Лабораторная работа № 7

Исследование асимметричных шифров

Цель: изучение и приобретение практических навыков разработки и использования приложений для реализации асимметричных 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.

7.1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

7.1.1 Основные свойства асимметричных криптосистем

Две известные нам проблемы, связанные с практическим использованием симметричных криптосистем, стали важными побудительными мотивами для разработки принципиально нового класса методов шифрования: криптографии с *открытым* ключом или *асимметричной криптографии*.

Концепция нового подхода предложена Уитфилдом Диффи (Whitfield Diffie) и Мартином Хеллманом (Martin Hellman), и, независимо, Ральфом Мерклом (Ralph Merkle).

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.

Определение 1. *Односторонней функцией* (one-way function) называется математическая функция, которую относительно легко вычислить, но трудно найти по значению функции соответствующее значение аргумента, т. е., зная x, легко вычислить $f(x)$, но по известному $f(x)$ трудно найти подходящее значение x.

Практически первой реализацией идеи Диффи-Хеллмана стал алгоритм согласования по открытому каналу тайного ключа между абонентами A и B [2, 4].

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.

В данной работе мы будем работать над аспектами решения первой из указанных задач.

По мнению Диффи и Хеллмана алгоритм шифрования с открытым ключом, должен:

- вычислительно легко создавать пару (открытый ключ, e – закрытый ключ, d),
- вычислительно легко зашифровывать сообщение M_i открытым ключом,
- вычислительно легко расшифровывать сообщение C_i, используя закрытый ключ,
- обеспечивать непреодолимую вычислительную сложность определения соответствующего закрытого ключа при известном открытом ключе,
- обеспечивать непреодолимую вычислительную сложность восстановления исходного (открытого сообщения, M_i) зная только открытый ключ и зашифрованное сообщение, C_i.

7.1.2 Криптоалгоритм на основе задачи об укладке ранца

7.1.2.1 Общая характеристика алгоритма

Алгоритм разработан Р. Мерклом и М. Хеллманом. Стал первым алгоритмом шифрования с открытым ключом широкого назначения.

Определение 2. *Ранцевый* (*рюкзачный*) *вектор* $\boldsymbol{S} = (s_1, \ldots, s_z)$ – это упорядоченный набор из z, $z \geq 3$, различных натуральных чисел s_i. Входом задачи о

ранце (рюкзаке) называем пару ($\boldsymbol{S}$, S), где $\boldsymbol{S}$ – рюкзачный вектор, а S – натуральное число.

Решением для входа ($\boldsymbol{S}$, S) будет такое подмножество из $\boldsymbol{S}$, сумма элементов которого равняется S.

В наиболее известном варианте задачи о ранце требуется выяснить, обладает или нет данный вход ($\boldsymbol{S}$, S) решением. В варианте, используемом в криптографии, нужно для данного входа ($\boldsymbol{S}$, S) построить решение, зная, что такое решение существует. Оба эти варианта являются NP-полными.
Имеются также варианты этой задачи, которые не лежат даже в классе NP.

Как видим, проблема укладки ранца формулируется просто. Дано множество предметов общим числом z различного веса. Спрашивается, можно ли положить некоторые из этих предметов в ранец так, чтобы его вес стал равен определенному значению S? Более формально задача формулируется так: дан набор значений $s_1, s_2, \ldots, s_z$ и суммарное значение S. Требуется вычислить значения s_i такие, что

$$S = b_1 s_1 + b_2 s_2 + \ldots + b_z s_z. \qquad (7.1)$$

Здесь b_i может быть либо нулем, либо единицей. Значение $b_i = 1$ означает, что предмет m_i кладут в рюкзак, а $b_i = 0$ – не кладут.

Суть метода для 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 $\boldsymbol{e}$, который легко использовать для зашифрования, но невозможно – для расшифрования. В качестве закрытого ключа $\boldsymbol{d}$ применяется легкий для укладки ранец, который предоставляет простой способ расшифрования сообщения.

В качестве закрытого ключа $\boldsymbol{d}$ (легкого для укладки ранца) используется *сверхвозрастающая последовательность, состоящая из z элементов:* $d_1, d_2, \ldots, d_z$: $\boldsymbol{d} = \{d_i\}$, $i = 1, \ldots, z$.

Определение 3. *Сверхвозрастающей* называется последовательность, в которой каждый последующий член больше суммы всех предыдущих.

Пример 1. Последовательность {2, 3, 6, 13, 27, 52, 105, 210} ($z = 8$) является сверхвозрастающей, а {1, 3, 4, 9, 15, 25, 48, 76} – нет.

7.1.2.2 Алгоритм укладки ранца

на основе сверхвозрастающей последовательности

Необходимо по очереди анализировать некоторый «текущий вес» S предметов, составляющих сверхвозрастающую последовательность; в результате анализа нужно упаковать (доупаковать) ранец.

1. В качестве текущего выбирается число S, которое сравнивается с «весом» самого тяжелого предмета (d_z);

если текущий вес меньше веса данного предмета, то его в ранец не кладут (0), в противном случае его укладывают ($b_z = 1$) в ранец и переходят к анализу очередного (в общем случае – i-го предмета).

2. Если на предыдущем (i-м шаге) предмет пополнил ранец, то текущий вес уменьшают на вес положенного предмета ($S = S – d_i$); переходят к следующему по весу предмету в последовательности: d_{i-1}.

Шаги повторяются до тех пор, пока процесс не закончится.

Если текущий вес уменьшится до нуля ($S = 0$), то решение найдено. В противном случае – нет.

Пример 2. Пусть полный вес ранца равен 270, а последовательность весов предметов равна: {2, 3, 6, 13, 27, 52, 105, 210} (d_1 =2, d_2 =3, d_3 =6 и т.д.).

Шаг 1. S=270. Самый большой вес предмета ($d_z = d_8$)– 210. Он меньше 270, поэтому предмет весом 210 кладут в ранец (1): вычитают 210 из 270 и получают 60: $S = S – d_8 = 270 – 210 = 60$.

Шаг 2. Следующий наибольший вес последовательности равен 105. Он больше 60, поэтому предмет весом 105 в ранец не кладут (0): $S = S – 0 = 60$.

Шаг 3. Следующий самый тяжелый предмет имеет вес 52. Он меньше 60, поэтому предмет весом 52 также кладут в рюкзак (1): $S = S – 52 = 8$.

На 4-м и на 5-м шагах рюкзак не пополняется. Текущий вес его остается неизменным.

На 6-м шаге в ранец кладут предмет весом 6 и на 8-м шаге – весом 2.

В результате полный вес уменьшится до 0, т. е. получили текущее значение$S = 0$.

Если бы этот ранец был бы использован для расшифрования, то открытый текст, полученный из значения шифртекста 270, был бы равен 10100101.

Открытый ключ $\boldsymbol{e}$ 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 $\boldsymbol{e}$ ($\boldsymbol{e} = \{e_i\}$, $i = 1,\ldots, z$) все значения закрытого ключа умножаются на некоторое число a по модулю n:

$$e_i = d_i\, a \pmod{n}. \qquad (7.2)$$

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

Пример 3. Сумма элементов последовательности {2, 3, 6, 13, 27, 52, 105, 210} равна 418: 2+3+6+13+27+52+105+210=418. Элементы d_i этой последовательности являются элементами ключа $\boldsymbol{d}$: $\boldsymbol{d} = \{d_i\}$. Примем, что $n = 420$ и a=31.

В соответствии с этими при использовании (7.2) результат построения нормальной последовательности (открытого ключа, $\boldsymbol{e}$) будет представлен следующим множеством: {62, 93, 186, 403, 417, 352, 315, 210}: $e_1 = 62$, $e_2 = 93$ и т. д.

7.1.2.3 Зашифрование сообщения

В основе операции лежит соотношение (7.1).

Для зашифрования сообщения (M) оно сначала разбивается на блоки, по размерам равные числу (z) элементов последовательности в ранце. Затем, считая, что *1* указывает на присутствие элемента последовательности в ранце, а *0* – на его отсутствие, вычисляются полные веса рюкзаков (S_i, $i = 1, \ldots, z$): по одному ранцу для каждого блока сообщения с использованием открытого ключа получателя, $\boldsymbol{e}$.

*Пример*4. Возьмем открытое сообщение M, состоящее из 7 букв (m_j), которые представим в бинарном виде (1 символ текста – 1 байт). Бинарное представление символов дано в первом столбце нижеследующей таблицы (табл. 7.1).

Открытый ключ, $\boldsymbol{e}$: {62, 93, 186, 403, 417, 352, 315, 210}

Результат зашифрования (упаковки ранца) каждого блока (буквы) сообщения с помощью открытого ключа представлен в правом столбце таблицы 7.1.

Таблица 7.1

Пояснение к примеру зашифрования сообщения укладкой ранца

Бинарный код символа m_j сообщения	Укладка ранца	Вес ранца
11010000	62+93+403	558
11000010	62+93+315	470
11000000	62+93	155
11000001	62+93+210	365
11001110	62+93+417+352+315	1239
11000000	62+93	155

11001100	62+93+417+352	924

Таким образом, зашифрованное сообщение C = 558 470 155 365 1239 155 924: c_1 = 558, c_2 = 470 и т. д.

7.1.2.4 Расшифрование сообщения

Для расшифрования сообщения получатель (использует свой тайный ключ, ***d***: сверхвозрастающую последовательность) должен сначала определить обратное к a число: a^{-1}, такое что

$$a\ a^{-1} (\text{mod}\ n) = 1. \qquad (7.3)$$

Для вычисления обратных чисел по модулю можно использовать известный нам расширенный алгоритм Евклида.

После определения обратного числа каждое значение шифрограммы (c_i) преобразуется в соответствии со следующим соотношением:

$$S_i = c_i\ a^{-1} \text{ mod } n. \qquad (7.4)$$

Полученное на основании последней формулы для каждого блока число далее рассматривается как заданный вес ранца, который следует упаковать по изложенному выше алгоритму, используя сверхвозрастающую последовательность (тайный ключ получателя).

Продолжим рассмотрение примера 4.

В нашем примере значение a^{-1} = 271: 31*271 mod 420 = 1.

Вспомним, что сверхвозрастающая последовательность равна ***d***: ***d***= {2, 3, 6, 13, 27, 52, 105, 210}, а также n = 420, a = 31; шифртекст: 155 365 558 155 924 1239 470

Расшифрование первого блока шифртекста. Сначала вычисляем, используя (7.4), вес первого ранца (при c_1=155):

$$S_1 = c_1 * a^{-1} (\text{mod}\ n) = 155*271\ (\text{mod}\ 420) = 5.$$

Используем S_1 =5 и с помощью сверхвозрастающей последовательности ({2, 3, 6, 13, 27, 52, 105, 210}) и известного алгоритма упаковки ранца получаем m_1= 11000000. Понятно, что последней бинарной последовательности должен соответствовать некоторый символ алфавита в используемой таблице кодировки.

Расшифрование остальных блоков шифртекста производится аналогично.

7.1.3 Безопасность криптоалгоритма на основе задачи об укладке ранца

Криптостойкость алгоритма во многом определяется скоростью (временем) поиска нужного варианта укладки ранца. Понятно, что для последовательности из шести-десяти или немногим более того элементов нетрудно решить задачу укладки ранца, даже если последовательность не является сверхвозрастающей. При практической же реализации алгоритма ранец должен содержать не менее нескольких сотен элементов. Длина каждого члена сверхвозрастающей последовательности должна быть несколько сотен бит, а длина числа n – от 100 до 200 бит. Для получения этих значений практические реализации алгоритма используют генераторы ПСП.

С другой стороны, известный способ определения, какие предметы кладутся в ранец, является проверка возможных решений до получения правильного. Самый быстрый алгоритм, принимая во внимание различную эвристику, имеет экспоненциальную зависимость от числа возможных предметов. Если добавить к последовательности весов еще один член, то найти решение станет вдвое труднее. Это намного труднее сверхвозрастающего ранца, где, при добавлении к последовательности одного элемента, поиск решения увеличивается на одну операцию.

Ранцевые криптосистемы не являются криптостойкими. А. Шамир и Р. Циппел обнаружили, что, зная числа a, a^{-1} и n («секретную лазейку»), можно восстановить сверхвозрастающую последовательность по нормальной последовательности [4]. Важно то, что числа a и n («секретная пара») не обязательно должны быть теми же, что использовались при создании системы легальным пользователем.

Достаточно подробное и понятное для начинающего криптоаналитика рассмотрение криптостойкости ранцевого алгоритма изложено в [37].

7.2 ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Разработать авторское оконное приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться доступными библиотеками либо программными кодами.

В основе вычислений – кодировочные таблицы Base64 и ASCII.

Приложение должно реализовывать следующие операции:

- генерация сверхвозрастающей последовательности (тайного ключа); старший член последовательности – 100-битное число; в простейшем

случае принимается $z = 6$ (для кодировки Base64) и $z = 8$ (для кодировки ASCII);

- вычисление нормальной последовательности (открытого ключа);
- зашифрование сообщения, состоящего из собственных фамилии, имени и отчества;
- расшифрование сообщения;
- оценка времени выполнения операций зашифрования и расшифрования.

2. Проанализировать время выполнения операций зашифрования/расшифрования при увеличении числа членов ключевой последовательности: при использовании разных таблиц колировки.

3. Результаты оформить в виде отчета по установленным правилам.

ВОПРОСЫ ДЛЯ КОНТРОЛЯ И САМОКОНТРОЛЯ

1. Что такое «ранцевый (рюкзачный) вектор»? Дать определение.
2. Сформулировать задачу укладки ранца.
3. Если вектор рюкзака имеет вид (14, 28, 56, 82, 90, 132, 197, 284, 341, 455), то какими следует принять коэффициенты b_i из (7.1), чтобы получить $S = 517$? Каким будет решение задачи для $S = 516$?
4. Что такое сверхвозрастающая последовательность? Привести примеры.
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?
6. Записать в виде псевдокода алгоритм зашифрования и алгоритм расшифрования сообщения на основе задачи об укладке ранца.
7. Используя некоторый вектор $\boldsymbol{S}$ = (103, 107, 211, 430, 863, 1716, 3449, 6907, 13807, 27610), вычислить ключи для зашифрования и расшифрования сообщений.
8. Можно ли, с Вашей точки зрения, одновременно зашифровывать (и, соответственно, – одновременно расшифровывать) более, чем по одному символу текста. Обосновать решение. Привести примеры решений.
9. Что такое «секретная лазейка»?
10. Охарактеризовать криптостойкость алгоритма на основе задачи об укладке ранца.