листинг" lofTitle: "Список иллюстраций" lotTitle: "Список таблиц" lolTitle: "Листинги"

Misc options

indent: true header-includes:

- \usepackage{indentfirst}
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{H} # keep figures where there are in the text

---

Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Теоретическое введение

Гаммирование - наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Основная формула, необходимая для реализации однократного гаммирования: Ci = Pi XOR Ki, где Ci - i-й символ зашифрованного текста, Pi - i-й символ открытого текста, Ki - i-й символ ключа. В данном случае для двух шифротекстов будет две формулы: C1 = P1 xor K и C2 = P2 xor K, где индексы обозначают первый и второй шифротексты соответственно. Если нам известны оба шифротекста и один открытый текст, то мы можем найти другой открытый текст, это следует из следующих формул: C1 xor C2 = P1 xor K xor P2 xor K = P1 xor P2, C1 xor C2 xor P1 = P1 xor P2 xor P1 = P2. Более подробно см. в [1].

Выполнение лабораторной работы

Код программы

Jupyter lab08 Last Checkpoint: last month

File Edit View Run Kernel Settings Help

Code

[43]:
```
import random
from random import seed
import string
```

[44]:
```
def cipher_text_function(text, key):
    if len(key) != len(text):
        return "ключ и текст должны быть одной длины!"
    cipher_text = ''
    for i in range(len(key)):
        cipher_text_symbol = ord(text[i]) ^ ord(key[i])
        cipher_text += chr(cipher_text_symbol)
    return cipher_text
```

[45]:
```
text_1 = " С Новым Годом,друзья! "
text_2 = " поздравляем с 8 марта!"
```

[46]:
```
key = ''
seed(23)
for i in range(len(text_1)):
    key += random.choice(string.ascii_letters + string.digits)
print(key)
```

```
7X8s51fbLtByHwiUmrCaoND
```

[47]:
```
cipher_text_1 = cipher_text_function(text_1, key)
cipher_text_2 = cipher_text_function(text_2, key)
print('Первый шифротекст:' , cipher_text_1)
print('Второй шифротекст:' , cipher_text_2)
```

```
Первый шифротекст:  ϭγ ѮђЃЭўlѦѼэѶыEѡЭбѵЭPod
Второй шифротекст:  ѧIфЁѱièѶлѶxhжImMюѳCЭѾe
```

[48]:
```
print('первый открытый текст:', cipher_text_function(cipher_text_1, key))
print('второй  открытый текст:', cipher_text_function(cipher_text_2, key))
```

```
первый открытый текст:  С Новым Годом,друзья!
второй  открытый текст:  поздравляем с 8 марта!
```

• In[43]: импорт необходимых библиотек 6 • In[44]: функция, реализующая сложение по модулю два двух строк • In[45]: открытые/исходные тексты (одинаковой длины) • In[46]: создание ключа той же длины, что и открытые тексты • In[47]: получение шифротекстов с помощью функции, созданной ранее, при условии, что известны открытые тексты и ключ • In[48]: получение открытых текстов с помощью функции, созданной ранее, при условии, что известны шифротексты и ключ

```
[53]: cipher_text_xor = cipher_text_function(cipher_text_1, cipher_text_2)
      print('первый шифротекст XOR второй шифротекст:', cipher_text_xor)

      первый шифротекст XOR второй шифротекст:  O*
      Б Л\O}Ќ

[50]: print('первый открытый текст:', cipher_text_function(cipher_text_xor, text_2))
      print('второй открытый текст:', cipher_text_function(cipher_text_xor, text_1))

      первый открытый текст:  С Новым Годом,друзья!
      второй открытый текст:  поздравляем с 8 марта!

[51]: text_1_ = text_1[3:6]
      print('часть первого открытого текста:', text_1_)

      часть первого открытого текста: Нов

[52]: cipher_text_xor_ = cipher_text_function(cipher_text_1[3:6], cipher_text_2[3:6])
      print('часть второго открытого текста:', cipher_text_function(cipher_text_xor_, text_1_))

      часть второго открытого текста: здр
```

• In[53]: сложение по модулю два двух шифротекстов с помощию функции, созданной ранее • In[50]: получение открытых текстов с помощью функции, созданной ранее, при условии, что известны оба шифротекста и один из открытых текстов • In[51]: получение части первого открытого текста (срез) • In[52]: получение части второго текста (на тех позициях, на которых расположены символы части первого открытого текста) с помощью функции, созданной ранее, при условии, что известны оба шифротекста и часть первого открытого текста

Выводы

В ходе выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Список литературы

Однократное гаммирование [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.php/1651641/mod_resource/content/2/008-lab_cryptokey.pdf.