МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность Информационные системы и технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7 НА ТЕМУ:

Исследование ассиметричных шифров

Выполнила:

Студентка 3 курса 1 группы ФИТ

Шимчёнок Елизавета Константиновна

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации ассиметричных шифров.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости асимметричных шифров.
2. Разработать приложение для реализации указанных преподавателем методов генерации ключевой информации и ее использования для асимметричного зашифрования/расшифрования.
3. Выполнить анализ криптостойкости асимметричных шифров.
4. Оценить скорость зашифрования/расшифрования реализованных шифров.
5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Теоретические сведения**

В основу асимметричной криптографии положена идея использовать ключи парами: один – для зашифрования (открытый, или публичный, ключ), другой – для расшифрования (тайный ключ). Отметим, что указанная пара ключей принадлежит получателю зашифрованного сообщения.

Все алгоритмы шифрования с открытым ключом основаны на использовании односторонних функций, к числу которых, как известно, относится вычисление дискретного логарифма.

Односторонней функцией называется математическая функция, которую относительно легко вычислить, но трудно найти по значению функции соответствующее значение аргумента, т. е. зная х, легко вычислить *f*(*x*), но по известному *f*(*x*) трудно найти подходящее значение *x*.

Алгоритмы шифрования с открытым ключом можно использовать для решения следующих задач:

- зашифрования/расшифрования передаваемых и хранимых данных в целях их защиты от несанкционированного доступа;

- формирования цифровой подписи под электронными документами;

- распределения секретных ключей, используемых далее при шифровании документов симметричными методами.

Ранцевый (рюкзачный) вектор *S*= (*s*1, ..., *sz*) – это упорядоченный набор из *z*, *z* ≥ 3, различных натуральных чисел *si*. Входом задачи о ранце (рюкзаке) называем пару (***S***, *S*), где *S* – рюкзачный вектор, а *S* – натуральное число.

Решением для входа (***S***, *S*) будет такое подмножество из ***S***, сумма элементов которого равняется *S*.

Суть методадля шифрования состоит в том, что существуют две различные задачи укладки ранца: одна из них решается легко и характеризуется линейным ростом трудоемкости, а другая решается трудно. Легкий для укладки ранец можно трансформировать в трудный.

Трудный для укладки ранец применяется в качестве открытого ключа, который легко использовать для зашифрования, но невозможно – для расшифрования. В качестве закрытого ключа применяется легкий для укладки ранец, который предоставляет простой способ расшифрования сообщения.

Открытый ключ *e* представляет собой нормальную (не сверхвозрастающую) последовательность. Он формируется на основе закрытого ключа и не позволяет легко решить задачу об укладке ранца.

Для получения открытого ключа все значения закрытого ключа умножаются на некоторое число *a* по модулю *n*:

Значение модуля *n* должно быть больше суммы всех чисел последовательности; кроме того, НОД (*а, n*) = 1.

Закрытый ключ представляет сверхвозрастающую последовательность (последовательность, в которой каждый последующий член больше суммы всех предыдущих).

**Ход работы**

В ходе лабораторной работы было написано приложение, позволяющее делать следующие задачи:

- генерация сверхвозрастающей последовательности (тайного ключа);

- вычисление нормальной последовательности (открытого ключа);

- зашифрование сообщения, состоящего из ФИО;

- расшифрование сообщения;

- оценка времени выполнения операций зашифрования и расшифрования.

Для реализации генерации сверхвозрастающей последовательности была разработана функция, которая вычисляет последовательно случайные числа, с условием, что каждый последующий элемент будет больше суммы всех предыдущих. Результат можно увидеть на рисунке 1.

Для вычисления нормальной последовательности была разработана функция, которая вычисляет элементы последовательности *e* = {*ei*} по формуле: *ei = di*\**a* (*mod n*), где *di* – члены тайного ключа, *n* больше суммы всех чисел последовательности, причем НОД (*a*, *n*) = 1. Результат можно увидеть на рисунке 1.

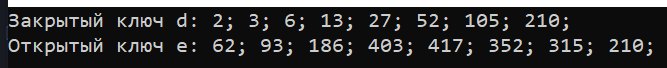


Рис. 1 – Вычисленные программой закрытый и открытый ключи

Следующим этапом было зашифрование сообщение: «Шимчёнок\_Елизавета\_Константиновна». Для шифрования сообщения каждый его символ был представлен в двоичной форме (рис. 2).

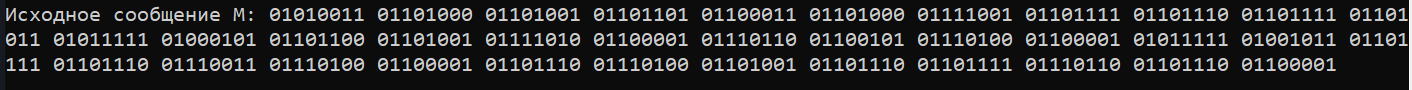


Рис. 2 – Исходное сообщение в бинарном виде

После этого была реализована последовательная проверка каждого символа на равенство единице или нулю. В первом случае соответствующий элемент открытого ключа суммировался к переменной, вычисляемой для каждого символа сообщения. Результат зашифровки изображен на рисунке 3.

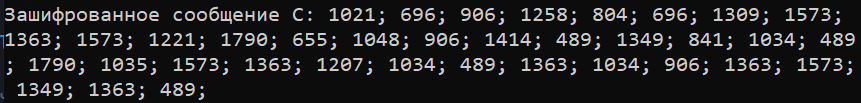


Рис. 3 – Зашифрованное программой сообщение

Для расшифрования сообщения был использован сгенерированный тайный ключ d. Алгоритм следующий:

- определить a-1, такое что *a*\**a-*1(*mod n*) *=* 1;

- каждый символ шифротекста *ci* преобразовать: *Si = ci*\**a*-1 *mod n*;

- используя вычисленное *Si* с помощью тайного ключа *d* и известного алгоритма упаковки получить расшифрованные символы *mi*.

Результат выполнения расшифровки представлен на рисунке 4.

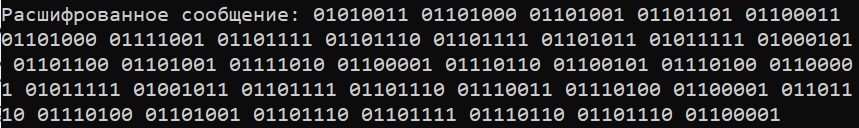


Рис. 4 – Расшифрованное сообщение в кодах ASCII

Исходное сообщение соответствует результату, который дала расшифровка. Если перевести на английский язык, результат будет: «Шимчёнок\_Елизавета\_Константиновна».

Также была оценена скорость выполнения зашифрования/расшифрования сообщения. Вычисленное время составило 40 мс и 10 мс соответственно, что является неплохим результатом. Однако с большим объемом данных обработка займет ощутимо больше времени.

Увеличение числа членов ключевой последовательности приводит к увеличению размера ключа, следовательно и к увеличению количества операций, необходимых для зашифрования/расшифрования данных.

При использовании разных таблиц кодировки (*ASCII* и *Base64*) время выполнения операций зашифрования/расшифрования может отличаться. Использование таблицы кодировки *Base64* имеет более эффективная и приводит к более быстрой работе алгоритма шифрования.

**Ответы на вопросы**

1. Что такое «ранцевый (рюкзачный) вектор»? Дать определение.

"Ранцевый вектор" или "рюкзачный вектор" – это вектор, состоящий из целочисленных значений, которые представляют вес каждого из предметов в задаче о рюкзаке.

1. Сформулировать задачу укладки ранца.

Задача укладки ранца (или задача о рюкзаке) заключается в выборе оптимального набора предметов из заданного множества, чтобы заполнить рюкзак до максимально возможного веса, не превышая его грузоподъемность.

1. Если вектор рюкзака имеет вид (14, 28, 56, 82, 90, 132, 197, 284, 341, 455), то какими следует принять коэффициенты *bi*, чтобы получить *S* = 517? Каким будет решение задачи для *S* = 516?

Решений для задачи, где *S* = 516 нет.

Чтобы получить *S* = 517, следует взять следующие коэффициенты: 197, 132, 90, 56, 28, 14.

Результат: (11101110).

1. Что такое сверхвозрастающая последовательность? Привести примеры.

Сверхвозрастающей называется последовательность, в которой каждый последующий член больше суммы всех предыдущих.

Последовательность {2, 4, 8, 16, 32, 64, 128} является сверхвозрастающей.

1. Можно ли последовательности чисел: {89, 3, 11, 2, 45, 6, 22}, {3, 41, 5, 1, 21, 10}, {2, 3, 11, 29, 45, 6, 39} преобразовать в сверхвозрастающие?

В первой последовательности сумма 2, 3 и 6 равна 11, а такой элемент существует в последовательности. Первую без замены элементов преобразовать нельзя.

Вторая последовательность является сверхвозрастающей, если ее отсортировать по возрастанию.

Третья последовательность не может быть сверхвозрастающей, по той же причине, что и первая.

1. Записать в виде псевдокода алгоритм зашифрования и алгоритм расшифрования сообщения на основе задачи об укладке ранца.

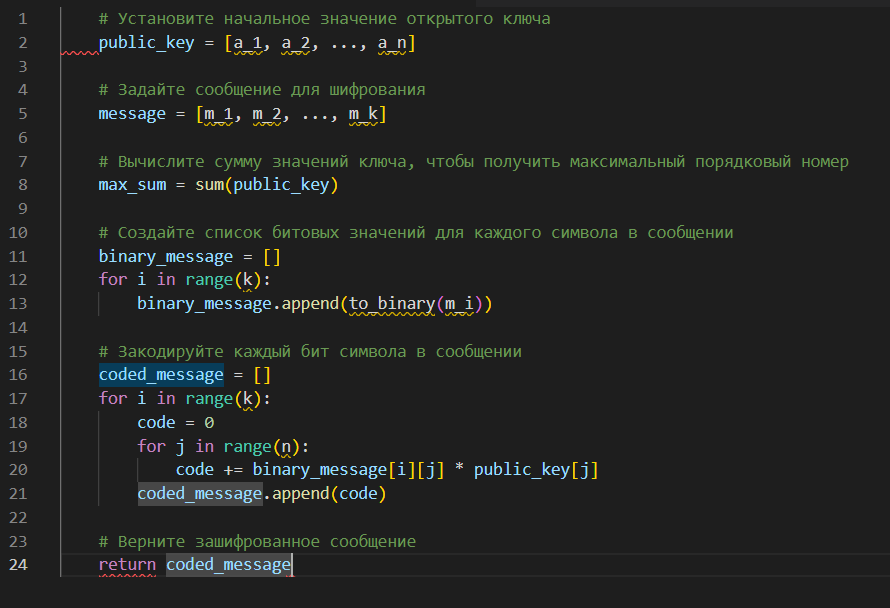


Рис. 5 – Псевдокод зашифрования для задачи о рюкзаке

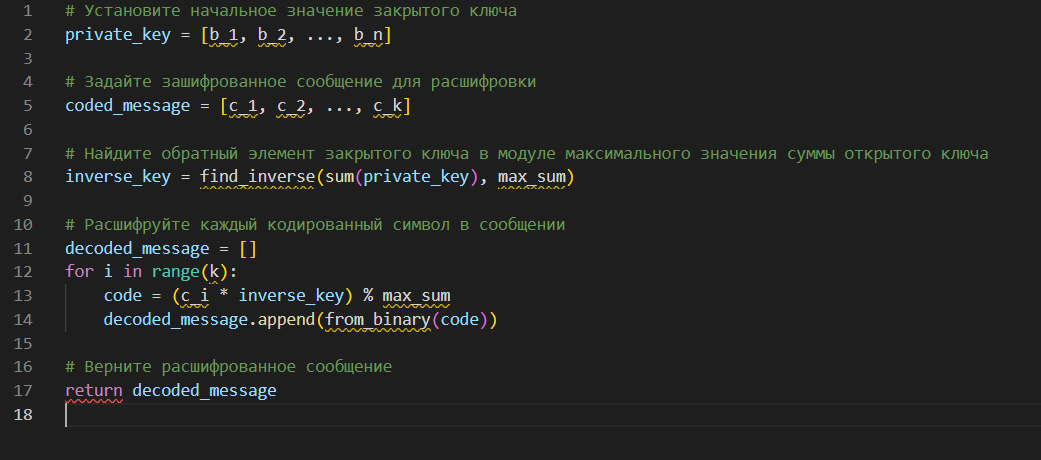


Рис. 6 – Псевдокод расшифрования для задачи о рюкзаке

1. Используя некоторый вектор S = (103, 107, 211, 430, 863, 1716, 3449, 6907, 13807, 27610), вычислить ключи для зашифрования и расшифрования сообщений.

Сумма всех элементов: 55203.

Предположим, что *n* = 55204. Так как *n* > 55203, сделаем *a* = 3001. НОД (*a*, *n*) = 1.

Найдем члены открытого ключа по формуле: *ei* = *di \* a* *mod* *n.* В результате получим закрытый ключ: 103, 107, 211, 430, 863, 1716, 3449, 6907, 13807, 27610; и открытый ключ: 33083, 45087, 25967, 20738, 50478, 15744, 27301, 26407, 31807, 51610.

1. Можно ли одновременно зашифровывать (и, соответственно, одновременно расшифровывать) более, чем по одному символу текста? Обосновать решение.

В алгоритме на основе задачи об укладке рюкзака нельзя одновременно зашифровывать и расшифровывать более, чем по одному символу текста. Это связано с тем, что каждый символ перед шифрованием должен быть преобразован в некоторое числовое значение, которое затем будет умножено на соответствующий элемент вектора. Если мы попытаемся зашифровать несколько символов одновременно, мы не сможем однозначно определить, какому символу соответствует каждое число в векторе.

1. Что такое «секретная лазейка»?

Ранцевые криптосистемы не являются криптостойкими. Зная числа *а*, *a*–1 и *n* («секретную лазейку»), можно восстановить сверхвозрастающую последовательность по нормальной последовательность.

1. Охарактеризовать криптостойкость алгоритма на основе задачи об укладке ранца.

Криптостойкость алгоритма на основе задачи об укладке ранца зависит от выбора секретного вектора, который определяет ключ шифрования. Если вектор выбран случайно и его элементы являются большими случайными числами, то алгоритм может быть криптостойким.

Однако, алгоритм на основе задачи об укладке ранца имеет некоторые ограничения, которые могут уменьшать его криптостойкость. Например, если вектор является периодическим или имеет другие структурные особенности, то он может быть уязвим для атак, таких как атака методом грубой силы или атака с помощью алгоритмов решения задачи об укладке ранца.

Также алгоритм на основе задачи об укладке ранца не обеспечивает аутентификацию и целостность данных, что может приводить к возможности атак с подменой данных или внедрению вредоносного кода.

Таким образом, криптостойкость алгоритма на основе задачи об укладке ранца зависит от выбора секретного вектора и от соблюдения определенных условий.

**Вывод:** в ходе лабораторной работы были приобретены практические навыки разработки и использования приложений для реализации асимметричных шифров. Было разработано приложение для реализации методов генерации ключевой информации и ее использования. Также была оценена скорость зашифрования/расшифрования.