
Front matter

title: "Математические основы защиты информации и информационной безопасности" subtitle:
"Отчёт по лабораторной работе №3:Шифрование гаммированием" author: "Ахлиддинзода
Аслиддин"

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

I18n polyglossia

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

I18n 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:

- parenttracker=true

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

Pandoc-crossref LaTeX customization

figureTitle: "Рис." tableTitle: "Таблица" listingTitle: "Листинг" 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

Основная цель работы — освоить шифрование гаммированием.

Выполнение лабораторной работы

Шифрование гаммированием

Реализация алгоритма шифрования гаммированием конечной гаммой

Код программы

```
function build_keystream(data_len::Int, key::Vector{UInt8}; repeat_key::Bool=true)
    if length(key) == 0
        throw(ArgumentError("Ключ не может быть пустым!"))
    end
    if length(key) >= data_len
        return key[1:data_len]
    elseif repeat_key
        n = ceil(Integer, data_len / length(key))
        replicated = repeat(key, n)
        return replicated[1:data_len]
    else
```

```

        throw(ArgumentError("Ключ короче текста, a repeat_key=false"))
    end
end
function encrypt_bytes(data::Vector{UInt8}, key::Vector{UInt8}; repeat_key::Bool=true)
    ks = build_keystream(length(data), key; repeat_key=repeat_key)
    return xor.(data, ks)
end
function bytes_to_hex(b::Vector{UInt8})
    hexbytes = map(x -> lpad(string(x, base=16), 2, '0'), b)
    return join(hexbytes, "")
end
function hex_to_bytes(hexstr::AbstractString)
    if length(hexstr) % 2 != 0
        throw(ArgumentError("Длина hex-строки должна быть чётной"))
    end
    return [parse(UInt8, hexstr[i:i+1], base=16) for i in 1:2:length(hexstr)]
end
println("Введите текст, который нужно зашифровать:")
plaintext = readline()

println("Введите ключ для шифрования:")
key_str = readline()

# Преобразуем в массивы байт
pt_bytes = collect(codeunits(plaintext))
key_bytes = collect(codeunits(key_str))

# Шифрование
cipher_bytes = encrypt_bytes(pt_bytes, key_bytes; repeat_key=true)
cipher_hex = bytes_to_hex(cipher_bytes)

println("\nЗашифрованный текст (hex): $cipher_hex")

# Дешифрование (для проверки)
dec_bytes = encrypt_bytes(cipher_bytes, key_bytes; repeat_key=true)
recovered_text = String(dec_bytes)

println("Дешифрованный текст: $recovered_text")

```

Вывод

В ходе выполнения лабораторной работы был реализован алгоритм шифрования гаммированием конечной гаммой.