Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчет по лабораторной работе №8**

**«Исследование асимметричных шифров RSA и Эль-Гамаля»**

Выполнил:

Cтудент 3 курса 1 группы

Парибок И. А.

Вариант 5

Минск 2023

**Теоретические сведения**

**Алгоритм RSA**

Для генерации двух ключей: тайного и открытого (а по сути – двух взаимосвязанных частей одного ключа, т. е. ключа, принадлежащего одному физическому лицу (или группе лиц), либо одному юридическому лицу), используются два больших случайных простых числа *p* и *q*. Для максимальной большей криптостойкости нужно выбирать *p* и *q* равной длины. Рассчитывается произведение: *n* = *pq*. Это есть один из трех компонент ключа, состоящего из чисел *n*, *e*, *d*.

Затем случайным образом выбирается второй компонент ключа (открытый ключ или ключ зашифрования, *e*, такой что *e* и (*p* – 1)(*q* – 1) являются взаимно простыми числами; вспомним, что (*p* – 1)(*q* – 1) = φ(*n*) – функция Эйлера).

Наконец, расширенный алгоритм Евклида используется для вычисления третьего компонента ключа: ключа расшифрования d такого, что выполняется условие:

****

Другими словами:

****

Для зашифрования/расшифрования используется ключ получателя: отправитель шифрует сообщение открытым ключом, а получатель расшифровывает шифртекст своим тайным ключом.

***Зашифрование***. Если шифруется сообщение *М*, состоящее из *r* блоков: *m1*, *m2*, …, *mi*, …, *mr*, то шифртекст *С* будет состоять из такого же числа (*r*) блоков, представляемых числами:

****

***Расшифрование***. Для расшифрования каждого зашифрованного блока производится вычисление вида:

****

**Алгоритм Эль-Гамаля**

***Генерация ключевой информации*.** Выбирается простое число *р*. Выбирается число (*g*, *g* < *p*), являющееся первообразным корнем числа р – очень важный элемент с точки зрения безопасности алгоритма.

Далее выбирается число *х* (*х* < *p*) и вычисляется последний компонент ключевой информации:

****

Владельцу сформированной ключевой информации, состоящей из 4 чисел, может посылаться некоторый шифртекст, созданный с использованием открытого ключа получателя: p, g, y. Расшифрование шифртекста получатель производит своим тайным ключом: p, g, х.

***Зашифрование***. Предположим, что сообщение *М* = {*mi*}, где *mi* – *i*-й блок сообщения.

Зашифрование отправителем (каждого отдельного блока *mi* исходного сообщения) предусматривает использование, как это особо подчеркивалось выше, некоторого случайного числа *k* (1 < *k* < *p* – 1).

Блок шифртекста (*ci*) состоит из двух чисел – *аi* и *bi*:

**Изображение выглядит как Шрифт, текст, белый, типография

Автоматически созданное описание**

Случайное число k должно сразу после вычисления уничтожаться.

***Расшифрование***. Выполняется по следующей формуле:

****

****

где (*ax*) –1 – обратное значение числа *ax* по модулю *p*.

**Практическое задание**

Целью первого заключалась в создании консольного приложения для вычисления параметра **y** по заданной формуле: **y ≡ ax mod n** и замера времени выполнения этой операции для различных значений параметров **a**, **x** и **n**.

Для достижения этой цели, был использован язык программирования C# и создано консольное приложение, которое генерирует случайные числа из указанных диапазонов и вычисляет параметр **y** для каждой комбинации параметров **a**, **x** и **n**. Для замера времени выполнения операции использовался класс **Stopwatch**.

Был использован следующий алгоритм вычисления параметра **y**:

1. Инициализировать переменные **result** и **power** равными 1 и **a** соответственно.
2. Пока **x** больше 0, выполнить следующие действия:
   1. Если младший бит **x** равен 1, то умножить **result** на **power** по модулю **n**.
   2. Вычислить **power** в квадрате по модулю **n**.
   3. Сдвинуть **x** на 1 бит вправо.
3. Вернуть **result**.

Для генерации простых чисел из указанного диапазона была использована функция **IsPrime**, которая проверяет, является ли число простым.

Результаты работы программы были представлены в виде таблицы, показывающей значения параметров **a**, **x**, **n** и **y**, а также время выполнения операции в миллисекундах для каждой комбинации. Были использованы 2 случайных значения **a** из диапазона от 5 до 35 и 5 случайных простых значений **x** из диапазона от 103 до 10100.

В результате работы программы было произведено успешное вычисление параметра **y** для всех возможных комбинаций параметров **a**, **x** и **n**. Время выполнения операции было замерено и отображено в таблице результатов. (рисунок 1).

Изображение выглядит как текст, снимок экрана, меню, число

Автоматически созданное описание  
 Рисунок 1 – Зависимости времени вычисления параметра у

В задании номер два требовалось создать реализацию асимметричных алгоритмов шифрования, включая RSA и Эль-Гамаля. В качестве сообзение выступала строка «Paribok Ulsha Alex». Параметы RSA алгоритма: p ­18014398241046527, q – 1298074214633706835075030044377087.В алгоритме Эль-Гамаля p – «433494437». На рисунках 2 и 3 представлены результаты выполнения программ c использованием алгоритма RSA и Эль-Гамаля соответственно.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Рисунок 2 – Результат выполнения RSA

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Рисунок 3 – Результат выполнения Эль-Гамаля

После анализа данных результатов можно отметить следующее: скорость расшифрования выше, чем скорость шифрования для обоих алгоритмов, что является положительным фактором. Также можно сравнить размеры сообщений до и после шифрования. В алгоритме RSA длина исходного сообщения равна длине закодированного сообщения, в то время как в алгоритме Эль-Гамаля длина исходного сообщения в два раза меньше зашифрованного сообщения (каждый символ исходного текста кодируется двумя символами шифртекста).

**Вывод**: в результате выполнения данной лабораторной работы были получены как теоретические, так и практические навыки работы с асимметричными алгоритмами шифрования RSA и Эль-Гамаля.

**Ответы на контрольные вопросы**

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

Ранцевый (рюкзачный) вектор - это последовательность целых чисел, которая может использоваться для решения задачи укладки ранца. Она представляет собой набор весовых значений, которые могут быть уложены в ранец таким образом, чтобы суммарный вес элементов в рюкзаке был равен определенному значению.

**2. Сформулировать задачу укладки ранца.**

Задача укладки ранца - это задача о выборе определенных элементов из набора элементов с различными весами таким образом, чтобы сумма весов выбранных элементов была равна заданной величине и оставшееся место в рюкзаке было заполнено наиболее эффективным образом.

**3. Если вектор рюкзака имеет вид (14, 28, 56, 82, 90, 132, 197, 284, 341, 455), то какими следует принять коэффициенты** bi **из выражения (7.1), чтобы получить *S* = 517? Каким будет решение задачи для *S* = 516**

Для получения S = 517 вектор значений bi должен иметь вид (1, 1, 0, 1, 0, 0, 1, 0, 0, 1). Фактически, мы выбираем элементы из вектора рюкзака, которые в сумме дают требуемую сумму 517. Если решаем задачу для S = 516, то необходимо получить вектор значений bi, чтобы сумма произведений значений bi на соответствующие элементы вектора рюкзака была равна 516. В этом случае можно получить вектор (1, 0, 0, 0, 0, 1, 1, 0, 1, 0), который соответствует выбору элементов из вектора рюкзака, в сумме дающих 516. При решении задачи укладки ранца также важно учитывать конкретные требования к рюкзаку и выбор оптимальных элементов, чтобы заполнить рюкзак наиболее эффективным образом.

**4. Что такое сверхвозрастающая последовательность? Привести примеры.**

Сверхвозрастающая последовательность - это последовательность целых чисел, такая что каждый следующий элемент больше суммы всех предыдущих элементов. Примеры сверхвозрастающих последовательностей: {1, 3, 7, 16, 34, 72, 151, 315}, {1, 2, 5, 13, 34, 89, 233, 610}.

**5. 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?**

можно преобразовать последовательности {89, 3, 11, 2, 45, 6, 22}, {3, 41, 5, 1, 21, 10}, {2, 3, 11, 29, 45, 6, 39} в сверхвозрастающие последовательности.

Чтобы получить сверхвозрастающую последовательность, мы можем выбирать элементы из последовательности, каждый раз стараясь сделать сумму предыдущих элементов как можно меньше:

Для последовательности {89, 3, 11, 2, 45, 6, 22} мы можем получить следующую сверхвозрастающую последовательность: {1, 2, 4, 8, 16, 32, 64}. Для этого мы можем выбрать элементы 2, 6, 11, 22 и 45, таким образом, чтобы любая сумма предыдущих элементов была меньше следующего элемента в выбранной сверхвозрастающей последовательности.

Для последовательности {3, 41, 5, 1, 21, 10} мы можем получить следующую сверхвозрастающую последовательность: {1, 2, 4, 8, 16, 32}. Для этого мы можем выбрать элементы 1, 3, 5, 21 и 41, таким образом, чтобы любая сумма предыдущих элементов была меньше следующего элемента в выбранной сверхвозрастающей последовательности.

Для последовательности {2, 3, 11, 29, 45, 6, 39} мы также можем получить сверхвозрастающую последовательность: {1, 3, 6, 13, 27, 55}. Для этого мы можем выбрать элементы 2, 3, 6, 11, 29, 39 и 45, таким образом, чтобы любая сумма предыдущих элементов была меньше следующего элемента в выбранной сверхвозрастающей последовательности.

Таким образом, все эти последовательности могут быть преобразованы в сверхвозрастающие последовательности путем выбора определенных элементов их членами.

**6. Записать в виде псевдокода алгоритм шифрования и алгоритм расшифрования сообщения на основе задачи об укладке ранца.**

Алгоритм зашифрования на основе задачи об укладке ранца:

* Выбрать рюкзак с заданным вектором весов (w1, w2, ..., wn) таким, чтобы любая сумма его элементов могла быть выражена в виде линейной комбинации чисел 1, 2, ..., m-1, где m - натуральное число, большее суммы элементов рюкзака.
* Выбрать случайную последовательность (b1, b2, ..., bn) из чисел 0 и 1 длиной n.
* Зашифровать сообщение, заменив каждый символ на взвешенную сумму соответствующих элементов рюкзака, взятых с коэффициентами из последовательности b (то есть, зашифрованное сообщение будет представлять собой последовательность сумм bi \* wi для i от 1 до n).

Алгоритм расшифрования на основе задачи об укладке ранца:

* Найти обратную к вектору весов (w1, w2, ..., wn) по модулю m-1 (если такая обратная существует), используя расширенный алгоритм Евклида.
* Для каждого зашифрованного символа bi \* wi вычислить произведение (bi \* wi) \* r по модулю m-1, где r - найденная в п. 1 обратная величина.
* Каждое произведение из п. 2 можно выразить в виде линейной комбинации чисел 1, 2, ..., m-1, используя теорему об укладке в ранец. Для этого нужно найти такие числа a1, a2, ..., am-1, что (bi \* wi) \* r = a1 \* w1 + a2 \* w2 + ... + a(n-1) \* w(n-1) + a(n) \* wn.
* Значения ai представляют собой биты расшифрованного символа. Дешифрованное сообщение можно получить, объединив биты в символы.

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

Для зашифрования и расшифрования сообщений на основе задачи об укладке ранца необходимо выполнить следующие шаги для данного вектора S:

1. Сгенерировать сверхвозрастающую последовательность, например: {1, 3, 7, 16, 34, 72, 151, 315, 654, 1353}.

2. Выбрать произвольное число из последовательности, например 430, и найти комбинацию элементов из последовательности, которая будет равна этому числу: {7, 16, 34, 72, 151, 315}.

3. Полученный набор элементов будет ключом для зашифрования и расшифрования сообщений.

Таким образом, ключ для данного вектора S будет равен {1, 1, 0, 1, 1, 0}, потому что сумма выбранных элементов {7, 16, 34, 72, 151, 315} будет равна 430, а соответствующие биты ключа равны: первый элемент - 1, второй элемент - 1, третий элемент - 0, четвертый элемент - 1, пятый элемент - 1, шестой элемент - 0.

**9. Что такое «секретная лазейка»?**

«Секретная лазейка» (англ. "backdoor") в криптографии – это специально встроенный механизм в криптографический алгоритм или программное обеспечение, который позволяет обходить защиту и получать доступ к зашифрованным данным без необходимости знания ключа шифрования или пароля.

**10. Охарактеризовать криптостойкость алгоритма на основе задачи об укладке ранца.**

Ранцевые криптосистемы не являются криптостойкими. А. Шамир и Р. Циппел обнаружили, что, зная числа *а*, *a–1* и *n* («секретную лазейку»), можно восстановить сверхвозрастающую последовательность по нормальной последовательности