Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное   
образовательное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Лабораторная Работа №4.   
сеть фейстеля

Отчет по лабораторной работе №4 по учебной дисциплине   
«Методы защиты информации»

по специальности 09.03.01 Информатика и вычислительная техника

|  |  |
| --- | --- |
|  | Руководитель  / Т. В. Жгун  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |
|  | Студент группы 3091  / Р. А. Михайлов  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Оглавление

[Решётка Кардано 3](#_Toc6393)

[Шифр Цезаря 5](#_Toc29947)

[Приложение А 9](#_Toc22184)

[Приложение Б 15](#_Toc25017)

Сеть Фейстеля

Реализовать работу сети Файстеля для блока 16 бит (шифрование и расшифрование). Число раундов 8. Исходный текст представлен в Приложениие А. В качестве ключа шифрования выступает пароль «Пустой колос голову кверху носит». Ниже представлене схема сети Фейстеля.

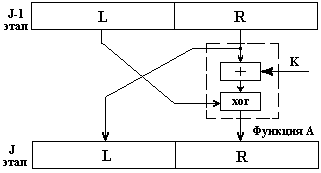


Рисунок 1 – Схема сети Файстеля

Рауновые ключи получаются путем сложение числовых значений символов по модулю 256.

Основной шаг криптопреобразования - наложение раундового ключа и сдвиг влево на 1 бит. Реализуемая программа выводит промежуточные результаты в соответсвующие файлы.

Гистограммы полученных промежуточных результатов представлены ниже. Полученный шифротекст представлен в приложении А.

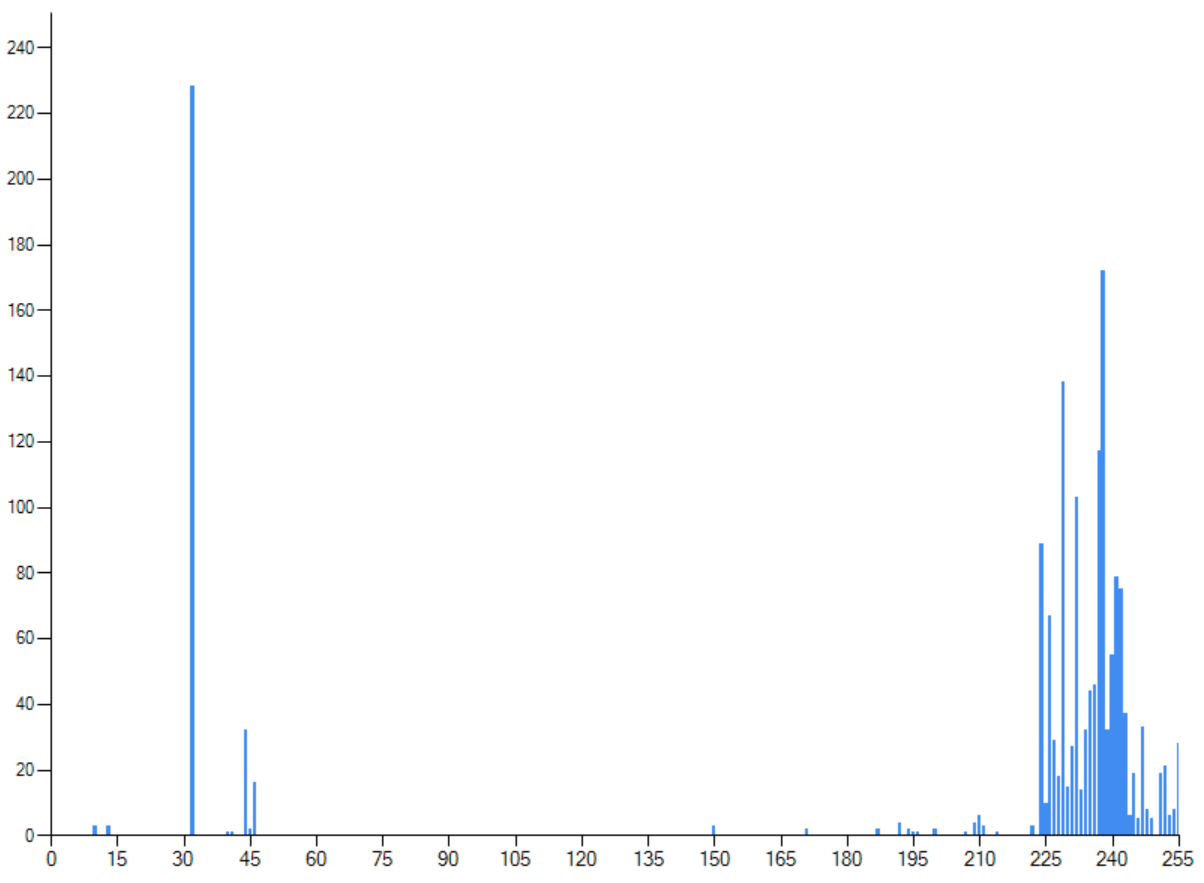


Рисунок 1 – Гистограмма открытого текста, H(x) = 4,607 бит

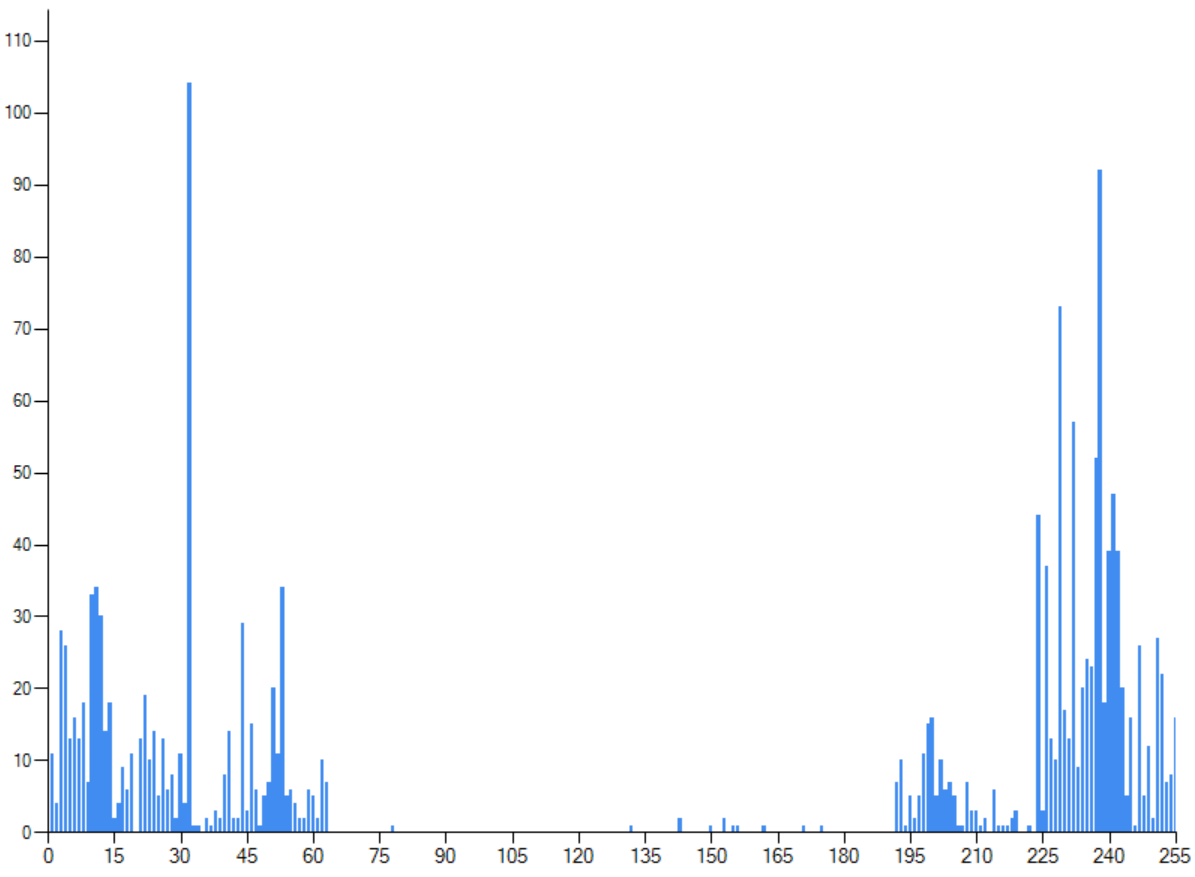


Рисунок 2 – Гистограмма шифротекста 1 раунда, H(x) = 6,201 бит

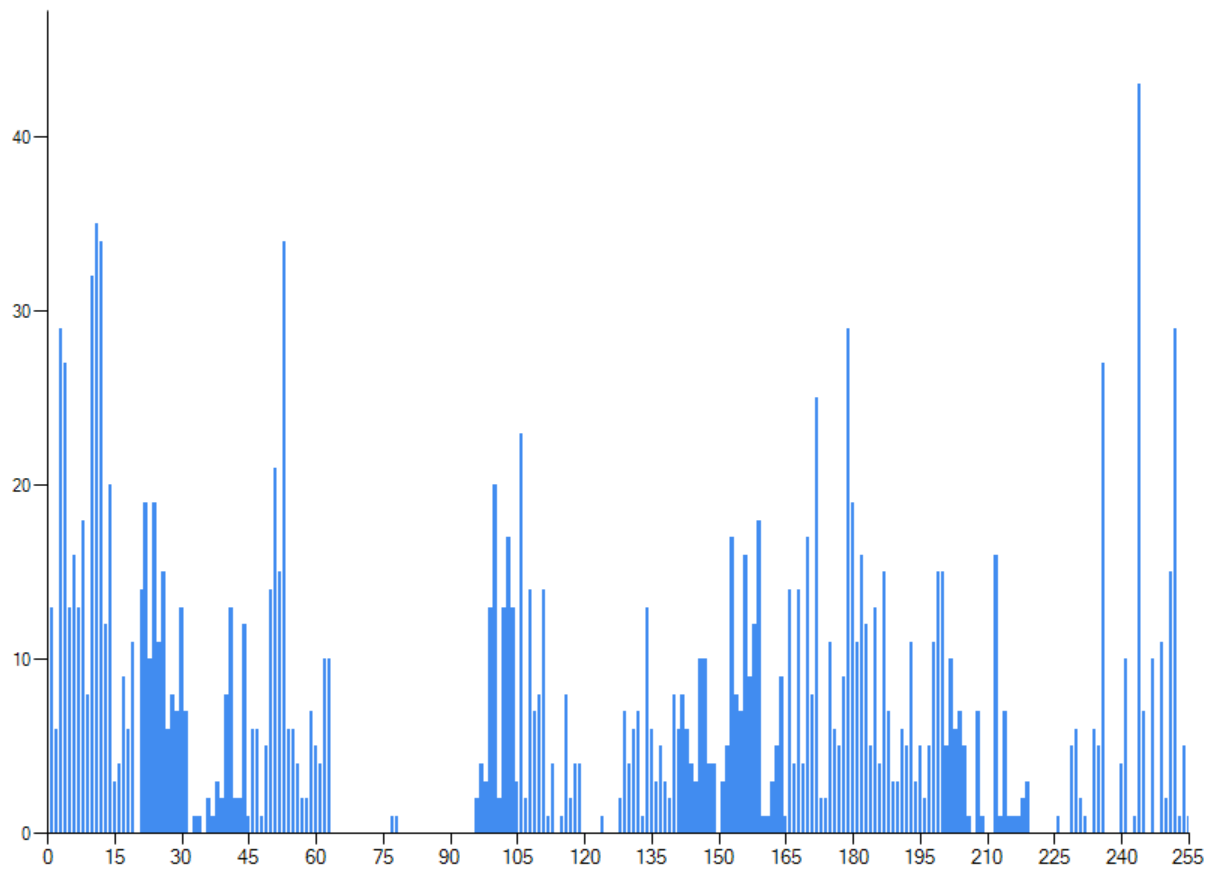


Рисунок 2 – Гистограмма шифротекста 2 раунда, H(x) = 7,115 бит

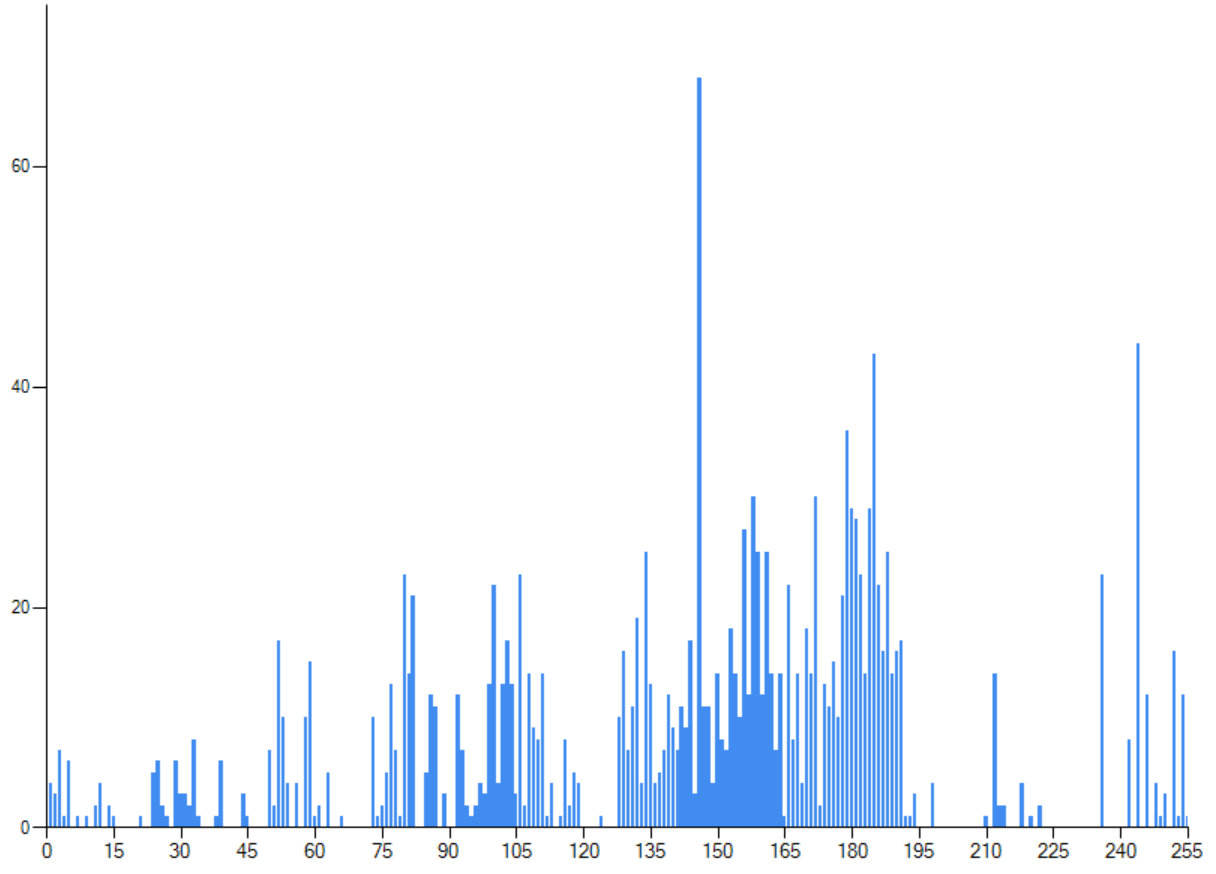


Рисунок 2 – Гистограмма шифротекста 3 раунда, H(x) = 6,806 бит

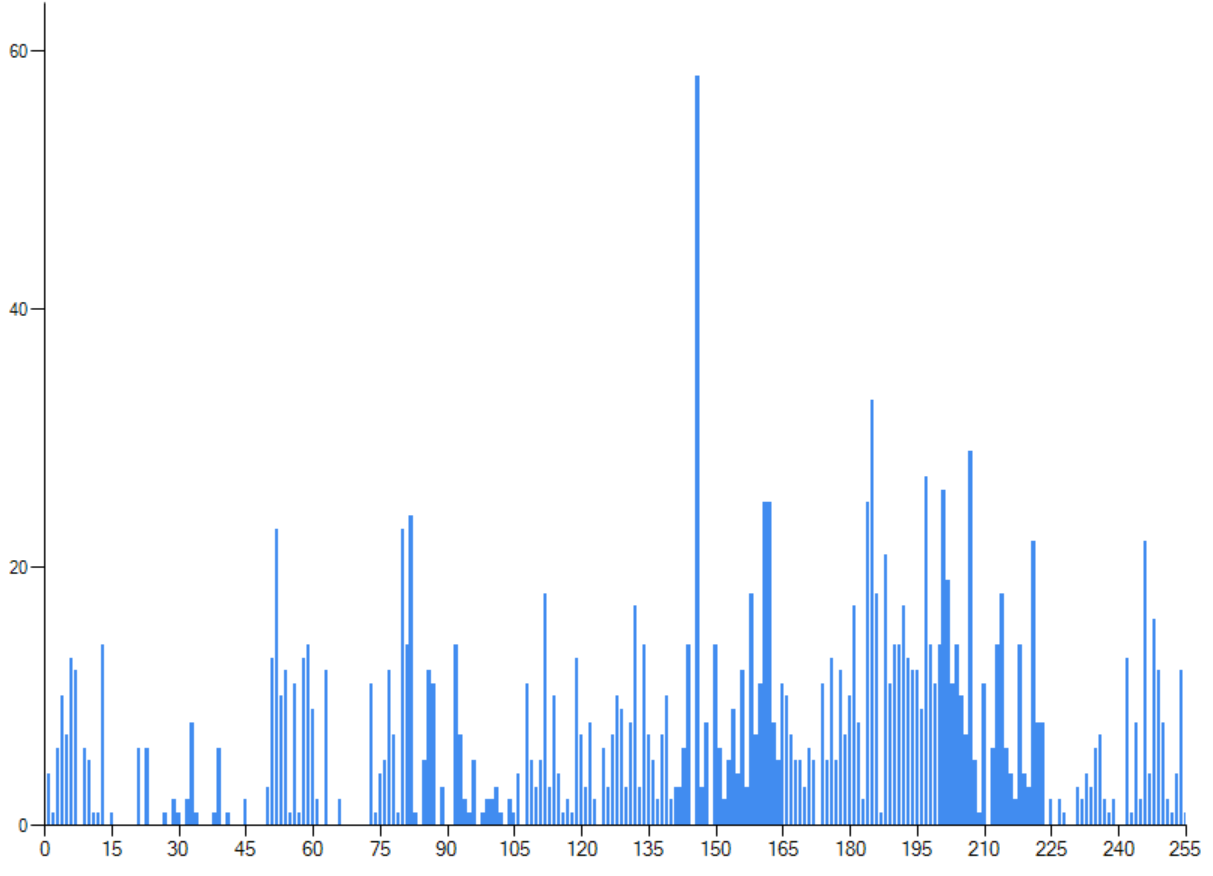


Рисунок 2 – Гистограмма шифротекста 4 раунда, H(x) = 7,185 бит

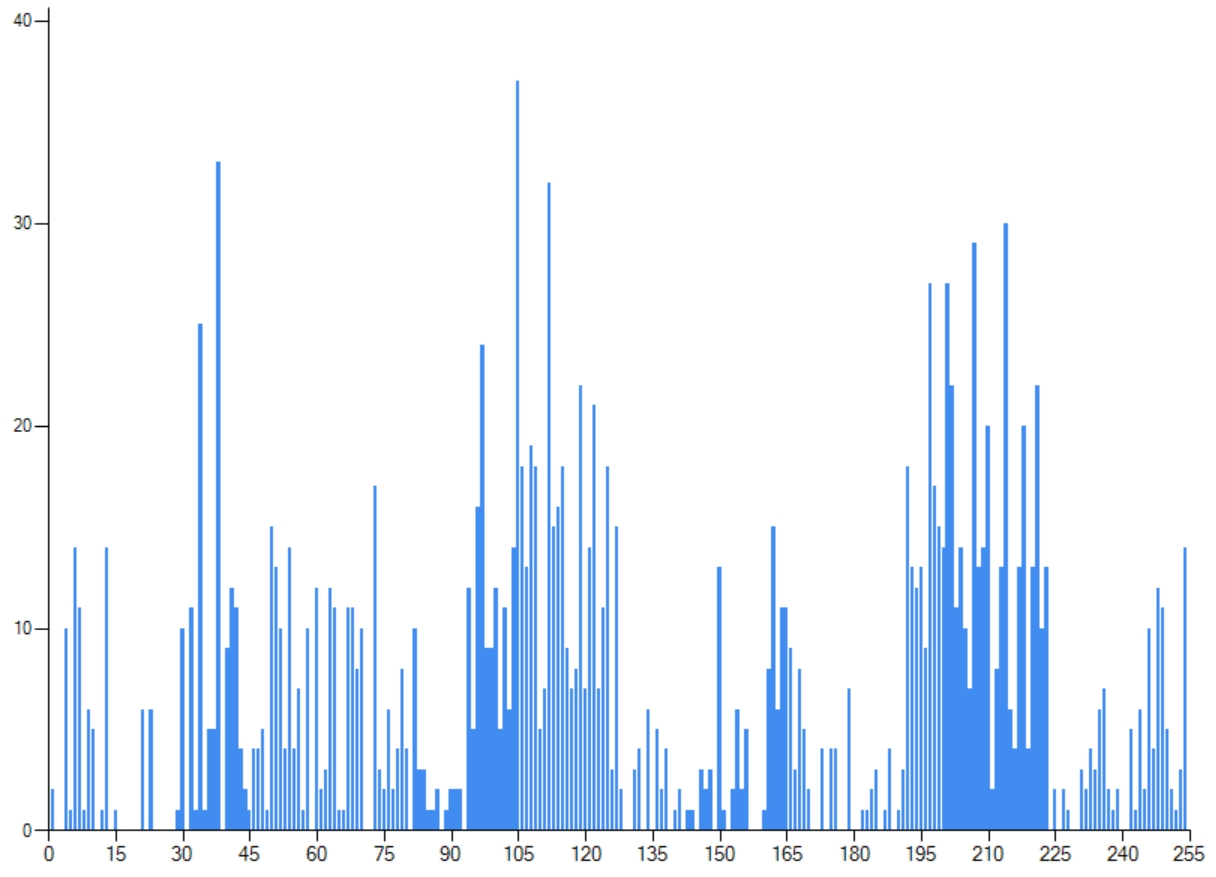


Рисунок 2 – Гистограмма шифротекста 5 раунда, H(x) = 7,183 бит

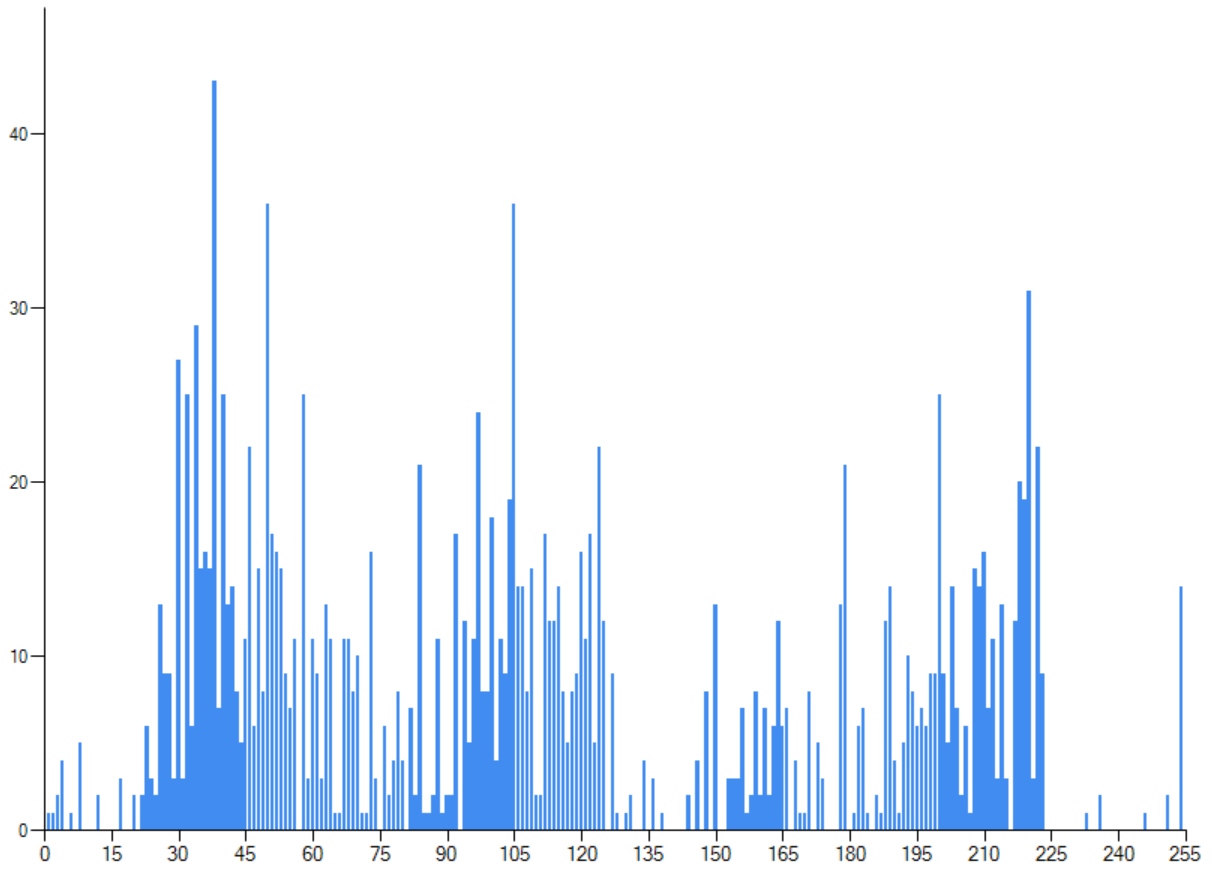


Рисунок 2 – Гистограмма шифротекста 6 раунда, H(x) = 7,075 бит

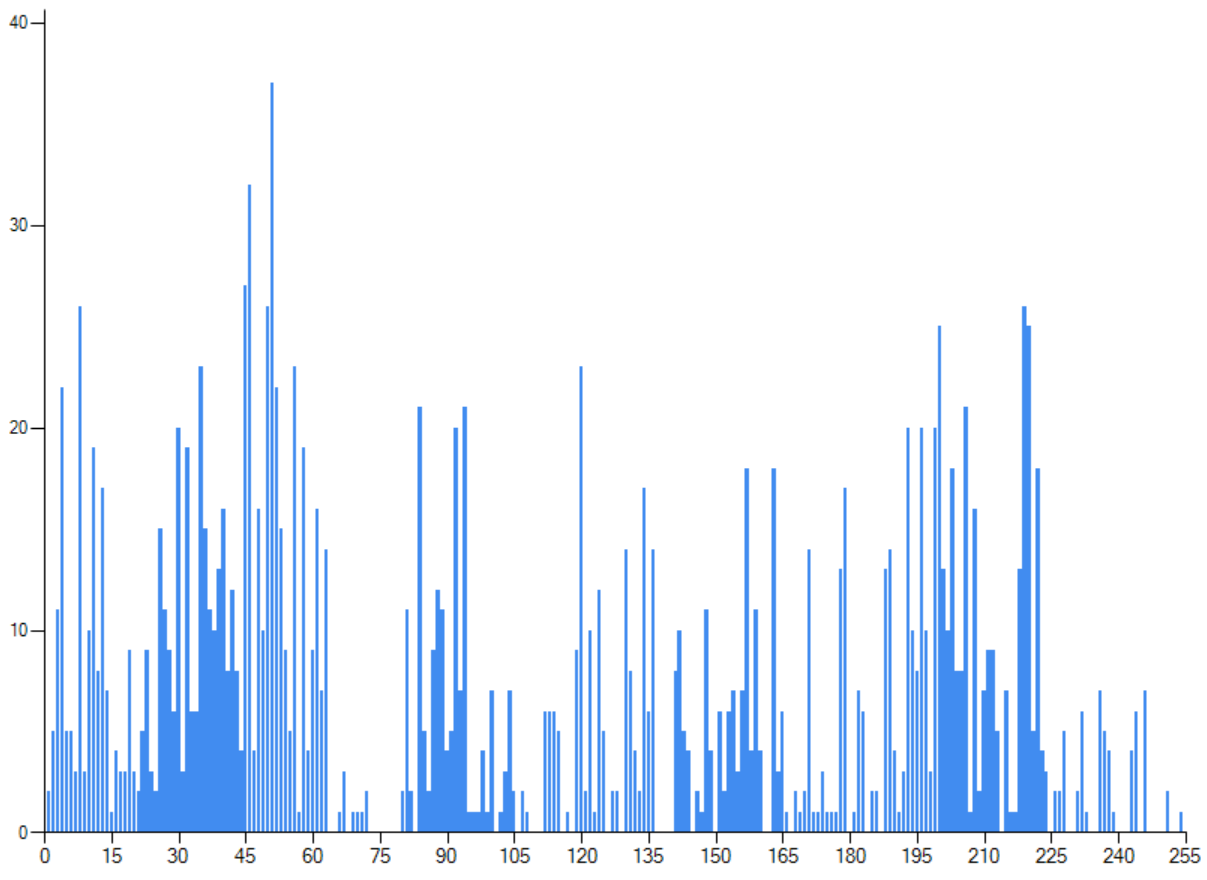


Рисунок 2 – Гистограмма шифротекста 7 раунда, H(x) = 7,175 бит

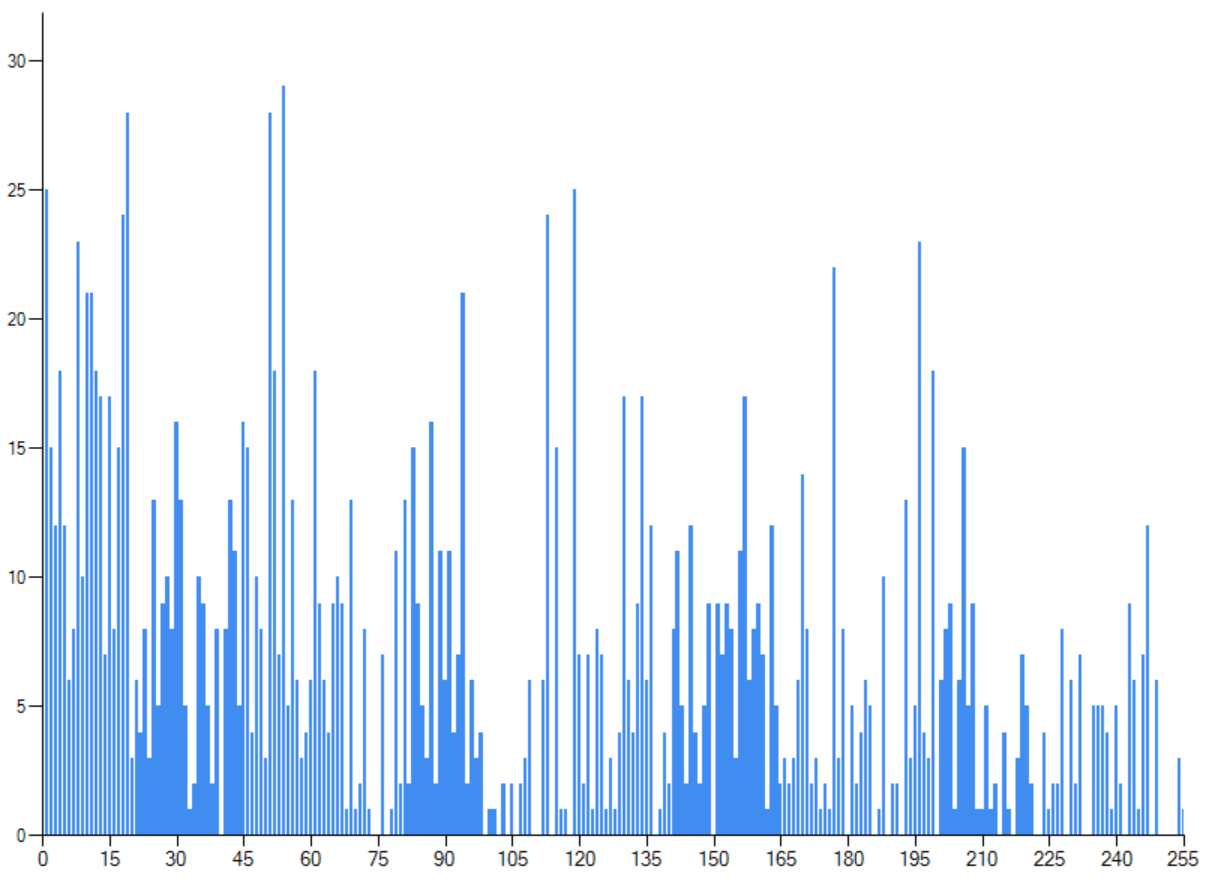


Рисунок 2 – Гистограмма шифротекста 8 раунда, H(x) = 7,369 бит

# Приложение А

Открытый текст:

То, что в жизни, наступившей ныне, императоров, видимо за отсутствием империи, уже нет, Тереховым не оценивается вообще никак. Сочинение его, притом что оно сплошь и рядом населено политиками и спецслужбистами, совершенно безоценочно. Вернее сказать, у Терехова, как и у всех вышеназванных, ощущение времени возникает не посредством позитивно или негативно окрашенного публицистического высказывания, но через атмосферу текста.

И атмосфера эта в той или иной степени определяется словом «морок». У Терехова замороченность всего происходящего чувствуется особенно остро, в меньшей степени это есть и у Юзефовича. Аномально поведение главного героя у Самсонова – что, собственно, отражено уже в названии романа. Аномальна, на наш вкус, и главная сюжетная коллизия «Цены отсечения» Архангельского. То есть времена срослись, но, как ни странно, полегчало ненадолго. После прочтения каждой из этих книг остается горький осадок. Горчит по-разному – но горчит неизменно.

И не только по причине невольного погружения в эту выморочную, аномальную, нездоровую атмосферу. Другой особенностью всех четырех книг является острое, порой даже патологическое, мучительное внимание к межполовым отношениям. У Архангельского это чувствуется чуть менее (хотя, пожалуй, он в этом вопросе циничней всех). У Самсонова, у Юзефовича и в особенности у Терехова сексуальные мотивы являются едва ли не навязчивыми.

Самсонов сохранил хоть какие-то признаки идеализма при взгляде на мужчину и женщину, Юзефович – почти уже нет, зато он самый, что ли, снисходительный из числа рассматриваемых нами. В любом случае брезгливые, а иногда и злые описания греховной человеческой породы просматриваются у каждого.

Шифротект, полученнный на 8 раунде:

䖝䁢བྷ孃㜇㘍꧅珄ጐꥸⰐ歫䲏珄䉡꫚ǁ㬊䊎ᄓ侇䖝⬛絽땳矆ř᜽꡹浺ጩ讔՝⤅︭녱꧅࿊膚珄Ḅ䖝ᔶ䏬㸠र՝쐣燇ᄪਮ它Ḅ䢍놣㜉3ɞਮᏛᎰ㴠⬌ళ讔벂䖝䲈饛ἳ埶ግ燇㘈叐㘋꽿闭ᥑ䞎㴋՝ꪆ芙㘈矎㴄饛릃䆍ᥑᥑ꡽纨롱ᰪ嚚ሸ䆍꧅叐䊎ᬤ㐋བྷ惗㠆ᄪ㘈꙽량㄃䊎쐣ɞ뉰㜆ᄓ࿊ᆳ䊎粫䇍燋ㄋǁ࿊놣鋮रᬤꪆ烢㌄ᰵ髊燇1ᠯਮɞ㬓⨚ళᥑ̽㘍릃뎜ꯓ쐣矎酞䏬쾬ጩἳ矎ᴐ饛狼㘈ǁ4珄ꒅᄓ狼㘈燇㌋ıళ㘈벂矎ᨭ4ꝷሸἾᰇ벂놣ስᏛḳ它ਮⲼ珄བྷ䏬⑐ᄓ땳Ḿ䊎飤㔖蕒⬓浪⏃셙ଧ酞彠ᄩ叴⬓놣凓䆆ǁ4ǁ㘈麚ሸ퐁ళᥑᴸ䏬࿐ᨳḄ䢍叐㐊̽՝侇ܕ켪ᆳ䊎粫ř꒐땳ꅷ㐋酞⬗놣벂矎ꅷ魘㸠팂ᠯ벂埶ሸᤈḄ䢍燇ሌ惗㐋བྷ酞쬃鿳놣Ǎళ┅䲏叐㔖︭ਮ叴ግሸ闭ᥑ飤궠㨒Ἶ뇇膹㘈薝ꁸ㘈矎媂ళ讔콚ნ┞ᷜབྷ竑ꅷ鳨飤옂靔㤊ř枏䁢鿳叐ሸꖆ㐋鿳燇쬃놟ళབྷᰪɞ孃ꪆ烢㌄㤉ǉἳ㌄뇇膹㘈薝ꁸꪆ䖝ꪆ珄ı놣놀⺶ǁ콚ნ┞᜾叐Ḅꪆ鳨⨏㼎ἶ＂ㄊᰵ燇浺ᐹళἶꁬ慩ሮ᳝␝잤벂侗ሸ闭㬓⨚ళřز酞4了䖝㘈䖝Ꮫ鎛ਮ叐쬃㌄㘈䖝『芙ሳ뾂བྷ쐣ꪆ㴋誗량矮酞芙矎ꅷ䁢ళἶ燋놟㴋麚㼎叴ᬋ鋮㴆ߒ燇ሸ뎜浺鳨벂ㄍ쌃麚酞ꆅܕ侗屢䞆饛矎맊ᄩ㘍ἳ飤枏㘈콚侇咈ꯓ쐣㼎ވ㐋량㦱쐣凓㴄ᔿབྷ『矎녱咈쐣珄뺔㴄⟠ᷜབྷ『䢄ᰪళἶ놣叴ꥸ놣း侇㴠㰍⿃셙㘈薝ꁸ㰍⿃珄㐉㴋ꅷᤈ鿧ᄓ땳Ḿ䊎氲淕䋯䆆燇ሌ惗㐋酞⬗鿧綪粫埶Ḅ<粫燋ਮࡔᨳḄ䢍燇ሸꅷ꫚矎侇䆆놟ɞ貘ግ㐊墜뉰馒䖝䎞咈㔖ꁸ㘈ɞ┞⤅㌄讔燋ǍỘ『㐊र՝᜽㘈ᬤਮ걼侗룄慩ሮ᳝␝잤벂叴ግ埶ሸᤈḄ䢍埶쒦闭ވ쐣ɞ௄᜽Ⲽ矎육뾂ꩵ䖝㤊놣叴ግ՝̽䲈酞ɞꝷਮ꙽䲏놣簿侗룄옂靔㤊䇍࿊鄗굻⬛瓠ǁ놣燇ሌ惗㐋酞ᬋ࿊ᆳ䊎粫ř蕒ᠤꁸᰵɞ땳ᬋर埾준䂎浺鳨媂它翂ᥑ쐣珄燣ᘿ咈रᔅ浪攒䆍ሌ㘈孃ሌሾ㌄㜊双᜽闭Ꮫ쌃賳ግ矎녱㘍ㄋᥑ্芞4䒄ř䲈ᥑ䦄㈇汤ɞꪆǍᰪ咈㰍ǁళ꺂㰍䖝궠㨒Ἶᆯ矎㴠ᬋ࿊膚珄Ḅ䖝꒐ግ燇⩆☢ጆ뛎埶ግ翂꧅叐ਮ灷Ί饛␝⟠ꉽǁ⑐咈롱řᄩျ薝쬃狼뎜ጆ鋮ꪆᔅ侗믣ᐺȼ՝롱숃뎜뎠䏬躛4र꫚셙ǁ㘈뢘řᥑ뺃ሩ燇Ⱀ☢ਮ鳨䢄粫㘈麚㐊ꖆ뉰䆆矎侇Ί怬䲈酞薝쬃狼ᄷ浺鳨飤Ꮫ鞓ᷜ량

# Приложение Б

Листинг файла MZI\_4.cpp

#include <iostream>

#include <string>

#include <fstream>

#include <Windows.h>

#include "Feistel.h"

const std::string outFile = "data";

const std::string reset = "\033[0m";

const std::string red = "\033[31m";

const std::string green = "\033[32m";

std::string cipher(char mode, std::string data);

std::string get\_data();

void save(std::string data);

std::string key = "Пустой колос голову кверху носит";

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char choice = 0;

while (choice != '5') {

system("cls");

std::cout << "1. Зашифровать\n2. Расшифровать\n3. Изменить ключ\n4. Вывести текущий ключ\n5. Выход\n>> ";

std::cin >> choice;

system("cls");

switch (choice)

{

case '1': {

std::string data = get\_data();

if (data == "")

break;

std::cout << red << "\nИсходный текст:\n" << reset << data;

std::string result = cipher(0, data);

std::cout << green << "\n\nЗашифрованный текст:\n" << reset << result << std::endl << std::endl;

save(result);

break;

}

case '2': {

std::string data = get\_data();

if (data == "")

break;

std::cout << red << "\nИсходный текст:\n" << reset << data;

std::string result = cipher(1, data);

std::cout << green << "\n\nРасшифрованный текст:\n" << reset << result << std::endl << std::endl;

save(result);

break;

}

case '3': {

std::cout << "Текущий ключ: " << key << "\n\nВведине новый ключ\n>> ";

std::cin.ignore();

getline(std::cin, key);

break;

}

case '4': {

std::cout << "Текущий ключ: " << key << "\n\n";

system("pause");

break;

}

}

}

}

std::string cipher(char mode, std::string data) {

Feistel Feistel;

std::vector<std::string> result\_v;

if(!mode)

result\_v = Feistel.encode(data, key);

else

result\_v = Feistel.decode(data, key);

for (int i = 0; i < result\_v.size(); i++) {

std::ofstream out(outFile + "\_" + std::to\_string(i + 1) + "\_round", std::ios::binary);

for (char it : result\_v[i]) {

out.write(&it, sizeof(char));

}

out.close();

}

return result\_v.back();

}

std::string get\_data() {

std::cout << "Введите путь к файлу\n>> ";

std::string filepath;

std::cin >> filepath;

std::string data;

if (filepath == "-m") {

std::cin.ignore();

std::cout << "\nВведите текст вручную\n>> ";

getline(std::cin, data);

return data;

}

std::ifstream in(filepath, std::ios::binary);

if (!in.is\_open()) {

std::cout << red << "\nНе удалось открыть файл (неверный путь)!\n\n" << reset;

system("pause");

return std::string();

}

char temp = 0;

while (in.read(&temp, sizeof(char))) {

data += temp;

}

in.close();

return data;

}

void save(std::string data) {

std::cout << "Сохранить полученный текст в файл " << outFile << "? (Y/N, m для ручного выбора файла)\n>> ";

char ch;

std::cin >> ch;

if (std::toupper(ch) == 'Y') {

std::ofstream out(outFile, std::ios::binary);

for (auto it : data) {

out.write(&it, sizeof(char));

}

out.close();

system("cls");

std::cout << green << "Данные были успешно сохранены в " << outFile << "\n\n" << reset;

system("pause");

}

else if (ch == 'm') {

std::string filepath;

std::cout << "\nВведите путь к файлу для сохранения\n>> ";

std::cin.ignore();

getline(std::cin, filepath);

std::ofstream out(filepath, std::ios::binary);

for (auto it : data) {

out.write(&it, sizeof(char));

}

out.close();

system("cls");

std::cout << green << "Данные были успешно сохранены в " << filepath << "\n\n" << reset;

system("pause");

}

}

Листинг файла Feistel.h

#pragma once

#include <bitset>

#include <vector>

#include <string>

#include <algorithm>

class Feistel {

public:

std::vector<std::string> encode(std::string data, std::string key) {

return cipher(data, generate\_keys(key));

}

std::vector<std::string> decode(std::string data, std::string key) {

std::vector<unsigned char> keys = generate\_keys(key);

std::reverse(keys.begin(), keys.end());

return cipher(data, keys);

}

private:

std::vector<std::string> cipher(std::string data, std::vector<unsigned char> keys) {

std::vector<std::string> result;

if (data.size() % 2 == 1)

data += 'x';

for(int i = 0; i < 7; i++) {

result.push\_back(round(data, keys[i]));

data = result.back();

}

result.push\_back(last\_round(data, keys[7]));

data = result.back();

return result;

}

std::string round(std::string data, unsigned char key) {

for (size\_t i = 0; i < data.size(); i += 2) {

unsigned char a = (unsigned char)data[i];

unsigned char b = (unsigned char)data[i + 1];

unsigned char temp = b;

b = F(b, key) ^ a;

a = temp;

data[i] = (char)a;

data[i + 1] = (char)b;

}

return data;

}

std::string last\_round(std::string data, unsigned char key) {

for (size\_t i = 0; i < data.size(); i += 2) {

unsigned char a = (unsigned char)data[i];

unsigned char b = (unsigned char)data[i + 1];

unsigned char temp = b;

b = F(b, key) ^ a;

a = temp;

data[i] = (char)b;

data[i + 1] = (char)a;

}

return data;

}

unsigned char F(int byte, int key) {

unsigned char res = (byte + key) % 256; // Наложение раундового ключа по модулю алфавита

res = (res << 1) | (res >> (8 - 1)); // Циклический сдвиг

return res;

}

std::vector<unsigned char> generate\_keys(std::string str) {

if (str.size() % 2 == 1)

str += 'x';

std::vector<unsigned char> keys;

for (int i = 0; i < 8; i++) {

unsigned char key = 0;

for (int j = 0; j < str.size(); j += 2) {

std::swap(str[j], str[j + 1]);

}

for (int j = 0; j < str.size(); j += 2) {

key = (key + (unsigned char)str[j]) % 256;

}

str = str.substr(1) + (char)key;

keys.push\_back(key);

std::reverse(str.begin(), str.end());

}

return keys;

}

};