Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование потоковых шифров

Студент: Голодок А. Ю.

ФИТ 3 курс 4 группа

Преподаватель: Сазонова Д. В.

Минск 2024

1. **Теоретическая часть**

Потоковый шифр - это метод симметричного шифрования, где каждый символ открытого текста преобразуется в символ шифртекста на основе ключа и позиции символа в тексте. Он работает, представляя открытый текст символ за символом или бит за битом.

Потоковые шифры делятся на два класса: синхронные и асинхронные. Синхронные генерируют поток ключей независимо от текста, что обеспечивает нераспространение ошибок, но делает их уязвимыми к атакам на отдельные биты шифртекста. Асинхронные зависят от секретного ключа и предыдущих знаков шифртекста, что может привести к распространению ошибок.

Генераторы ключевых последовательностей, также известные как генераторы случайных или псевдослучайных чисел, создают последовательности чисел для систем потокового шифрования. Стандарт СТБ 34.101.47–2017 в Беларуси определяет ключевые термины в этой области. Случайные числа - это последовательности, где каждый элемент невозможно предсказать на основе предыдущих. Псевдослучайные числа - это последовательности, полученные из алгоритма и используемые вместо случайных чисел.

Часто используемый алгоритм генерирования ПСП реализуется на основе так называемого линейного конгруэнтного генератора, описываемого следующим рекуррентным соотношением:  


где и – соответственно *t*-й (предыдущий) и (*t* + 1)-й (текущий, вычисляемый) члены числовой последовательности; *а*, *с* и *n* – константы. Период такого генератора (период ПСП) не превышает *n*.

Генератор же ПСП на основе RSA устроен следующим образом. Последовательность генерируется с использованием соотношения



Начальными параметрами служат *n*, большие простые числа *p* и *q* (причем *n* = *pq*), целое число е, взаимно простое с произведением (*р* – 1)(*q* – 1), а также некоторое случайное начальное значение .

Выходом генератора на *t*-м шаге является младший бит числа .

Безопасность генератора опирается на сложность взлома алгоритма RSA, т. е. на разложение числа *n* на простые сомножители.

1. **Практическая часть**

**2.1 RC4**

Для реализации алгоритма RC4 был разработан класс с соответствующим названием, код представлен на рисунке 2.1.

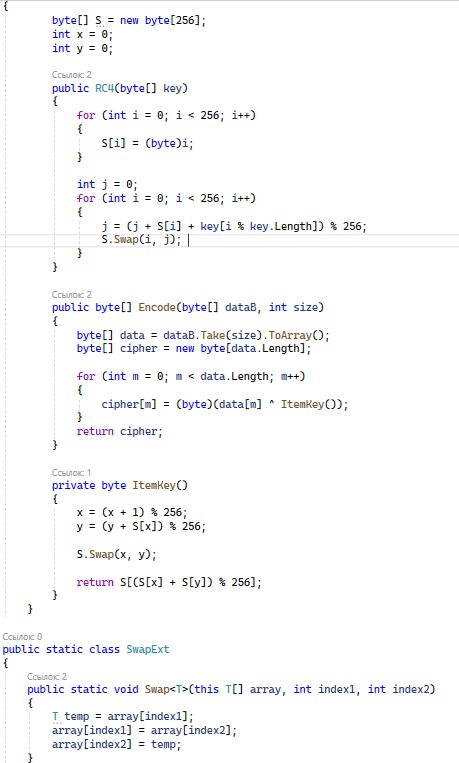


Рисунок 2.1 – реализация RC4

Класс RC4 содержит поля для массива S размером 256 байт и переменных x и y, а также конструктор и методы для шифрования данных и генерации ключа.

В конструкторе класса RC4 происходит инициализация массива S и перестановка его элементов на основе ключа. Это достигается путем выполнения псевдослучайной перестановки элементов массива S в соответствии с ключом.

Метод Encode принимает данные и размер входного массива, затем каждый байт данных XOR'ится с соответствующим байтом, сгенерированным методом ItemKey(), и результат записывается в массив шифротекста.

Метод ItemKey обновляет значения переменных x и y, а затем осуществляет перестановку элементов массива S и возвращает байт, используемый в процессе шифрования.

Класс SwapExt содержит статический метод Swap, который выполняет обмен значений двух элементов в массиве. Этот метод используется в классе RC4 для перестановки элементов массива S.

**2.2 BBS**

Для реализации генерации псевдослучайной последовательности (ПСП) на основе алгоритма BBS была разработана функция, представленная на рисунке 2.2.

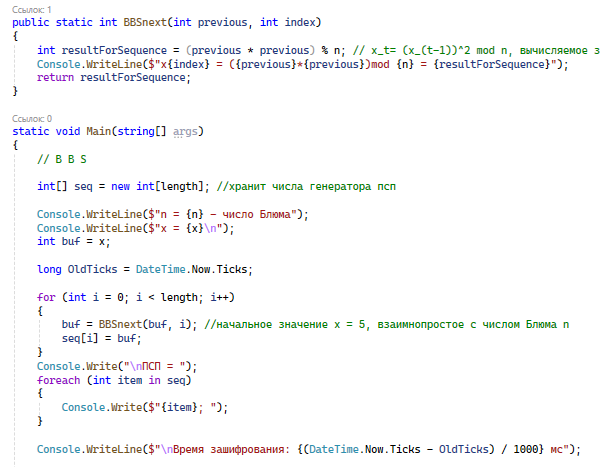


Рисунок 2.2 – реализация BBS

Переменные n, x и length определяют параметры алгоритма: n - число Блюма (253, произведение простых чисел 11 и 23), x - начальное значение генератора, length - количество генерируемых чисел в последовательности.

Метод BBSnext вычисляет следующее значение последовательности BBS на основе предыдущего значения. Он использует формулу (previous \* previous) % n для вычисления нового значения генератора.

В методе Main происходит инициализация и заполнение массива seq с помощью метода BBSnext. Затем выводится полученная последовательность псевдослучайных чисел.