Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОССУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе №2

По дисциплине «Методы защиты информации»

По теме «Симметричная криптография. СТБ 34.101.31-2011»

Выполнил:

студент гр. 653502

Максименко П.В.

Проверил:

Артемьев В. С.

Минск 2019

1. Постановка зада4и

Написать программу, реализующую разли4ные режимы работы алгоритма СТБ 34.101.31-2011. Для этого были выбраны «шифрование режимом простой замены» и «[режим сцепления блоков шифротекста](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B6%D0%B8%D0%BC_%D1%81%D1%86%D0%B5%D0%BF%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%BB%D0%BE%D0%BA%D0%BE%D0%B2_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0)».

2. Теорети4еская 4асть

Electronic Codebook (ECB)

В [ГОСТ 28147—89](https://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2_28147%E2%80%9489) этот режим называется режимом простой замены.

Пусть дано сообщение {\displaystyle P}([открытый текст](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82), последовательность бит, данные).

Шифрование в режиме ECB (режиме электронной кодовой книги)

1. Сообщение делится на блоки одинакового размера. Размер (длина) блока равен n и измеряется в [битах](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82). Последний блок при необходимости дополняется до длины {\displaystyle n}n.
2. Каждый блок {\displaystyle P\_{i}} шифруется алгоритмом шифрования {\displaystyle E\_{k}} с использованием ключа k:

Cipher Block Chaining (CBC)

Шифрование в режиме CBC (режиме сцепления блоков шифротекста)

* [Сообщение](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82) разбивается на блоки одинакового размера. Размер (длина) блока равен *n* и измеряется в [битах](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82). При необходимости последний блок дополняется до длины {\displaystyle n}n.
* Шифрование очередного (*i*-го) блока [сообщения](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82) {\displaystyle P\_{i}}выполняется с использованием предыдущего [зашифрованного](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82) (*(i-1)*-го) блока. Для первого блока зашифрованного блока не существует, поэтому первый блок шифруют с использованием «вектора инициализации»

3. Блок-схема

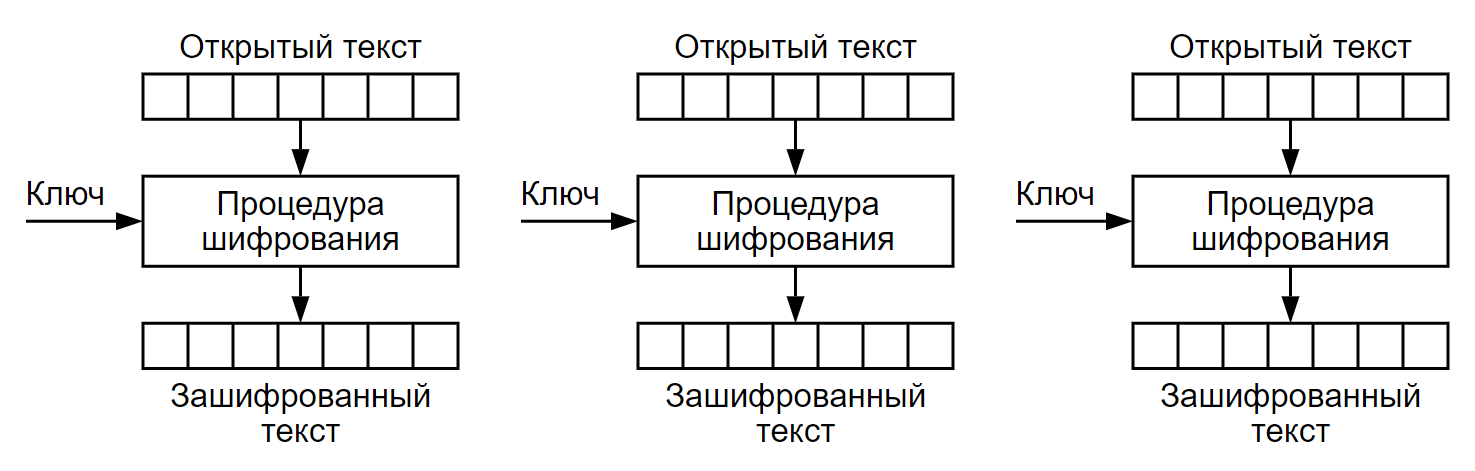


Рис 3.1. Блок-схема работы ECB алгоритма.

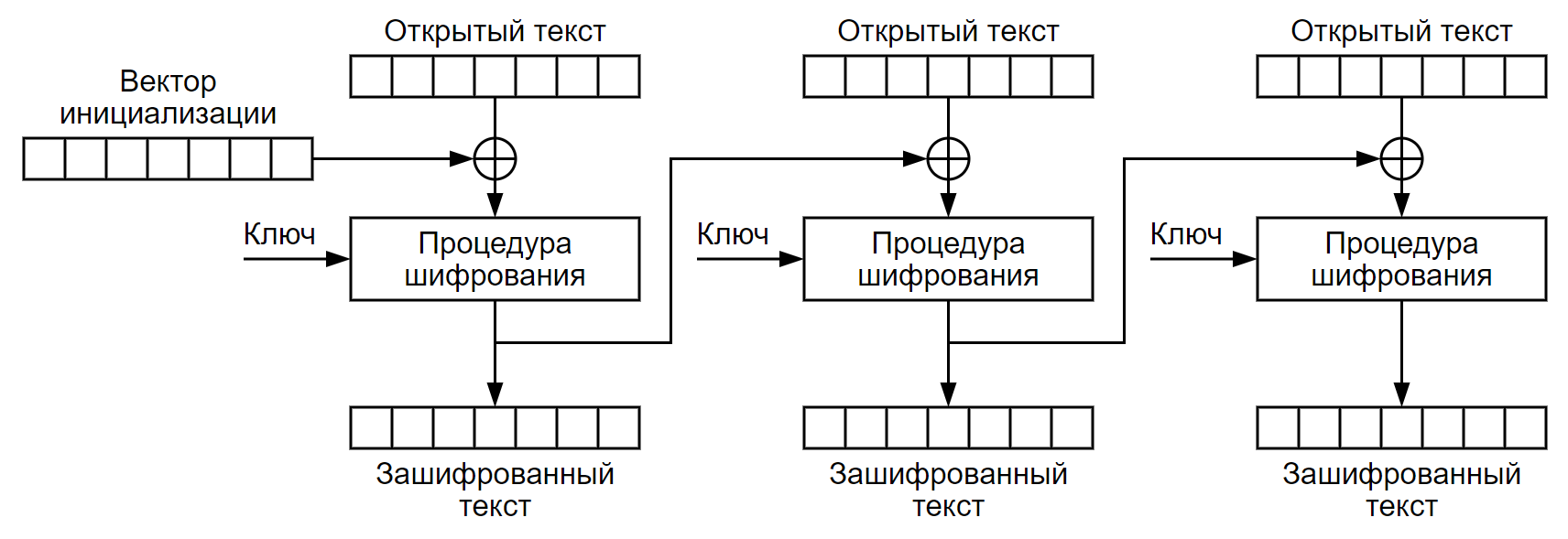


Рис 3.2. Блок-схема работы CBC алгоритма.

4. Ход программы



Рис 4.1. Данные, поступающие на вход



Рис 4.2. Данные после обработки CBC алгоритмом



Рис 4.3. Данные после обработки CBC алгоритмом

5. Код программы

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | |  | #include <iostream> | |  | #include <fstream> | |  | #include <string> | |  |  | |  | using namespace std; | |  |  | |  | std::string ECB(string inp\_data, string inp\_key) | |  | { | |  | string key = inp\_key; | |  | string data\_help = inp\_data; | |  | std::cout << '\n'; | |  | std::cout << data\_help; | |  |  | |  | string data = data\_help; | |  | // breack up into 4 16b blocks | |  | for (int i = 0; i < 4; i++) { | |  | for (int j = 0; j < 16; j++) { | |  | // encrypt data by key | |  | if (key[i \* 16 + j] == '1') { | |  | if (j % 2 == 0) { | |  | data[i \* 16 + j] = data\_help[i \* 16 + j]; | |  | } | |  | else | |  | { | |  | if (data\_help[i \* 16 + j] == '1') | |  | { | |  | data[i \* 16 + j] = '0'; | |  | } | |  | else if (data\_help[i \* 16 + j] == '0') | |  | { | |  | data[i \* 16 + j] = '1'; | |  | } | |  | } | |  | } | |  | else | |  | { | |  | data[i \* 16 + j] = '0'; | |  | } | |  | } | |  | } | |  | std::cout << '\n'; | |  | std::cout << data; | |  |  | |  | return data; | |  | } | |  |  | |  |  | |  | std::string CBC(string inp\_data, string inp\_key, string start\_vector) | |  | { | |  | string key = inp\_key; | |  | string data\_help = inp\_data; | |  | std::cout << '\n'; | |  | std::cout << data\_help; | |  |  | |  | string data = data\_help; | |  | // breack up into 4 16b blocks | |  | for (int i = 0; i < 4; i++) { | |  | for (int j = 0; j < 16; j++) { | |  | // encrypt data by key | |  | if (key[i \* 16 + j] == '1') { | |  | if (j % 2 == 0) { | |  | data[i \* 16 + j] = data\_help[i \* 16 + j]; | |  | } | |  | else | |  | { | |  | if (data\_help[i \* 16 + j] == '1') | |  | { | |  | data[i \* 16 + j] = '0'; | |  | } | |  | else if (data\_help[i \* 16 + j] == '0') | |  | { | |  | data[i \* 16 + j] = '1'; | |  | } | |  | } | |  | } | |  | else | |  | { | |  | data[i \* 16 + j] = '0'; | |  | } | |  | } | |  | // encrypt by prev result | |  | if (i == 0) | |  | { | |  | data[0] = start\_vector[0]; | |  | data[15] = start\_vector[7]; | |  | } | |  | else if (i != 0) | |  | { | |  | for (int j = 0; j < 16; j++) { | |  | data[i \* 16 + j] = data[(i - 1) \* 16 + j]; | |  | data[i \* 16 + j] = data[(i - 1) \* 16 + j]; | |  | } | |  | } | |  | } | |  |  | |  | std::cout << '\n'; | |  | std::cout << data; | |  |  | |  | return data; | |  | } | |  |  | |  |  | |  | int main() | |  | { | |  | SetConsoleCP(1251); | |  | SetConsoleOutputCP(1251); | |  |  | |  | string wdata; | |  | string rdata; | |  | rdata = read\_input\_file(); | |  |  | |  | // lab2 | |  | // simple replacement mode | |  | string key = "1001100101101110111100101000011111101001101111101000011011011110"; | |  | wdata = ECB(rdata, key); | |  |  | |  | // ciphertext block coupling mode | |  | string start\_vector = "01100110"; | |  | wdata = CBC(rdata, key, start\_vector); | |  |  | |  | return 0; | |  | } | |  |

6. Вывод

В ходе выполнения лабораторной работы были успешно реализованы алгоритмы СТБ 34.101.31-2011 шифрования в режимах простой замены и сцеплением блоков. Из преимуществ алгоритмов стоит отметить возможность распараллеливания работы ECB, а также постоянную скорость обработки блоков данных CBC алгоритма. Недостаток обоих алгоритмов – невысокая криптонадёжность.