МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Методическое пособие к лабораторным работам и практическим занятиям

2021

Составитель: д-р техн. наук, проф. Ю.М. Краковский

Содержится краткое изложение теоретического материала с указанием списка литературы и контрольных вопросов, а также даны методические рекомендации и варианты лабораторных работ.

Приведены лекционные материалы для проведения трех практических занятий.

Данная дисциплина является продолжением дисциплины «Информационная безопасность», поэтому нумерация лабораторных работ начинается с номера 7.

ОГЛАВЛЕНИЕ

Лабораторная работа № 7
«Поточное шифрование информации»

Цель работы – освоить технологию создания ключа для поточного шифрования на основе регистра сдвига с линейной обратной связью.

Введение

В современных системах симметричного шифрования информации широкое применение находят системы поточного шифрования.

Необходимо различать процедуру создания поточного ключа и процедуру поточного шифрования (хотя эти процедуры очень связаны и могут выполняться совместно). В связи с этим разделим поточное шифрование на две процедуры:

а) само поточное шифрование;

б) создание ключа для поточного шифрования.

Примечание – в российских национальных стандартах, такое разделение не выделяется, что усложняет процесс их понимания.

Поточное шифрование

Современные поточные системы при шифровании используют побитное сложение по модулю 2 (операцию *XOR*). Мы ее уже не однократно использовали, например, при контроле целостности информации, когда рассматривали контроль целостности при случайных воздействиях.

Пусть m – исходное электронное сообщение, конфиденциальность которого необходимо обеспечить; c – зашифрованное сообщение, полученное из исходного; K – секретный поточный ключ. Тогда при зашифровании используется следующая операция

$$c_i=m_i(+)K_i, \quad i=1,2,\ldots,I, \tag{7.1}$$

где m_i – i-й бит исходного сообщения; K_i – i-й бит поточного ключа; c_i – i-й бит зашифрованного сообщения; I – число битов в сообщениях и в ключе; (+) – операция сложения по модулю 2.

При расшифровании используется та же операция

$$m_i=c_i(+)K_i, \quad i=1,2,\ldots,I. \tag{7.2}$$

Убедимся, что

$$c_i(+)K_i =(m_i(+)K_i)(+)K_i)= m_i(+)(K_i(+)K_i)= m_i.$$

Для поточного шифрования справедливо равенство, которое вытекает из свойства операции сложения по модулю 2

$$K_i=m_i(+)c_i, \quad i=1,2,\ldots,I. \tag{7.3}$$

Заметим, что в российских национальных стандартах поточное шифрование называют гаммированием, а поточный ключ – гаммой.

Подчеркнем, что поточный ключ K по длине должен быть не меньше длин исходного и зашифрованного сообщений. Дополнительными условиями

являются его случайность при создании (равновероятность) и одноразовость (используется один раз).

При этих условиях в соответствии с теоремой К. Шеннона поточная система шифрования является совершенно криптостойкой.

Рассмотрим пример.

Отправитель *A* создает исходное сообщение и поточный ключ необходимой длины. Пусть исходное сообщение: 1100100100110101; поточный ключ: 1001110010110011.

Тогда используя (7.1), отправитель при зашифровании выполняет следующую операцию, получая зашифрованное сообщение.

$$\begin{array}{r} A\text{: } m - 1100100100110101 \\ (+) \\ K - 1001110010110011 \\ c - 0101010110000110 \end{array}$$

Зашифрованное сообщение (*c*) отправляется получателю *B*. Получатель, получив это сообщение и сам создав поточный ключ (он обязательно должен совпадать с поточным ключом, созданным отправителем), а также используя (7.2), осуществляет расшифрование.

$$\begin{array}{r} B\text{: } c - 0101010110000110 \\ (+) \\ K - 1001110010110011 \\ m - 1100100100110101 \end{array}$$

Убедимся, что расшифрованное сообщение, полученное получателем, равно исходному, созданное отправителем (в нашем случае совпадение получено). Убедимся также в справедливости равенства (7.3).

$$\begin{array}{r} m - 1100100100110101 \\ (+) \\ c - 0101010110000110 \\ K - 1001110010110011 \end{array}$$

Что необходимо выделить в приведенном примере: при поточном шифровании поточный ключ не распределяется заранее. Отправитель и получатель создают его каждый для себя. Более подробно этот вопрос будет рассмотрен ниже.

Важным достоинством поточного шифрования является его высокая скорость (пока не рассматривается вопрос создания поточного ключа).

Недостатком поточного шифрования является проблема синхронизации сообщений при зашифровании и расшифровании (вставка или потеря битов).

Выделяют синхронные поточные криптосистемы, когда гамма создается независимо от исходного сообщения, и самосинхронизирующие поточные криптосистемы, когда такая связь имеется.

В синхронных поточных криптосистемы, в отличии от самосинхронизирующихся, отсутствует эффект размножения ошибок, т.е. количество ошибок в зашифрованном сообщении равно количеству ошибок в расшифрованном. Но при использовании синхронных поточных криптосистем необходимо решить задачу синхронизации генераторов гаммы у отправителя и получателя. Кроме того, выпадение или вставка битов в зашифрованном сообщении приведет к ошибкам в расшифрованном тексте.

В синхронных шифрах используют специальные маркеры, которые вставляют в сообщения, а в самосинхронизирующих проблема синхронизации решается технологией создания поточного ключа или средствами контроля целостности.

Исторически первой поточной криптосистемой является система Вернама (1917), в которой в качестве ключевой последовательности использовалась уникальная случайная гамма. При этом размер ключа соответствовал длине исходного текста.

Создание поточного ключа

Как отмечено выше, современные поточные криптосистемы используют при шифровании побитное сложение по модулю 2. Многообразие поточных криптосистем определяется технологией создания поточного ключа.

Создание поточных ключей это значительное по многообразию методов направление в криптографии. Этому направлению уделяется значительное место в литературе по криптографии. Существуют различные классификации этих методов. Мы рассмотрим следующую.

Все многообразие методов создания поточных ключей разделим на три группы:

1) использование всевозможных регистров сдвига, среди которых можно выделить регистры сдвига с линейной обратной связью;

2) использование блочных алгоритмов шифрования;

3) использование всевозможных генераторов псевдослучайных последовательностей (генераторов случайных чисел).

В российских национальных стандартах поточный ключ создается блоками с использованием блочного шифрования, поэтому этот вопрос мы рассмотрим позже, когда будем рассматривать эти стандарты.

В этой лабораторной работе рассматривается технология создания поточного ключа на основе регистра сдвига с линейной обратной связью (РСЛОС). Подобные регистры имеют большое практическое значение в современной криптографии.

Как правило под регистром понимают некую память, содержащую n битов. Как правило нумерация битов происходит справа налево. Подобные

регистры используются, например, в процессорах для промежуточного хранения информации.

Существуют регистры, в которых информацию можно сдвигать влево, вправо или циклически.

В рассматриваемом методе сдвиг происходит вправо на один бит.

Введем понятие исходного состояния регистра, такта и шага. Исходное состояние регистра, это некоторый заданный набор его битов. Если используется регистр сдвига при n=8 (реально такие регистры не используются), то можно выбрать любую комбинацию, например: 11000101.

Такт это два шага, которые позволяют получить новое состояние регистра сдвига и один бит поточного ключа.

На первом шаге регистр сдвигается вправо на один бит. Тем самым началось создание следующего состояния регистра (назовем его текущим). При этом значение младшего бита «выдавливается» и попадает в последовательность битов поточного ключа. В результате сдвига значение старшего бита текущего регистра неопределённо, поэтому его необходимо определить.

В данном методе используется понятие отводной последовательности. Отводная последовательность – это некоторая совокупность битов регистра, выбранная специальным образом (выбор номеров битов отводной последовательности осуществляется на основе рекомендаций специального раздела математики). На втором шаге такта биты отводной последовательности предыдущего состояния регистра складываются по модулю два. Полученное значение заносится в старший бит текущего регистра.

Таким образом, за один такт, содержащий два шага, создано новое состояние регистра сдвига и получен один бит в последовательности битов поточного ключа. Такты последовательно выполняются требуемое число раз (в зависимости от необходимой длины поточного ключа).

Если n – число битов (разрядов) в регистре сдвига, то максимальный период этого генератора равен 2^n-1 (регистр должен принять значения всех возможных комбинаций значений битов, исключая нулевую комбинацию). Отсюда вытекает, что начальное состояние регистра может быть любой комбинацией битов.

Для того, чтобы конкретный РСЛОС имел максимальный период, многочлен, ассоциированный с отводной последовательностью, должен быть примитивным по модулю 2, т.е. не раскладываться на произведение двоичных многочленов меньшей степени. Например, многочлен

$$x^{32}+x^7+x^5+x^3+x^2+x+1 \qquad (7.4)$$

примитивен по модулю 2. Степень многочлена задает длину РСЛОС в битах.

Тогда многочлен (7.4) определяет 32-битовый регистр сдвига, когда новый старший бит определяется сложением по модулю 2 битов с номерами: 32, 7, 5, 3, 2, 1 (эти биты являются правильной отводной последовательностью). Максимальная длина гаммы равна 2^{32}-1 битов (более четырех миллиардов).

Рассмотрим пример.

Рассмотрим четырехбитный регистр сдвига, для которого многочлен, примитивный по модулю 2 имеет вид

$$x^4+x+1,$$

поэтому отводная последовательность определяется первым и четвертым битами. При такой отводной последовательности максимальный период равен $2^4-1=15$.

Выберем в качестве начального состояние регистра следующую комбинацию: 1010 (подчеркнутые биты являются отводной последовательностью). Правый бит имеет номер 1 (младший бит), левый бит имеет номер 4 (старший бит), сдвиг происходит вправо на один бит.

Напомним, что такт состоит из двух шагов: сдвиг вправо и вычисление старшего бита путем сложения по модулю 2 битов отводной последовательности (не определенный старший бит после первого шага обозначим ≈).

На первом шаге получим: ≈101→0. Один бит поточного ключа сформировали (0).

На втором шаге сначала вычислим бит, используя значения отводной последовательности предыдущего состояния регистра: 1(+)0=1. Затем занесем его в старший бит текущего состояния регистра, получим: 1101 (создали новое состояние регистра). Эти такты повторяем.

Ниже приведены состояния регистра сдвига и полученная гамма длиной 15 бит (рис. 7.1)

1010 1101 0110 0011 1001 0100 0010 0001 – Состояния
1000 1100 1110 1111 0111 1011 0101 1010 – регистра
010110010001111 0 – поточный ключ

Рис. 7.1. Состояния регистра сдвига и значения поточного ключа

Как мы видим, 16-е состояние регистра совпало с первым, а 16-й бит поточного ключа совпал с 1-м.

Рассмотрим реальный поточный алгоритм *A*5, который используется при шифровании *GSM* европейского стандарта для цифровых сотовых телефонов. В нем используются три РСЛОС длиной 19, 22 и 23 битов; *i*-й бит гаммы является функцией от трех РСЛОС (в этом алгоритме используется так называемое управление тактированием, когда в каждом такте взаимодействуют два РСЛОС).

Доказано, что если длины регистров взаимно просты и у них правильно выбраны отводные последовательности, то итоговый ключ имеет максимально возможную длину. В алгоритме *A*5 это условие выполняется.

Секрет, который должна знать лишь одна пара (отправитель-получатель), – это начальное состояние регистра сдвига. Если регистров сдвига несколько, то секрет это начальные состояния всех регистров. Зная

этот секрет, отправитель и получатель смогут создать одинаковый поточный ключ.

Имея поточный ключ, можно провести шифрование сообщений.

Рассмотрим пример.

В этом примере отправитель *A* сформировал 12-битное сообщение (*m*) и подготовил для него 12-битный поточный ключ *K*. Далее он провел зашифрование сообщения, используя операцию (7.1). Затем он отправил зашифрованное сообщение *c* получателю.

Получатель *B* получил зашифрованное сообщение *c*, подготовил для него поточный ключ *K* (ключи отправителя и получателя совпадают) и провел расшифрование, используя операцию (7.2). В результате он получил сообщение *m*, которое сформировал отправитель.

$$
\begin{aligned}
A - m&: 100110110010 \\
&\qquad\qquad (+) \\
K&: 100011001111 \\
c&: 000101111101 \rightarrow B \\
B - c&: \ 000101111101 \\
&\qquad\qquad (+) \\
K&: 100011001111 \\
m&: 100110110010
\end{aligned}
$$

К настоящему моменту созданы и используются различные поточные криптосистемы, например, *A*5, *RC*4, *SEAL* и др. Также используются специальные режимы поточного шифрования в национальных стандартах, например, ГОСТ 28147-89, ГОСТ Р 34.13-2015, система *AES* и др.

Описание лабораторной работы

В лабораторной работе №7, используя 8-ми битовый РСЛОС, необходимо получить ключ для поточной криптосистемы максимальной длины. Биты в РСЛОС нумеруются справа налево.

Исходное состояние регистра задается двухзначным 16-ричным числом. Таблица соответствия для 16-ричных цифр (тетрад) приведена в таблице 7.1.

Таблица 7.1

Таблица соответствия двоичных тетрад и шестнадцатиричных цифр

ДСС	ШСС	ДСС	ШСС	ДСС	ШСС	ДСС	ШСС
0000	0	0100	4	1000	8	1100	C
0001	1	0101	5	1001	9	1101	D
0010	2	0110	6	1010	A	1110	E
0011	3	0111	7	1011	B	1111	F

Отводная последовательность выбрана произвольно, поэтому гамма не обязательно имеет длину 2^8-1. Студент для своего варианта должен определить ключ (гамму) длиной 15 битов.

Необходимо проверить условие (7.3).

Исходные данные для своего варианта приведены в таблице 7.2.

Лабораторную работу можно выполнить «вручную» или написав программу. После получения ключа студент должен провести поточное шифрование «вручную», используя фрагмент ключа, например, используя полученный ключ с 3-го бита.

Список контрольных вопросов

1. Основные понятия криптографии.
2. Классификация криптосистем.
3. Общая схема симметричного шифрования.
4. В чем отличие шифрования от дешифрования.
5. Способы формирования ключей для поточного шифрования.
6. Какой аспект безопасности обеспечивает шифрование.
7. Что такое отводная последовательность.
8. Как работает РСЛОС.
9. Вопросы по поточному шифрованию.

Варианты лабораторной работы

Таблица 7.2

№	Начальное состояние	Отводная последовательность	Примечание
1	A3	1,3,8	
2	B5	1,4,8	
3	A9	1,2,8	
4	C7	1,5,8	
5	F3	1,3,8	
6	F7	2,4,8	
7	E7	2,3,8	
8	D6	2,5,8	
9	D7	2,6,8	
10	A7	2,7,8	
11	B7	3,6,8	
12	C9	3,7,8	
13	F9	3,5,8	
14	E5	3,4,8	
15	D5	4,6,8	
16	78	2,6,8	
17	94	1,3,8	
18	E7	2,5,8	
19	D6	2,3,8	
20	D7	2,7,8	
21	A7	2,6,8	
22	BA	3,6,8	
23	CB	3,4,8	
24	F7	3,5,8	
25	EA	3,4,8	
26	D7	4,6,8	
27	7B	2,6,8	
28	85	1,4,8	

Лабораторная работа № 8
«Генерирование секретных ключей для симметричной криптосистемы»

Цель работы – ознакомиться с технологией генерирования секретных ключей.

Введение

Пусть нам необходимо получить значения равномерно распределенной случайной последовательности (РРСП) с элементами

$$x_1, \ldots x_t, \ldots \quad (8.1)$$

Можно выделить два подхода к получению этой последовательности:

1) создание и реализация алгоритмов, обеспечивающих высокое быстродействие;

2) создание и реализация алгоритмов, обеспечивающих высокую криптостойкость.

Первое направление в большей степени используется в имитационном моделировании, а второе – в криптографии и, в частности, при создании секретных ключей при поточном или блочном шифровании.

Выделяют три типа генераторов РРСП: табличный, физический и программный. Программный генератор РРСП – программа имитации на компьютере реализации РРСП. Имитируемая последовательность (8.1) называется псевдослучайной, т.к. она вычисляется по детерминированному алгоритму. В тоже время ее статистические свойства «близки» по определенным критериям к свойствам РРСП. Поэтому эти числа называют псевдослучайными (в отличии от настоящих случайных, которые реализуются физическими генераторами).

Важным параметром генераторов РРСП является их период T – длина неповторяющихся (несовпадающих) значений x_t: $x_t=x_{t+T}=x_{t+2T}=\ldots$

Выделяют несколько подходов к построению алгоритмов генерации. Мы рассмотрим конгруэнтный алгоритм как наиболее распространенный метод. Среди конгруэнтных алгоритмов используются линейные, мультипликативные и с простым модулем.

Мультипликативный конгруэнтный алгоритм с простым модулем

Рассмотрим в качестве примера мультипликативный конгруэнтный алгоритм с простым модулем m (результат – 4-х байтовое целое число)

$$x_{t+1}=a \cdot x_t \bmod m, \; t=0, 1, \ldots, \quad (8.2)$$

где x_0 – начальное значение (любое значение из диапазона от 1 до m-1), a – специально выбранный ненулевой множитель (открытая информация), m – модуль (простое число) (открытая информация). Чтобы получить максимальный период $T=m$-1, значения (8.2) должны принять все возможные значения

$$x_t=(1,2,\ldots,(m\text{-}1)).$$

Максимальный период будет при правильном выборе a и m. В нашем случае a=630 360 016, m=2 147 483 647 (выбор этих чисел осуществляется на основе рекомендаций специального раздела математики). Существуют генераторы (8.1) с большим размером чисел.

Коммерческая криптостойкость алгоритма (8.2) обеспечивается двумя факторами:

1) секретностью начального значения x_0;

2) секретным выбором номера итерации, с которого формируется секретный ключ.

Эти два секрета должна знать только одна пара (отправитель-получатель). Зная эти секреты, они смогут создавать одинаковые ключи. Чтобы обеспечить требуемую криптостойкость, эти секреты необходимо периодически обновлять.

Содержание лабораторной работы

1. Получить n целых чисел по алгоритму (8.2) и записать их в массив A.

2. Вычислить целое число i ={$random \cdot n$}; {} – операция округления до большего целого значения, *random* – функция вычисления значения случайной величины, равномерно распределенной на интервале (0, 1) (имя функции зависит от системы программирования).

3. Определить секретный ключ $k=A(i)$.

4. Дополнительно вычислить среднее значение нормированного элемента массива A: $x_s=\sum_i(A(i)/m)/n$, i=1,…,n. Это число должно быть близким к 0,5 (объяснить почему?).

5. Вывести значения следующих величин:

$$i, k, x_s.$$

Замечание 1: во всех вариантах n=3000.

Замечание 2: студенты, которые имеют затруднения при программировании алгоритма (8.2), могут получать эти числа по алгоритму: {$random \cdot (m-1)$}, {} – операция округления до большего целого значения.

Если данная лабораторная работу выполняется «вручную», то студенты изучают технологию получения ключей и должны объяснить: почему эти ключи являются коммерчески криптостойкими.

Список контрольных вопросов

1. Как осуществляется блочное симметричное шифрование.
2. Что такое угроза и уязвимость.
3. Конфиденциальность информации и методы ее защиты.
4. Чем обеспечивается криптостойкость алгоритма (8.2).
5. Какие значения может принимать ключ $k=A(i)$.
6. Что такое период T для генераторов РРСП.
7. Почему величина x_s близка к значению 0,5.

Лабораторная работа № 9
«Протокол (обмен) Диффи-Хеллмана»

Цель работы – ознакомиться с протоколом «Диффи-Хеллмана», как одним из вариантов создания общего секретного ключа при симметричном шифровании.

Введение

В двухключевых (несимметричных, асимметричных, с открытым ключом) криптосистемах адресатом, которому необходим закрытый (секретный) ключ, создаются два ключа (e, d), связанные между собой математической зависимостью (он их может также заказать). При этом один ключ генерируется, а другой вычисляется (это обеспечивается за счет математической связи между ключами).

Один ключ (e) объявляется открытым (публичным), а другой (d) – закрытым (приватным или секретным). Открытый ключ публикуется и доступен любому, кто желает послать сообщение адресату. Закрытый ключ сохраняется в тайне и находится у его создателя (владельца). В дальнейшем закрытый ключ будет называться секретным или закрытым в зависимости от ситуации.

Обозначение открытого ключа (e) условно, при использовании эллиптических функций открытый ключ – это точка на эллиптической кривой с двумя координатами (x и y).

Ключи всегда создает (заказывает) тот, кому нужен закрытый ключ, поэтому этот ключ никуда не передается в отличие от секретного ключа симметричной криптосистемы. Это важное преимущество двухключевых криптосистем.

Обмен Диффи-Хеллмана

Наиболее эффективным средством защиты конфиденциальной информации при автоматизированной ее обработке является шифрование. С точки скорости шифрования данных более эффективным средством является симметричное шифрование, когда отправитель и получатель пользуются одинаковым секретным ключом. В связи с этим, симметричное шифрование имеет недостаток, связанный с передачей секретного ключа.

Предложены различные технологии решения этой проблемы, одной из которых является выработка общего секретного ключа на основе обмена «Диффи-Хеллмана».

Диффи и Хеллман предложили одностороннюю функцию вида (дискретное возведение в степень)

$$y=a^x \bmod p, \qquad (9.1)$$

где a и p – открытые числа, обладающие некоторыми свойствами (для повышения криптостойкости p является большим простым числом); x - значение

секретного ключа; y – значение открытого ключа, $y<p$. Отправитель и получатель сначала генерируют закрытые ключи, а затем, используя функцию (9.1), вычисляют открытые. Так как функция (9.1) является односторонней, то злоумышленник не может вычислить закрытый ключ, зная открытый ключ, а также значения a и p.

Обратной функцией является дискретное логарифмирование, вычисление которой является вычислительно трудоемкой задачей.

В настоящий момент рекомендуется размер числа p не менее 1024 битов. Это число имеет более 300 десятичных знаков и это очень и очень большое число.

Считается, что с точки зрения криптостойкости, длина модуля 1024 бита соответствует длине 160 битов секретного ключа симметричной криптосистемы. Напомним, что в российских ГОСТах длина секретного ключа 256 битов.

Обозначим пару ключей отправителя (x_o, y_o), а получателя – (x_n, y_n). Обмен «Диффи-Хеллмана» заключается в следующем:

1. Отправитель (О) и получатель (П) по открытому каналу обмениваются своими открытыми ключами. Отправитель получает ключ y_n, а получатель – y_o.

2. Отправитель выполняет следующее математическое преобразование:

$$k_o = y_n^{xo} mod\ p = a^{xn \cdot xo} mod\ p = k.$$

3. Получатель выполняет следующее математическое преобразование:

$$k_n = y_o^{xn} mod\ p = a^{xn \cdot xo} mod\ p = k.$$

4. Ключ k, который вычислили отправитель и получатель, является сеансовым секретным ключом для симметричной криптосистемы. Его размер определяется размером числа p.

Замечание. Выполнение операции $a^x \ mod\ p$ выполняется в цикле от 1 до x: $[(\ldots((1 \cdot a\ mod\ p) \cdot a\ mod\ p)\ldots) \cdot a\ mod\ p]$.

При выполнении этой лабораторной работы проверяются знания по разделу «Криптография – шифрование электронных сообщений». Ниже приведены вопросы для тестирования. Далее имеется раздел с лекционным материалом.

Тестовые задания

1) Первый стандарт шифрования данных США имеет наименование:

А) AES
Б) IDEA
В) DES
Г) SEAL

2) Размер блока и размер ключа в первом стандарте шифрования данных США имеют следующие значения в битах:

А) 64 и 64
Б) 64 и 256
В) 64 и 48
Г) 64 и 56

3) Режим имитовставки это:

А) Поточный режим шифрования данных
Б) Метод контроля целостности данных средствами симметричной криптосистемы
В) Метод контроля целостности данных средствами несимметричной криптосистемы
Г) Блочный режим шифрования данных

4) Российский алгоритм ГОСТ 28147-89 позволяет:

А) Блочное шифрование данных
Б) Поточное шифрование данных
В) Формировать электронно-цифровую подпись
Г) Осуществлять контроль целостности информации

5) Укажите блочные криптосистемы с длиной блока 64 бита:

А) RSA
Б) AES
В) DES
Г) ГОСТ 28147-89

6) Возможная длина ключа стандарта криптографической защиты AES в битах:

А) 64
Б) 128
В) 192
Г) 256

7) Недостатком симметричных криптосистем по сравнению с несимметричными является их низкая производительность при шифровании данных:

А) ДА

Б) НЕТ

8) Особенностью сети Фейстеля является то, что в каждую итерацию преобразуется половина блока:

А) ДА

Б) НЕТ

9) При симметричном шифровании в режиме простой замены одинаковые исходные тексты при одинаковом ключе имеют различные зашифрованные сообщения:

А) ДА

Б) НЕТ

10) Тройной DES это криптосистема, которая:

А) Одновременно шифрует 3 блока

Б) В 3 раза быстрее шифрует текст по сравнению с DES

В) В 3 раза медленнее шифрует текст по сравнению с DES

Г) Число итераций в 3 раза меньше по сравнению с DES

11) Размер блока и размер ключа в российском стандарте шифрования данных ГОСТ 28147-89 имеют следующие значения в битах:

А) 64 и 64

Б) 64 и 256

В) 64 и 48

Г) 64 и 56

12) Стандарт криптографической защиты данных в США имеет наименование:

А) AES

Б) IDEA

В) DES

Г) SEAL

13) Укажите блочные криптосистемы с длиной блока 128 бита:

А) RSA

Б) AES

В) DES

Г) ГОСТ 28147-89

14) В архитектуре «Квадрат», используемой в криптосистеме AES, за один раунд преобразуется половина блока:

А) ДА

Б) НЕТ

15) В симметричной криптосистеме при шифровании используются различные ключи:

А) ДА

Б) НЕТ

16) В асимметричной криптосистеме при шифровании используются различные ключи:

А) ДА

Б) НЕТ

17) Укажите число итераций в алгоритме DES:

10

12

14

16

18) Укажите число итераций в российском алгоритме шифрования ГОСТ 28147-89 в режиме простой замены:

12

16

24

32

19) Укажите криптосистемы, созданные на основе сети Фейстела:

А) RSA

Б) AES

В) DES

Г) ГОСТ 28147-89

20) Укажите число возможных раундов в стандарте AES:

10

12

14

16

21) В режиме имитовставки получатель сравнивает свою имитовставку с имитовставкой отправителя и если они совпадают, то он считает, что конфиденциальность сообщения не нарушена:

А) ДА

Б) НЕТ

22) В симметричной блочной криптосистеме при шифровании сообщение в обязательном порядке разбивается на блоки длиной в размер ключа:
А) ДА
Б) НЕТ

23) При шифровании симметричной криптосистемой секретный ключ создает получатель:
А) ДА
Б) НЕТ

24) Алгоритм DES может использоваться для создания ключа при поточном шифровании данных:
А) ДА
Б) НЕТ

25) Укажите блочные криптосистемы с длиной блока 128 битов:
А) Магма
Б) AES
В) Кузнечик
Г) ГОСТ 28147-89

26) В ГОСТ 34.12-2018 описаны два шифра: «Магма» и «Кузнечик»:
А) ДА
Б) НЕТ

27) В шифре «Кузнечик» размер блока и размер ключа имеют следующие значения в битах:
А) 64 и 64
Б) 128 и 256
В) 64 и 256
Г) 64 и 56

28) В шифре «Магма» размер блока и размер ключа имеют следующие значения в битах:
А) 64 и 64
Б) 128 и 256
В) 64 и 256
Г) 64 и 56

29) Укажите число раундов в шифре «Кузнечик»:
10
12

14
16

30) В ГОСТ 34.13-2018 реализовано следующее число режимов работы:
4
6
8
5

Лекционный материал по лабораторной работе

Шифрование информации заключается в криптографическом преобразовании данных с применением дополнительной информации, которая называется ключом. Отправитель зашифровывает текст (преобразует исходный текст в зашифрованный) и отправляет его по открытому каналу, а получатель расшифровывает текст (преобразует зашифрованный текст в исходный).

Открытость канала связи отправителя и получателя означает, что к нему имеет доступ злоумышленник.

Если при зашифровании и расшифровании используется один и тот же секретный ключ, то говорят о симметричном шифровании, а если используется два ключа (один для зашифрования, а другой для расшифрования), то говорят об асимметричном (несимметричном) шифровании.

По сравнению с двухключевыми криптосистемами, симметричные при шифровании данных более производительные.

Симметричные криптосистемы при шифровании данных используют две технологии:

а) поточное шифрование;

б) блочное шифрование.

При поточном шифровании вырабатывается гамма (секретный ключ), как правило длиною равной шифруемому сообщению. При зашифровании и расшифровании используется операция сложения по модулю 2 (см. лабораторную работу № 7).

При блочном шифровании исходное сообщение (исходный текст) предварительно разбивается на блоки длиной *w* битов, а затем зашифровывается (зашифрованный текст) с применением ключа *k* длиной *n* битов.

При блочном шифровании применяются различные режимы:

1) режим простой замены – *ECB*, когда блоки шифруются независимо друг от друга

$$c_i=E_k(m_i),\ \ m_i= E^{-1}_k(c_i),\ i=1,\ldots,I. \qquad (9.2)$$

В формуле (9.2) E_k и E^{-1}_k – функции зашифрования и расшифрования с ключом k; m_i – исходный блок; c_i – зашифрованный блок, i – номер блока. В режиме простой замены одинаковые исходные тексты при одинаковом ключе имеют одинаковый зашифрованный текст, что является их недостатком;

2) режим сцепления блоков – *CBC*, когда каждый последующий блок (m_i) складывется по модулю 2 с предыдущим зашифрованным блоком, а затем шифруется (c_i). Зашифрование и расшифрование описываются преобразованиями

$$c_i=E_k(c_{i-1}(+)m_i),\ \ m_i=c_{i-1}(+)E^{-1}_k(c_i); \qquad (9.3)$$

3) режимы шифрования с обратной связью (*CFB, OFB*) и их модификации.

Положительной стороной шифрования в режиме взаимодействия блоков является то, что последний зашифрованный блок несет информацию о целостности сообщения. Этот факт используется при создании имитовставки.

Особенностью блочных алгоритмов шифрования является их итерационность, когда при шифровании блока многократно используются одни и теже криптографические преобразования. Эти однотипные операции называют итерациями или раундами.

Первым стандартом шифрования данных является алгоритм *DES*, который был введен в США в 1977 году. Данный алгоритм имеет такие показатели: длина блока 64 бита, длина ключа 56 битов, число итераций – 16. При шифровании данных используются различные режимы: простой замены, режим с обратной связью и т.д. Возможно применение алгоритма *DES* и для поточного шифрования.

Алгоритм *DES* оказался весьма успешным, но из-за короткого ключа со временем стал не криптостойким. Было предложено несколько усовершенствований этого алгоритма. Наиболее распространенным является тройной *DES*, когда каждый блок зашифровывается, расшифровывается и снова зашифровывается, поэтому время шифрования для него в три раза больше, чем у *DES*. Используется две основных модификации:

а) с двумя ключами - $k_1\ k_2\ k_1$;

б) с тремя ключами - $k_1\ k_2\ k_3$.

Криптостойкость второй модификации выше, чем у первой.

В 1989 году был принят Российский алгоритм шифрования данных – ГОСТ 28147-89, который имеет такие показатели: длина блока 64 битов, длина ключа 256 битов, число итераций – 32.

Российский алгоритм предусматривает три режима шифрования: простой замены, гаммирования и гаммирования с обратной связью. Последние два режима обеспечивают поточное шифрование. Они предусматривают сложение по модулю 2 исходного сообщения с гаммой. При выработке гаммы используется не секретная и секретная информация. Блоки поточного ключа создаются технологией зашифрования в режиме простой замены.

Российский алгоритм может работать также в режиме имитовставки, когда для исходного сообщения шифрованием с применением секретного ключа вырабатывается специальная информация – имитовставка, которая затем используется для проверки целостности сообщения. Для этого отправитель отправляет получателю зашифрованное сообщение и имитовставку. Получатель расшифровывает сообщение, а затем по сообщению создает свою

имитовставку и сравнивает ее с имитовставкой отправителя. Если они совпали, то считается, что целостность сообщения не нарушена. При этом получатель должен получить секретный ключ и некоторую дополнительную информацию.

Алгоритмы *DES* и ГОСТ 28147-89 созданы в соответствии с принципами сети Фейстела, особенностью которой является то, что в каждой итерации преобразуется лишь половина блока. Это требует при шифровании достаточно большого числа итераций.

В связи с развитием мощности компьютеров, средств криптоанализа, первый стандарт шифрования данных оказался не криптостойким, поэтому в 2002 году в США был введен новый стандарт криптографической защиты (*AES*), который основан на алгоритме *Rijndael* и использует архитектуру «Квадрат». В этом случае за один раунд преобразуется весь блок.

Стандарт криптографической защиты *AES* имеет такие показатели: длина блока 128 битов; длина ключа 128 битов и число раундов – 10; длина ключа 192 битов и число раундов – 12; длина ключа 256 битов и число раундов – 14. Таким образом, разработчики стандарта *AES* учли опыт стандарта *DES* и заложили сразу три ключа по их длине, что позволит более длительное время сохранять его криптостойкость.

В настоящее время в РФ введен в действие ГОСТ 34.12-2018 «Информационная технология. Криптографическая защита информации. Блочные шифры», который является межгосударственным стандартом для Армении, Киргизии, России и Таджикистана. Настоящий стандарт подготовлен на основе применения ГОСТ Р 34.12-2015.

ГОСТ 34.12-2018 внесен Техническим комитетом по стандартизации ТК 26 «Криптографическая защита информации».

Приказом Федерального агентства по техническому регулированию и метрологии от 4 декабря 2018 г. No 1061-ст межгосударственный стандарт ГОСТ 34.12-2018 введен в действие в качестве национального стандарта Российской Федерации с 1 июня 2019 г.

В ГОСТ Р 34.12-2018 описаны два шифра:

1) «Магма» – это блочный шифр по ГОСТ 28147-89 (блочный шифр в режиме *ECB*);

2) «Кузнечик» – это новый блочный шифр в режиме *ECB*, реализующий подстановочно-перестановочную сеть (*SP*-сеть).

Параметры блочного шифра «Кузнечик» следующие:

– длина блока (w) 128 битов (16 байт),

– длина ключа (n) 256 битов,

– число раундов (r) – 10.

Число раундов значительно меньше, чем число итераций в шифре «Магма». Это связано с тем, что в шифре «Кузнечик» используется схема, когда в каждом раунде преобразуется весь блок.

Число раундов меньше, чем число раундов системы *AES* при длине ключа 256 битов (14). Это связано с лучшим набором криптографических операций функции шифрования.

В шифре «Кузнечик» имеется специальная процедура, которая из секретного ключа (k) создает 10 различных раундовых ключа по 128 битов каждый.

Задание на лабораторную работу

Для этой лабораторной работы используется специальная программа. В этой программе лабораторная работа имеет номер 3.

1. Студент вводит в систему свой вариант. При этом выводятся исходные данные своего варианта. Параметры a и p описаны в формуле (9.1).

2. Необходимо пройти тестирование.

3. Если тест не пройден, то результаты лабораторной работы не будут выведены. Необходимо изучить теоретический материал и пройти повторное тестирование. В течение одного занятия студент может выполнить не более 3-х попыток.

4. Если тест пройден, то необходимо продолжить лабораторную работу. Выводятся результаты работы, которую необходимо защитить.

Лабораторная работа № 11
«Шифрование сообщений криптосистемой *RSA*»

Цель работы – ознакомиться с технологией асимметричного шифрования на примере алгоритма RSA.

Введение

В 1976 году произошло важное событие, существенно изменившее приложения прикладной криптографии. Этим событием является статья У. Диффи и М. Хеллмана «Новые направления в криптографии», которая заложила основы двухключевой (асимметричной, несимметричной) криптографии. В этом классе методов криптографии один ключ является открытым (не секретным), а другой закрытым (секретным). Эти ключи связаны математической зависимостью и создаются одновременно: один из ключей генерируется (выбирается), а другой после этого вычисляется.

Закрытый ключ всегда находится у той стороны, кому он нужен (эта сторона и создает пару ключей), поэтому отсутствует проблема передачи секретного ключа. Открытый ключ передается другой стороне по открытому каналу.

Помимо шифрования (обеспечение конфиденциальности информации) этот класс криптоалгоритмов успешно используется:

а) при создании электронно-цифровой подписи (ЭЦП),

б) в задачах аутентификации, как партнеров, так и сообщений,

в) при контроле целостности данных отдельного сообщения или потока сообщений,

г) при доказательстве принадлежности в случае отказа от переданного или принятого сообщения.

Одной из первых асимметричных криптосистем является система *RSA* (1977 год), получившая название по начальным буквам фамилий ее создателей (*Rivest, Shamir, Adleman*).

Описание алгоритма RSA

Односторонняя функция алгоритма *RSA* имеет вид

$$n=p\cdot q, \qquad (11.1)$$

где n – модуль (открытая информация); p и q – большие простые числа (секретная информация).

Для алгоритма Эль-Гамаля, который положен в основу российского стандарта на электронно-цифровую подпись, односторонней функцией является дискретное возведение в степень (9.1).

Открытый (e) и закрытый (d) ключи создаются по следующей технологии:

- Выбираем два больших простых числа p и q. Одно из требований к выбору p и q заключается в том, что по крайней мере одно из чисел p-1 или q-1 должно иметь один большой простой множитель.

- Определяем $n = p \cdot q$. Размер модуля n должен на современном этапе с позиции криптостойкости быть не менее 1024 битов, желательно 2048 битов или более.

- Вычисляем значение

$$k = (p - 1)(q - 1). \qquad (11.2)$$

Число k определяет количество целых чисел меньших n и взаимно простых с ним, когда $n = p \cdot q$ (взаимно простое число – это число, которое не имеет ни одного общего делителя, кроме числа 1); k называют функцией Эйлера.

- Выбираем открытый ключ шифрования e так, чтобы k и e были взаимно простыми числами.

- Вычисляем число d, для которого истинным является соотношение

$$(e \cdot d) \bmod k = 1. \qquad (11.3)$$

При решении (11.3) можно использовать расширенный алгоритм Евклида. Решение (11.3) единственно, т.к. ключ e и k взаимно простые числа.

- Принимаем в качестве открытого ключа пару чисел {e, n}.
- В качестве секретного ключа принимаем число d.

Замечание. После вычисления n и k, величины p и q рекомендуется уничтожить для повышения криптостойкости системы.

Для зашифрования и расшифрования передаваемых сообщений с помощью алгоритма *RSA* необходимо выполнить следующие операции:

1. Разбить шифруемый текст на числовые блоки, каждый из которых может быть представлен в виде числа m_i меньшего n.

2. Зашифровать последовательно каждый исходный блок m_i, используя открытый ключ $\{e, n\}$, по формуле

$$c_i = (m_i^{\,e}) \bmod n. \qquad (11.4)$$

Последовательность чисел $c_1 c_2 \ldots$ является зашифрованным текстом.

3. Расшифровать последовательно каждый блок зашифрованного текста c_i, используя закрытый ключ d, по формуле

$$m_i = (c_i^{\,d}) \bmod n. \qquad (11.5)$$

Последовательность чисел $m_1 m_2 \ldots$ является расшифрованным текстом.

Замечание. Криптостойкость алгоритма *RSA* базируется на вычислительной сложности определения сомножителей p и q, зная модуль n. Найдя числа p и q, злоумышленник найдет число k (11.2) и закрытый ключ d (11.3). На практике при использовании метода *RSA* длина p и q составляет сотни десятичных знаков, что и обеспечивает высокую криптостойкость алгоритма.

Тестовые задания

При выполнении этой лабораторной работы проверяются знания по разделу «Криптография – двухключевые криптосистемы и их применение». Ниже приведены вопросы для тестирования.

1) Укажите двухключевую криптосистему:

А) DES

Б) RSA

В) AES

Г) ГОСТ 28147-89

2) Укажите одностороннюю функцию для алгоритма *RSA*:

А) $(e \cdot d) \bmod k = 1;\ k=p \cdot (q-1);$

Б) $y = a^x \bmod p;$

В) $n = p \cdot q;$

Г) $c = m^e \bmod (n-1);$

3) При зашифровании данных несимметричной криптосистемой, используется:

А) секретный ключ

Б) открытый ключ

В) сначала открытый, а затем секретный ключ

Г) сначала секретный, а затем открытый ключ

4) Размер хэш-образа по российскому стандарту (ГОСТ-94) равен:

А) 256 бит или 512 бит

Б) 256 бит

В) 160 бит

Г) 320 бит

5) Двухключевая криптосистема по сравнению с одноключевой имеет более высокую производительность при шифровании данных:

А) ДА

Б) НЕТ

6) Современный протокол шифрования данных базируется на совместном применении как симметричной криптосистемы так и несимметричной:

А) ДА

Б) НЕТ

7) Хэш-функция – это криптографическое преобразование информации, переводящее строку битов произвольной длины в строку битов фиксированной длины:

А) ДА
Б) НЕТ

8) В электронном документообороте наилучшим способом контроля целостности данных является:
А) шифрование
Б) электронная цифровая подпись
В) хэширование

9) Увеличение длины ЭЦП в 2 раза по сравнению с хэш-образом это свойство ЭЦП:
А) ДА
Б) НЕТ

10) Размер ЭЦП по российскому стандарту (ГОСТ-2001) равен:
А) 256 бит 512 бит
Б) 512 бит
В) 1024 бит
Г) 320 бит

11) При шифровании данных несимметричной криптосистемой, используется:
А) секретный ключ
Б) открытый ключ
В) сначала открытый, а затем секретный ключ
Г) сначала секретный, а затем открытый ключ

12) При создании ЭЦП секретный ключ по длине больше, чем открытый:
А) ДА
Б) НЕТ

13) Размер хэш-образа по американскому стандарту *SHA*-1 равен:
А) 256 бит
Б) 512 бит
В) 160 бит
Г) 320 бит

14) В современном протоколе шифрования данных при шифровании сообщений используется двухключевая криптосистема:
А) ДА
Б) НЕТ

15) Размер ЭЦП по американскому стандарту *DSS* равен:

А) 256 бит
Б) 512 бит
В) 1024 бит
Г) 320 бит

16) В электронном документообороте наилучшим способом обеспечения конфиденциальности данных является:
А) шифрование
Б) электронная цифровая подпись
В) хэширование

17) Укажите одностороннюю функцию для алгоритма Эль-Гамаля:
А) $(e \cdot d) \bmod k = 1;\ k=p \cdot (q-1);$
Б) $y = a^x \bmod p;$
В) $n = p \cdot q;$
Г) $c = m^e \bmod (n-1);$

18) Выберите наиболее эффективную криптосистему для шифрования незначительных по объему данных:
А) RSA
Б) AES
В) DES
Г) ГОСТ 28147-89

19) Двухключевую криптосистему называют криптосистемой с открытым ключом, т.к. при шифровании открытый ключ создает отправитель, а секретный – получатель:
А) ДА
Б) НЕТ

20) Пару ключей при шифровании данных криптосистемой с открытым ключом создает отправитель:
А) ДА
Б) НЕТ

21) Владельцем пары ключей при создании ЭЦП является отправитель:
А) ДА
Б) НЕТ

22) Хэширование сообщения при создании ЭЦП осуществляется, чтобы:
А) обеспечить конфиденциальность информации
Б) увеличить время создания ЭЦП
В) стандартизовать время создания ЭЦП и ее размер

23) Хэш-функция может применяться в следующих случаях:

А) для создания сжатого образа сообщения, используемого в механизме цифровой подписи;

Б) для защиты пароля;

В) при контроле целостности данных;

Г) для сохранения конфиденциальности информации.

24) Коллизия в хэш-функции, это когда разные сообщения имеют одинаковый хэш-образ:

А) ДА

Б) НЕТ

25) В современном протоколе шифрования данных для шифрования секретного ключа симмеричной криптосистемы используется двухключевая криптосистема:

А) ДА

Б) НЕТ

26) Двухключевая криптосистема может применяться в следующих случаях:

А) для шифрования небольших по объему данных;

Б) при создании электронно-цифровой подписи;

В) в задачах аутентификации;

Г) для обеспечения доступности информации.

27) Размер хэш-образа по российскому стандарту (ГОСТ-2018) равен:

А) 256 бит или 512 бит

Б) 512 бит

В) 160 бит

Г) 256 бит

28) Размер ЭЦП по российскому стандарту (ГОСТ-2018) равен:

А) 256 бит 512 бит

Б) 512 бит или 1024 бит

В) 1024 бит

Г) 512 бит

29) Размер хэш-образа и блока по стандарту США SHA-512 равны:

А) 256 бит и 512 бит

Б) 512 бит и 256 бит

В) 512 бит и 1024 бит

Г) 512 бит и 512 бит

Как уже отмечалось, криптосистемы по числу ключей делятся на симметричные (одноключевые) и асимметричные (двухключевые).

При шифровании асимметричной криптосистемой пару ключей создает получатель, секретный ключ хранит у себя, а открытый (не секретный) ключ по открытому каналу передается отправителю. При зашифровании отправитель открытым ключом шифрует сообщение и передает его по каналу получателю. Получатель секретным ключом расшифровывает полученное сообщение. В связи с этим, двухключевые криптосистемы называют криптосистемами с открытым ключом.

Их положительной стороной является отсутствие проблемы передачи секретного ключа, а недостатком – низкая производительность по сравнению с симметричными криптосистемами. Поэтому асимметричные криптосистемы используются при шифровании коротких сообщений (ключей, хэш-образов и т.д.).

Положительные стороны одноключевых и двухключевых криптосистем положены в основу современного протокола шифрования данных, когда они используются совместно. Симметричная криптосистема используется при шифровании сообщения, а двухключевая криптосистема используется при зашифровании и расшифровании секретного ключа симметричной криптосистемы. Секретный ключ симметричной криптосистемы создается отправителем, а пара ключей для асимметричной криптосистемы – получателем. Отправитель отправляет получателю зашифрованный текст и зашифрованный ключ.

Одной из первых асимметричных криптосистем является система *RSA*. Как мы уже отмечали, односторонняя функция алгоритма *RSA* имеет вид (11.1).

Пара чисел $\{e, n\}$ является открытым ключом, число d – секретным ключом. Технология получения этих ключей, а также технология зашифрования и расшифрования описана в начале этой лабораторной работы (11.2-11.6).

Асимметричные криптосистемы используются не только для шифрования данных, но и в других приложениях. Важным приложениям является создание электронно-цифровой подписи (ЭЦП).

При создании ЭЦП сообщение предварительно подвергается преобразованию хэш-функцией.

Хэш-функция – это криптографическое преобразование информации, переводящее строку битов произвольной длины в строку битов фиксированной длины. Совпадение значений хэш-функции для разных сообщений называют коллизией.

Хэш-функция $y=h(m)$ должна обладать двумя основными свойствами:

1) для данного значения y вычислительно трудоемко найти аргумент m;

2) для данного аргумента *m* вычислительно трудоемко найти другой аргумент *m*' такой, что *h(m) = h(m').*

При выполнении этих двух свойств хэш-функция является односторонней и свободной от коллизий. Коллизии в принципе возможны, т.к. мощность образов меньше мощности сообщений. Но так как хэш-функция является односторонней, специально это сделать вычислительно трудоемко.

Криптографические хэш-функции подразделяются на два класса: с ключом и без ключа. Значение хэш-функции с ключом может вычислить лишь тот, кто знает некоторый секретный параметр-ключ. Часто в литературе они называются *MAC* – кодами аутентификации сообщений. Технология получения имитовставки в ГОСТе 28147-89 может рассматриваться как хэш-функция с ключом.

Хэш-функция имеет многоцелевое использование, например:

- для создания сжатого образа сообщения, применяемого в механизме цифровой подписи;
- для защиты пароля;
- для построения кода аутентификации сообщений (*MAC*), но в этом случае используются хэш-функции с ключом.

Операция хэширования нужна в ЭЦП для того, чтобы стандартизовать время ее создания и проверки, т.к. для больших сообщений это время может оказаться очень значительным, а также размер ЭЦП.

В функции хэширования ГОСТ Р 34.11-94 размер хэш-образа равен 256 битов. Алгоритм по ГОСТ Р 34.11-94 является итерационным с размером блока 256 битов. Он использует блочный алгоритм шифрования ГОСТ 28147-89 для формирования хэш-образа, при этом дополнительно к блокам сообщения формируется еще два блока: блок, содержащий информацию о длине сообщения, и блок, содержащий контрольную сумму сообщения.

В американском стандарте на хэш-функцию *SHA*-1 размер блока равен 512 битов, а размер хэш-образа равен 160 битов, в связи с этим, российская хэш-функция считается более криптостойкой. В настоящий момент используется стандарт *SHA512* с размером хэш-образа 512 битов и размером блока 1024 бита.

В настоящий момент в РФ используется хэш-функция «Стрибог» по ГОСТ 34.11-2018 (ГОСТ-2018). В этом стандарте предусмотрено два режима, когда размер хэш-функции равен 256 или 512 битов, размер блока равен 512 битов. Основное отличие от предыдущего алгоритма связано с шаговой функцией, которая в ГОСТ 34.11-2018 называется функцией сжатия.

Электронная ЦП – реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе (ФЗ от 2002 года).

Электронная подпись – информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для определения лица, подписывающего информацию (ФЗ № 63 от 2011 года).

Видами электронных подписей, отношения в области использования которых регулируются настоящим Федеральным законом, являются простая электронная подпись и усиленная электронная подпись. Различаются усиленная неквалифицированная электронная подпись (далее - неквалифицированная электронная подпись) и усиленная квалифицированная электронная подпись (далее - квалифицированная электронная подпись).

Простой электронной подписью является электронная подпись, которая посредством использования кодов, паролей или иных средств подтверждает факт формирования электронной подписи определенным лицом.

Подтверждение подлинности ЭЦП в электронном документе - положительный результат проверки соответствующим сертифицированным средством ЭЦП с использованием сертификата ключа подписи принадлежности ЭЦП в электронном документе владельцу сертификата ключа подписи и отсутствия искажений в подписанном данной ЭЦП электронном документе.

Технология применения ЭЦП в электронном документообороте следующая:

1) отправитель, который является владельцем пары ключей асимметричной криптосистемы (закрытого ключа подписи и открытого ключа проверки подлинности ЭЦП):

а) передает через сертификат открытый ключ получателю;

б) подписывает сообщение, для чего сообщение хэшируется, а полученный хэш-образ зашифровывается закрытым ключом;

в) передает по открытому каналу получателю сообщение и ЭЦП;

2) получатель проверяет подлинность полученного сообщения, для чего:

а) используя сообщение, ЭЦП и открытый ключ, создает свою компоненту ЭЦП;

б) если компоненты ЭЦП отправителя и получателя совпадают, то это означает подтверждение подлинности ЭЦП. А это означает целостность полученного сообщения и идентификация отправителя (свойство неотказуемости).

ЭЦП является наилучшим средством контроля целостности данных в электронном документообороте. Напомним, что наилучшим средством обеспечения конфиденциальности данных в электронном документообороте является шифрование.

В соответствии со стандартом на ЭП ГОСТ 34.10-2018 (ГОСТ-2018) размер ЭЦП равен 512 или 1024 битов. Этот ГОСТ использует хэш-функцию по ГОСТ 34.11-2018.

Удвоение длины ЭП по сравнению с длиной хэш-образа связано с особенностью алгоритма двухключевой криптосистемы, используемой в стандарте. В американском стандарте *DSS* размер ЭЦП равен 320 битов.

И в российском, и в американском стандартах на ЭП секретный ключ короче открытого. Это сделано для того, чтобы трудоемкость «подписания» сообщения была меньше, чем проверка подлинности. А это, в свою очередь, позволяет применять при «подписании» сообщений маломощные компьютеры.

Задание на лабораторную работу

При выполнении этой работы используется специальная программа. В этой программе лабораторная работа имеет номер 4.

1. Студент вводит в систему свой вариант (табл. 11.1). При этом выводятся исходные данные своего варианта. Параметры p и q описаны в формуле (11.1); e – открытый ключ; буква – вставляемая в конце исходного текста при необходимости буква; текст – номер исходного текста (рис. 11.1).

2. Необходимо пройти тестирование.

3. Если тест не пройден, то результаты лабораторной работы не будут выведены. Необходимо изучить теоретический материал и пройти повторное тестирование. В течение одного занятия студент может выполнить не более 3-х попыток.

4. Если тест пройден, то необходимо продолжить лабораторную работу. Выводятся результаты работы и необходимо их защитить.

Замечание. Для студентов умеющих программировать предусмотрено выполнение этой работы с написанием программы и сравнения полученных результатов с типовой лабораторной работой.

Исходные тексты

1. Современная государственная политика РФ в области защиты информации сформировалась в начале девяностых годов и базируется на соблюдении баланса интересов личности, общества и государства в информационной сфере.

2. Хотя сама информация не материальна, но она имеет материальные носители. Говоря о мере информации, выделяют ее количество и объем. Объем данных в сообщении измеряется количеством символов принятого алфавита.

3. Информатика как наука изучает свойства, структуру и функции информационных систем, основы их проектирования, создания, использования и оценки, а также информационные процессы в них происходящие.

4. Под информационной системой понимают систему, организующую, хранящую и преобразующую информацию. В этой системе основным предметом и продуктом труда является информация.

5. Под безопасностью информации понимают свойство передаваемой, накапливаемой, обрабатываемой и хранимой информации, характеризующее

ее степень защищенности от дестабилизирующего воздействия внешней среды и внутренних угроз.

Рис. 11.1. Исходные тексты

Варианты лабораторной работы

Таблица11.1

№	*p*	*q*	*e*	Буква	Текст
1	67	53	2005	А	1
2	53	67	2501	Б	1
3	67	53	1999	В	2
4	53	67	2503	Г	2
5	67	53	1999	Д	3
6	53	67	2503	Е	3
7	67	53	2005	Ж	4
8	53	67	2501	З	4
9	83	53	1999	И	5
10	53	83	2505	Й	5
11	83	53	1999	К	4
12	53	83	2505	Л	4
13	83	53	2003	М	3
14	53	83	2505	Н	3
15	83	53	2003	О	2
16	53	83	2505	П	2
17	79	41	1999	Р	1
18	41	79	2501	С	1
19	79	41	1999	Т	2
20	41	79	2501	У	2
21	79	41	2003	Ф	3
22	41	79	2503	Х	3
23	79	41	2003	Ц	4
24	41	79	2503	Ч	4
25	67	53	2017	А	5
26	53	67	2507	Б	5
27	67	53	2021	В	4
28	53	67	2521	Г	4
29	67	53	2027	Д	3
30	53	67	2537	Е	3
31	67	53	2033	Ж	2
32	53	67	2543	З	2

Практическое занятие № 1
«Алгоритм криптографической защиты, ГОСТ 28147-89»

Данное практическое занятие рекомендуется провести после 2-й лабораторной работы.

Лекционное содержание занятия

В нашей стране в конце 80-х гг. XX в. был принят алгоритм криптографического преобразования ГОСТ 28147-89 (алгоритм ГОСТ). Данный алгоритм является национальным стандартом и предназначен для систем обработки информации, удовлетворяет криптографическим требованиям и по своим возможностям не накладывает ограничений на степень секретности защищаемой информации.

Отметим, как и в системе *AES*, в российском стандарте отсутствуют слова: блочный, шифрования. Поэтому название этого алгоритма типа «блочный алгоритм шифрования» не корректно, хотя используется в учебной литературе.

Алгоритм ГОСТ имеет четыре режима:

1) режим простой замены – блочное шифрование в режиме *ECB*, основной режим;
2) режим гаммирования – поточное шифрование;
3) режим гаммирования с обратной связью – поточное шифрование;
4) режим выработки имитовставки – контроль целостности передаваемого зашифрованного сообщения при преднамеренных воздействиях.

В связи с этим, российский алгоритм – это не алгоритм шифрования, а алгоритм криптографической защиты, который помимо шифрования обеспечивает защиту целостности передаваемых сообщений.

ГОСТ введен 01.07.1990, до 1994 г. имел гриф ДСП (для служебного пользования).

Рассмотрим эти режимы.

Шифр блочного шифрования в режиме ECB

Из четырех режимов базовый – блочный режим простой замены. Он используется во всех других режимах. Блочный режим основан на схеме Фейстеля (4.16), но *f*-функция отлична от алгоритма *DES*.

Алгоритм блочного шифрования в режиме простой замены имеет:

– длину блока (w) 64 бита (8 байт),

– длину ключа (n) 256 битов,

– число итераций (r) –32.

Если длина исходного сообщения не кратна длине блока, то выполняют процедуру дополнения, а затем обратную процедуру. Большое число итераций связано с применением схемы Фейстеля. Число эффективных итераций,

пересчитанных на преобразование блока, равно 16 и это соизмеримо с числом раундов системы *AES* при длине ключа 256 битов (14).

Объем ключевого пространства (4.3) составляет 2^{256}, с учетом (4.17) и n= 256 битов, число десятичных знаков p=76, поэтому число ключей больше 10^{76}.

Это очень и очень большое число. Ни на одной из существующих в настоящее время или предполагаемых к реализации в недалеком будущем ЭВМ общего применения нельзя подобрать ключ простым перебором.

Зашифрование и расшифровывание блоков в режиме *ECB* описываются преобразованиями (4.12).

Ключ российского алгоритма – это массив, состоящий из восьми 32-битных подблоков (G_s): G_1, G_2, ... G_8). Циклический (итерационный) ключ j-го цикла k_j равен G_s, где ряду значений j от 1 до 32 соответствуют следующие значения s:

1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8,
1, 2, 3, 4, 5, 6, 7, 8, 8, 7, 6, 5,4, 3, 2, 1. (4.19)

Рассмотрим работу f-функции схемы Фейстеля для российского блочного режима:

1) 32-битный блок (R_{j-1}) складывается по модулю 2^{32} с циклическим ключом k_j, имеющим номер (4.19) для выбора 32-битного подблока (G_s), результат 32-битный полублок (X_j);

2) 32-битный полублок (X_j) разбивается на восемь частей по четыре бита. Каждая 4-битовая часть преобразуется через свой S-блок. Все восемь этих S-блоков состоят из 4-битовых чисел в диапазоне от 0 до 15, последовательность чисел своя и выбрана специальным образом. Пример возможного S-блока:

(4 10 9 2 13 8 0 14 6 11 1 12 7 15 5 3).

S-блоки выполняют нелинейные подстановки. Например, если первое 4- битное число (тетрада) в полублоке X_j равно 0, то оно заменится на 4, если второе число равно 7, то оно заменится на 14 и т.д. (тетрады являются значениями индексов массива, а замены – элементами массива).

Элементы массивов со всех восьми S-блоков создадут 32-битный полублок (Y_j);

3) 32-битный полублок (Y_j) сдвигается циклично влево на 11 битов (старшие биты полублока становятся младшими. В результате циклического сдвига получается 32-битный результат f-функции схемы Фейстеля для российского блочного режима.

В последней 32-й итерации полублоки меняются местами: L_r,$R_r \rightarrow R_r$, L_r.

В общем случае S-блоки российского ГОСТа не являются открытой информацией, что является его особенностью (принцип Керкгоффса не выполняется). Хотя имеется рекомендация в стандарте ГОСТ Р 34.11-94 (функция хэширования при создании которой используется ГОСТ в режиме простой замены). Эти восемь S-блоков следующие (табл. 4.1).

Таблица 4.1

Рекомендуемые значения *S*-блоков

0	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
1	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
2	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
3	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
4	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
5	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
6	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
7	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

S-блоки (табл. 4.1) используются в различных государственных (банковских) криптосистемах, использующих ГОСТ. Алгоритм расшифрования отличается от алгоритма зашифрования тем, что последовательность циклических ключей используется в обратном порядке.

Подчеркнем, что блочный режим простой замены ГОСТа является основным, так как используется в остальных трех режимах. Он за счет большой длины секретного ключа (256 битов) и хорошо организованным набором криптоопераций, включая выбор последовательности итерационных ключей, является очень криптостойким шифром даже при известных *S*-блоках.

Режим гаммирования

Режим гаммирования является поточным шифрованием и предусматривает сложение по модулю 2 исходного или зашифрованного сообщений с гаммой (поточным ключом), как описано в параграфе 4.2.

В данном случае мы опишем, как в этом режиме создается поточный ключ или гамма.

При выработке гаммы используются две специальные 32-битные не секретные константы (C_1 и C_2) и 64-битная синхропосылка (K_o), которая может быть секретной или не секретной и используемой один раз. Гамма создается блоками по 64 бита с использованием блочного шифра в режиме простой замены, рассмотренного выше (E_k).

Пусть необходимо зашифровать сообщение $m=m_1m_2...m_i..., m_I$, где m_i – i-й блок исходного сообщения; I – номер последнего блока исходного сообщения. Выберем синхропосылку K_0 длиной 64 бита и ключ k стандарта ГОСТ (эта синхропосылка нужна для получения первого блока поточного ключа).

Поточный ключ (гамма) создается блоками по 64 бита $K=K_1K_2,...,K_i,...,K_I$ нужной длины (по длине исходного сообщения). Блок поточного ключа (K_i) создается в два шага, $i=1, 2,...,I$:

1) На первом шаге создается промежуточное значение из предыдущего блока поточного ключа

$$K^*_i=E_k(K_{i-1}), \qquad (4.20)$$

Здесь E_k – функция зашифрования по алгоритму ГОСТ в режиме простой замены, k – секретный ключ этого режима;

2) На втором шаге полученное значение (K^*_i) рассматривается как два полублока по 32 бита. Первый полублок складывается по модулю 2^{32} с 16-ричной константой $C_1=1010101_{16}$; второй полублок складывается по модулю 2^{32}-1 с 16-ричной константой $C_2=1010104_{16}$.

В результате получается блок поточного ключа (K_i). Этот блок используется для поточного шифрования и для создания следующего блока поточного ключа (4.20). При расшифровании гамма вырабатывается аналогичным образом, поэтому синхропосылка должна быть передана получателю (при секретной синхрапосылке к секретному ключу добавляется 64 бита и он становится длиной 320 битов).

В режиме гаммирования поточное шифрование можно осуществлять двумя способами:

1) создать сначала весь поточный ключ, затем проводить зашифрование/расшифрование;

2) создавать поточный ключ блоками и блоками проводить зашифрование/расшифрование (4.4), (4.5).

Режим гаммирования с обратной связью

Пусть необходимо зашифровать сообщение $m=m_1m_2...m_i..., m_I$, где m_i – i-й блок исходного сообщения; I – номер последнего блока исходного сообщения. Выберем синхропосылку K_0 длиной 64 бита и ключ k стандарта ГОСТ (эта синхропосылка нужна для получения первого блока поточного ключа).

Поточный ключ (гамма) создается блоками по 64 бита $K=K_1K_2,...,K_i,\ldots,K_I$ нужной длины (по длине исходного сообщения).

В режиме гаммирования с обратной связью первый блок поточного ключа $K_1=E_k(K_0)$, последующие блоки

$$K_i=E_k(c_{i-1}),\ c_{i-1}=K_{i-1}(+)m_{i-1},\ i=2,3,\ldots I, \qquad (4.21)$$

где E_k – функция зашифрования по алгоритму ГОСТ в режиме простой замены, k – секретный ключ этого режима, (+) – сложение по модулю 2.

При расшифровании поточный ключ получается аналогичным образом

$$K_1=E_k(K_0),\ K_i=E_k(c_{i-1}),\ i=2,3,\ldots,I.$$

В режиме гаммирования с обратной связью поточное зашифрование можно осуществлять только одним способом: создавать поточный ключ блоками и блоками проводить зашифрование (4.4). Это связано с тем, что при создании поточного ключа в этом режиме используются блоки зашифрованного сообщения (4.21). Расшифрование можно проводить двумя способами, как и в режиме гаммирования.

Сравнивая эти режимы поточного шифрования, следует отметить следующее:

1) режим гаммирования: если при передаче зашифрованного сообщения в каких-то блоках произошло изменение некоторых битов, то эти изменения не увеличиваются в процессе расшифрования. Эти изменения сохраняются на тех

же местах в расшифрованном сообщении. Это связано с тем, что в этом режиме при создании поточного ключа не используются блоки зашифрованного сообщения (4.20);

2) режим гаммирования с обратной связью: если при передаче зашифрованного сообщения в каких-то блоках произошло изменение некоторых битов, то эти изменения увеличиваются в процессе расшифрования. Они сохраняются в том же блоке расшифрованного сообщения и переходят в следующий блок расшифрованного сообщения через блок поточного ключа. Это связано с тем, что в этом режиме при создании текущего блока поточного ключа используется предыдущий блок зашифрованного сообщения (4.21);

3) при шифровании сообщений любым режимом желательно осуществлять контроль целостности зашифрованного сообщения.

Режим выработки имитовставки

В ГОСТе этот режим еще называют имитозащитой с таким определением: защита шифровальной связи от навязывания ложных данных.

В других терминах: режим имитовставки служит для контроля целостности зашифрованного сообщения при случайных и преднамеренных воздействиях. Контроль от преднамеренных воздействий является основным.

В связи с этим, имитовставка – это дополнительная контрольная информация, которая передается совместно с зашифрованным сообщением (c, I_m).

Формирование имитовставки I_m напоминает режим *CBC*, но отсутствует синхропосылка и число итераций при зашифровании в режиме простой замены равно 16.

Технология создания имитовставки следующая:

1) создается первый блок зашифрованного сообщения – $c_1=E_k(m_1)$;

2) затем в цикле выполняются операции

$$c_i=E_k(c_{i-1}(+)m_i),\ i=2,3,\ldots,I, \qquad (4.22)$$

где E_k – функция зашифрования по алгоритму ГОСТ в режиме простой замены, k – секретный ключ этого режима; I_m – это часть последнего зашифрованного блока (c_I).

В общем виде длина имитовставки равна l битов, $l<64$. Выбор длины имитовставки определяется требуемой достоверностью контроля целостности информации. При длине l битов, вероятность того, что искажения сообщения останутся не выявленными, равна 2^{-l}.

Учитывая (4.22), криптостойкость имитовставки определяется криптостойкостью секретный ключа (k).

Процедура контроля целостности следующая.

A – отправитель:

1) создает по описанной выше технологии имитовставку (I_m);

2) зашифровывает исходное сообщение по одному из трех рассмотренных режимов работы ГОСТа (получил c);

3) отправляет пару (c, I_m) по открытому каналу получателю.

B – получатель:

4) получил пару (c, I_m) по открытому каналу от отправителя;

5) расшифровал зашифрованное сообщение тем режимом, которым прошло зашифрование, получил исходное сообщение (m);

6) создал по описанной процедуре свою имитовставку (I^*_m);

7) сравнил свою имитовставку и полученную

$$I^*_m = I_m. \qquad (4.23)$$

Если равенство (4.23) выполнилось, то целостность зашифрованного сообщения подтверждается, иначе нет и необходимо принять решение, что делать дальше.

Чтобы работала описанная процедура необходимо заранее распределить секретный ключ для отправителя и получателя. В этом заключается недостаток режима выработки имитовставки.

Контроль целостности информации на основе имитовставки структурно совпадает с методами контроля целостности информации от случайных воздействий, рассмотренных в подп. 1.2.2. Основное отличие заключается в том, что при случайных воздействиях в методах контроля не используется секретная информация, а в режиме имитовставки используется секретный ключ. Поэтому режим имитовставки может противостоять не случайным воздействиям внешней среды, а преднамеренным воздействиям злоумышленника.

Список контрольных вопросов

1. Перечислите режимы алгоритма ГОСТ-89.
2. Дайте характеристику блочного режима шифрования..
3. Дайте характеристику поточных режимов шифрования.
4. Дайте характеристику режима контроля информации.
5. Как работает схема Фейстеля.
6. Что такое итерационные ключи в режиме *ECB*.
7. Что такое S-блоки и для чего они нужны.

Практическое занятие № 2
«Протоколы управления секретными ключами»

Данное практическое занятие рекомендуется провести после лабораторной работы № 9, посвященной обмену Диффи-Хеллмана.

Лекционное содержание занятия

Как уже отмечалось, симметричные криптосистемы имеют хорошее быстродействие, поэтому их основная функция – шифрование сообщений произвольной длины. Недостаток симметричных криптосистем – проблема распределения между пользователями секретных ключей.

Эффективность криптографической защиты информации во многом определяется надежностью протоколов управления ключами. Протоколы управления ключами включают в себя протоколы их генерации, распределения, хранения, смены и уничтожения.

Генерация секретных ключей для блочных шифров похожа на генерацию поточных ключей, которые рассмотрены в параграфе 4.2. Отличие заключается в том, что поточный ключ имеет произвольную длину, зависящую от длины сообщения. При блочном шифровании размер секретного ключа фиксирован.

Протоколы распределения ключей должны обеспечивать взаимную аутентификацию сторон, целостность сообщений и защиту от повторных запросов.

Протоколы распределения ключей – это действия, позволяющие пользователям получать секретные ключи. Используются различные типы протоколов: протоколы передачи уже сгенерированных ключей; протоколы совместной выработки общих ключей; протоколы предварительного распределения ключей.

Учитывая важность этого вопроса, в криптографии уделяется большое внимание управлению секретными ключами. Мы в этом параграфе рассмотрим только основные моменты этого вопроса.

При генерации секретных ключей можно укрупненно выделить два направления:

1) генерацию секретного ключа производит отправитель, далее он в зашифрованном виде отправляет его получателю. Эти протоколы называют протоколами «точка-точка», когда стороны сами создают и распределяют ключи;

2) генерацию секретных ключей производит третья доверенная сторона, которая затем распределяет их по парам пользователей.

Следует отметить, что если число пар пользователей равно p, то число необходимых секретных ключей равно

$$p_o = p \cdot (p - 1)/2. \qquad (4.32)$$

Как видно из (4.32), число необходимых секретных ключей находится в квадратической зависимости от числа пар. Например, число пар p=100, тогда число необходимых секретных ключей p_o=4950. Этот факт надо учитывать при выборе класса криптосистем.

Рассмотрим первое направление (протоколы «точка-точка»). В этом направление число возможных технологий зависит от числа используемых классов криптосистем. Сейчас мы рассмотрим протокол, когда используется только симметричная криптосистема.

В этом протоколе ключи делят на группы:

1) главные ключи или мастер-ключи (k_m), которые используются для генерирования других ключей и их для их шифрования при передаче по каналу связи. Генерирование и хранение этих ключей наиболее ответственный этап. Эти ключи могут быть «долгоживущими» (на несколько сеансов), они должны передаваться только по защищенному каналу;

2) сеансовые ключи (k_c), которые используются для зашифрования и расшифрования сообщений, обмениваемых между отправителем (A) и получателем (B). Они являются одноразовыми или действующими короткий период времени.

В качестве примера приведем стандарт *ISO* 8731, использующийся для генерации сеансового ключа k_c. Этот стандарт предлагает использовать протокол, основанный на мастер-ключе:

1. Вычисление промежуточного значения

$$J=E_{km}(DT),$$

где DT – дата и время отправки сообщения. На этом шаге эта информация зашифровывается симметричным блочным шифром с использованием мастер-ключа (k_m);

2. Генерация сеансового ключа

$$k_c=E_{km}(J(+)V),$$

где V – вектор инициализации стандарта криптографической системы (секретная информация);

3. Генерация нового значения вектора инициализации

$$V=E_{km}(k_c(+)V).$$

Вектор инициализации должен храниться в секрете. Криптостойкость сеансового ключа определяется криптостойкостью мастер-ключа и вектора инициализации.

Существуют и другие протоколы генерации секретных ключей.

Рассмотрим протокол, как имея мастер-ключ, можно решать проблему передачи сеансового ключа, который сгенерирован отправителем.

Создав сеансовый ключ, отправитель A:

а) зашифровывает сеансовым ключом исходное сообщение – $c=E_{kc}(m)$;

б) зашифровывает мастер-ключом сеансовый ключ – $c_k= E_{km}(k_c)$;

в) отправляет два файла получателю – (c, c_k).

Получатель B:

а) получает два файла от отправителя – (c, c_k);

б) расшифровывает мастер-ключом сеансовый ключ – $k_c = E^{-1}_{km}(c_k)$;

в) расшифровывает зашифрованное сообщение сеансовым ключом – $m = E^{-1}_{kc}(c)$.

Криптостойкость процесса шифрования сообщений обеспечивается высокой криптостойкостью сеансового ключа. Подчеркнем, что в этом протоколе по открытому каналу передается два зашифрованных файла – (c, c_k). Недостатком этого протокола является необходимость секретно распределять мастер-ключи.

Примечание – при шифровании мастер-ключа и сообщений могут использоваться различные режимы блочных шифров.

В параграфе 4.7 мы рассмотрим другие протоколы распределения секретных ключей для симметричной криптосистемы, которые дополнительно используют двухключевую криптосистему.

Список контрольных вопросов

1. Назначение протоколов управления секретными ключами.
2. Преимущество и недостаток симметричной криптосистемы относительно двухключевой криптосистемы.
3. Дайте характеристику протокола, основанного на симметричной криптосистеме.

Практическое занятие № 3
«Защита электронного документооборота»

Данное практическое занятие рекомендуется провести после 11-й лабораторной работы.

Лекционное содержание занятия

Введение

Документооборот – движение документов в организации с момента их создания или получения до завершения исполнения или отправления. Выделяют четыре основных стадии жизненного цикла документа: создание, обращение, хранение, уничтожение.

Электронный документооборот – технологии по работе с документами, представленными в электронном виде, с реализацией концепции «безбумажного делопроизводства».

Делопроизводство – отрасль деятельности, обеспечивающая документирование и организацию работы с официальными документами. В настоящее время термин делопроизводство связывают с неэлектронными документами, а документационное обеспечение управления – с электронными.

Документационное обеспечение управления – деятельность по созданию, обработке, хранению и использованию управленческих документов в целях принятия управленческих решений и контроля за их исполнением на основе использования соответствующих информационных технологий.

Система электронного документооборота – автоматизированная многопользовательская система, сопровождающая процесс управления работой иерархической организации с целью обеспечения выполнения ею своих функций.

В РФ функционирует система межведомственного электронного взаимодействия (СМЭВ), как цифровая среда предоставления услуг и исполнения государственных и муниципальных функций в электронной форме.

В настоящее время СМЭВ продолжает расширять свои возможности и вовлекать все большее количество участников взаимодействия. Это оказалось, как нельзя кстати, в том числе для коммерческих организаций, в частности банков, которые все больше стремятся перевести свои услуги в цифровую обработку.

Функционирует система межведомственного электронного документооборота (МЭДО) – федеральная информационная система, предназначенная для организации взаимодействия систем электронного документооборота участников межведомственного электронного документооборота.

С позиции защиты информации – электронный документооборот, это обработка документов произвольной длины, которые необходимо защищать.

Хотя между электронным документом и электронным сообщением имеются различия, в данном разделе это не учитывается.

Основными технологиями защиты, использующими криптографическое преобразование информации в электронном документообороте, являются: создание хэш-функции, создание и проверка электронной подписи.

С 1994 года по настоящее время в РФ создано три национальных стандарта для хэш-функции: ГОСТ Р 34.11-94, ГОСТ Р 34.11-2012, ГОСТ 34.11-2018. За это же время создано четыре национальных стандарта для электронной подписи: ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001, ГОСТ Р 34.11-2012, ГОСТ 34.10-2018.

В криптографии хэш-функцией называется криптографическое преобразование информации, переводящее строку битов произвольной длины в строку битов фиксированной длины – *N*. Эту строку битов называют хэш-кодом, хэш-образом или другим названием. Совпадение значений хэш-кодов для разных сообщений называют коллизией.

Хэш-функция $y_m=h(m)$ должна обладать тремя основными свойствами:

1) для данного значения y_m вычислительно трудоемко найти аргумент *m*, т.е. хэш-функция является односторонней;

2) для данного аргумента *m* вычислительно трудоемко найти другой аргумент *m*' такой, что $h(m) = h(m')$, т.е. хэш-функция является свободной от коллизий;

3) вычислительно трудоемко подобрать пару сообщений (m_1, m_2) у которых будут одинаковые хэш-коды: $h(m_1) = h(m_2)$.

Обзор зарубежных хэш-функций

В мировой практике используются различные хэш-функции. Рассмотрим некоторые из них.

Функция хэширования *SHA*-1 (США, 1994 г.), используется 160-битный хэш-код. В этом алгоритме блоки исходного текста имеют длину 512 битов (N_0=512). В последнем блоке хранится 64-х битовое представление длины сообщения (это представление завершает 512-ти битовый блок). Стартовый хэш-вектор состоит из пяти 32-битных констант (*N*=160 бита).

Позже этот стандарт усовершенствовали и назвали *SHA*-256 (хэш-код равен 256 битов). Криптостойкость хэш-функции увеличилась.

Затем была создана хэш-функция *SHA*-384 (хэш-код равен 384 битов, блок N_0=1024 бита). Последняя разработка – это хэш-функция *SHA*-512 (хэш-код равен 512 битов, блок N_0=1024 бита).

В США разработано семейство алгоритмов *MD* (дайджест), например, алгоритм *MD*5 с 128-битным хэш-кодом. По этой причине хэш-коды иногда называют дайджестами.

В Европе используется хэш-функция *RIPEMD*-160 (*N*=160 бита), созданная по инициативе европейского сообщества.

Хэш-функции *SHA*-256 и *RIPEMD*-160 используются в системе *Bitcoin* при определении адреса «кошелька» (рис. П3.1).

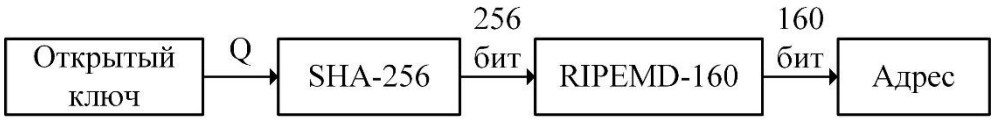

Рис. П3.1. Создание адреса «кошелька» в системе *Bitcoin*

Этот адрес получается из открытого ключа (Q) путем последовательного хэширования сначала функцией *SHA*-256, а затем функцией *RIPEMD*-160. В результате получается адрес «кошелька», содержащий 160 битов.

В мировой практике используются и другие хэш-функции.

Правовое обеспечение электронной подписи

С 2002 по 2013 г. в России действовал ФЗ «Об электронной цифровой подписи» от 10.01.2002 № 1-ФЗ. С 1 июля 2013 г. действует ФЗ «Об электронной подписи» № 63-ФЗ.

В соответствии с ФЗ № 63 электронная подпись (ЭП) – это «информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связанная с такой информацией и которая используется для определения лица, подписывающего информацию».

Из этого определения следует, что если m – электронное сообщение (подписываемая информация), то ЭП – это информация в электронной форме (z_m), присоединенная к сообщению m. Основной функцией ЭП является определение лица, подписавшего сообщение m (в дальнейшем он не может отказаться от подписи).

Хотя надо отметить, что в этом определении важное место занимает формулировка «или иным образом связанная с такой информацией». Это означает, что ЭП физически может отсутствовать, но функция по определению лица, подписавшего сообщение, может осуществиться.

Примечание – ЭП, это важное средство, позволяющее доверять электронному документообороту не в меньшей степени, чем бумажному. Прежде всего необходимо уметь определить лицо, отправившего электронный документ. Следующая важная функция – подтвердить целостность полученного документа.

Учитывая это, для ЭП, помимо национальных стандартов, созданы ФЗ. Например, для шифрования ФЗ отсутствуют.

Виды электронной подписи

Основным отличием действующего ФЗ от предыдущего является введение нескольких видов ЭП. В соответствии с ФЗ № 63 установлены следующие виды ЭП:

- простая электронная подпись;

- усиленная электронная подпись.

В свою очередь, усиленная электронная подпись разделяется на две:

- усиленная неквалифицированная электронная подпись (далее – неквалифицированная ЭП);

- усиленная квалифицированная электронная подпись (далее – квалифицированная ЭП).

Таким образом, в действующем ФЗ № 63 выделяют три вида электронной подписи (рис. П3.2):

- простую ЭП;

- неквалифицированную ЭП;

-квалифицированную ЭП.

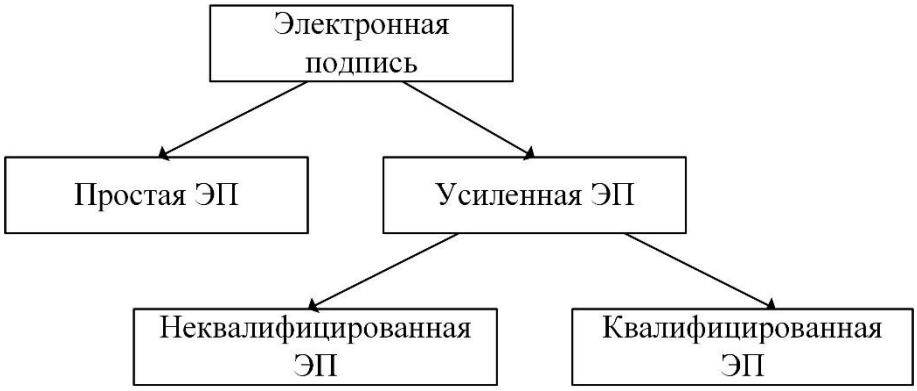

Рис. П3.2. Виды электронной подписи

Простая ЭП – это подпись, которая посредством использования кодов, паролей или иных средств подтверждает факт формирования электронной подписи определенным лицом.

Неквалифицированная ЭП – это подпись, которая получена в результате криптографического преобразования информации с использованием ключа электронной подписи. Она позволяет определить лицо, подписавшее электронный документ, а также обнаружить факт внесения изменений в электронный документ после момента его подписания (контроль целостности информации). Создается эта подпись с использованием средств электронной подписи (двухключевая криптосистема, использующая хэш-функцию).

Квалифицированная ЭП – это подпись, которая соответствует всем признакам неквалифицированной ЭП и следующим дополнительным признакам: ключ проверки электронной подписи указан в квалифицированном сертификате; для создания и проверки ЭП используются средства ЭП, получившие подтверждение соответствия требованиям, установленным в ФЗ № 63.

Таким образом, усиленная ЭП использует криптографическую систему с двумя ключами:

- ключ электронной подписи – уникальная последовательность символов, предназначенная для создания электронной подписи (секретный ключ;

- ключ проверки электронной подписи – уникальная последовательность символов, однозначно связанная с ключом электронной подписи и предназначенная для проверки подлинности электронной подписи (далее – проверка электронной подписи) (открытый ключ).

Важное отличие усиленной ЭП от простой связано с появлением второй функции – контроль целостности подписанной информации. Эта дополнительная функция появилась благодаря использованию криптографической системы с двумя ключами.

Первая функция ЭП – определение лица (его аутентификация), подписавшего сообщение m, также осуществляется усиленной подписью.

Можно дать такое определение усиленной ЭП – это информация в электронной форме (z_m), которая присоединена к другой информации в электронной форме (подписываемой информации, m), которая используется для аутентификации лица, подписавшего информацию, и для контроля целостности подписанной информации (m, z_m).

Отличие квалифицированной ЭП от неквалифицированной заключается в ее юридической силе. В технологии ЭП ее средства должны быть сертифицированы, а ключи и сертификаты должны выдаваться аккредитованным органом (удостоверяющим центром).

Удостоверяющий центр осуществляет создание и выдачу сертификата ключа проверки ЭП на основании соглашения между удостоверяющим центром и заявителем. Сертификат ключа проверки электронной подписи должен содержать следующую информацию:

1) дата начала и окончания срока его действия;

2) фамилия, имя и отчество (если имеется) – для физических лиц, наименование и место нахождения – для юридических лиц или иная информация, позволяющая идентифицировать владельца сертификата ключа проверки электронной подписи;

3) ключ проверки электронной подписи;

4) наименование используемого средства ЭП и (или) стандарты, требованиям которых соответствуют ключ ЭП и ключ проверки ЭП;

5) наименование удостоверяющего центра, который выдал сертификат ключа проверки ЭП;

6) иная информация, предусмотренная ч. 2 ст. 17 действующего федерального закона, – для квалифицированного сертификата.

Инфраструктура управления открытыми ключами

При использовании асимметричных криптосистем необходимо распределить открытые ключи по другим пользователям. Чтобы исключить атаку «человек по середине», необходимо это сделать, обеспечивая защищенность открытых ключей.

Защищенность открытых ключей для получателя заключается в следующем:

- необходимо определить владельца этого ключа;

- необходимо убедиться, что получен тот ключ, который отправлен.

Как раз эти функции выполняет ЭП.

В настоящее время одним из эффективных способов преодоления атаки «человек по середине» является технология *PKI* (инфраструктура открытых ключей).

Технология *PKI* основывается на цифровых сертификатах, в которых хранятся открытые ключи, аналогично российской технологии, использующей для выдачи сертификатов удостоверяющие центры. Отметим, что в настоящее время сертификаты выдаются и на секретные ключи, но в этом случае они не содержат секретные ключи, а содержат информацию о пользователе, о сроке действия ключа и другую служебную информацию. Секретные ключи хранятся, как правило, в специальных защищенных устройствах, например, *USB*-брелках.

В технологии *PKI* речь идет о сертификатах для открытых ключей.

В основу формирования сертификатов открытых ключей положены принципы строгой аутентификации, рекомендованные международным стандартом *X*.509.

Чтобы пользователь мог доверять процессу аутентификации, он должен извлекать открытый ключ другого пользователя из надежного источника, которому он доверяет.

Таким источником согласно стандарту *X*.509 является центр сертификации – доверенная сторона, обеспечивающая аутентификацию открытых ключей, содержащихся в сертификатах. Этот центр имеет свою пару ключей: секретный ключ для подписи сертификатов и опубликованный открытый ключ, который используется пользователями для проверки подлинности сертификатов с открытыми ключами других пользователей.

Сертификат открытого ключа, который отправляет центр сертификации – это два электронных сообщения: 1) сам сертификат, где помимо открытых ключей субъектов, хранится служебная информация; 2) ЭП, которую сформировал центр для защиты сертификата.

Служебная информация напоминает информацию, которая хранится в сертификате открытого ключа российского удостоверяющего центра. Подчеркнем, что одной из компонент служебной информации является информация о владельце сертификата (кто заказал этот сертификат совместно с секретным ключом).

Сертификаты открытого ключа обладают двумя важными свойствами:

1) каждый зарегистрированный в центре сертификации пользователь может извлечь открытый ключ сертификата;

2) ни одна сторона, помимо центра сертификации, не может изменить сертификат (сертификаты нельзя подделать).

Помимо центра сертификации в технологии *PKI* используется центр регистрации, как организационная компонента, назначение которой – регистрация пользователей.

Пользователь – владелец какого-либо сертификата (подлежит регистрации) или любой пользователь, запрашивающий сертификат, хранящийся в каталоге сертификатов. Каталог сертификатов – общедоступное хранилище сертификатов пользователей.

Таким образом, в технологии *PKI* при обслуживании пользователей используются центры сертификации, регистрации и каталог сертификатов.

Технология *PKI* выглядит следующим образом:

1) центр сертификации создает собственную пару ключей и формирует свой сертификат на открытый ключ;

2) пользователи регистрируются в центре регистрации;

3) зарегистрированные пользователя формируют запросы на ключи электронной подписи и сертификаты для ключей проверки подписи;

4) центр сертификации выдает сертификаты пользователям, обеспечивая их защиту средствами ЭП;

5) ключи электронной подписи передаются пользователям по защищенным каналам, включая применение программно-аппаратных решений, рассмотренных далее;

6) пользователи, получая сертификаты по каналам связи, проверяют подлинность ЭП, используя открытый ключ центра сертификации;

7) центр сертификации следит за реестром сертификатов в каталоге, учитывая время их действия.

В России технологию *PKI* поддерживают сертифицированные коммерческие центры различных компаний, например, компании «КРИПТО-ПРО», группа компаний «Информзащита» и др. Эти же компании, как правило, через свои учебные центры проводят обучение пользователей технологии электронных подписей и технологии *PKI*.

В настоящий момент различные компании создают и распространяют индивидуальные средства безопасности в виде программно-аппаратных решений в области аутентификации, защиты информации и электронной подписи. Примером этих решений в виде *USB*-брелков являются «РуТокены», «*e*-Токены» и их модификации.

Список контрольных вопросов

1. Приведите перечень созданных в РФ ГОСТов на хэш-функцию и электронную подпись.
2. Приведите примеры функций хэширования в США, укажите размеры хэш-кодов.
3. Какие хэш-функции и как используются в системе Bitcoin.
4. Приведите размеры хэш-кода и ЭП в ГОСТе 2018 года.
5. Виды ЭП в ФЗ № 63. В чем их отличие.
6. Дайте характеристику технологии PKI.

Список рекомендуемой литературы

Основная литература

1. Краковский Ю.М. Информационная безопасность и защита информации. Иркутск: Изд-во ИрГУПС, 2016. 224 с.
2. Краковский, Ю. М. Защита информации : учебное пособие / Ю. М. Краковский. — Ростов н/Д : Феникс, 2017. — 347 с.
3. Ерохин, В. В. Безопасность информационных систем : учебное пособие / В. В. Ерохин, Д. А. Погонышева, И. Г. Степченко. — 2-е изд. — Москва : ФЛИНТА, 2015. — 182 с.

Дополнительная литература

4. Мельников, Д. А. Информационная безопасность открытых систем : учебник / Д. А. Мельников. — 2-е изд. — Москва : ФЛИНТА, 2014. — 448 с.
5. Бузов, Г. А. Защита информации ограниченного доступа от утечки по техническим каналам : справочное пособие / Г. А. Бузов. — Москва : Горячая линия-Телеком, 2018. — 586 с.
6. Голиков, А. М. Защита информации от утечки по техническим каналам : учебное пособие / А. М. Голиков. — Москва : ТУСУР, 2015. — 256 с.
7. Долозов, Н. Л. Программные средства защиты информации : учебное пособие / Н. Л. Долозов, Т. А. Гультяева. — Новосибирск : НГТУ, 2016. — 63 с.
8. Программно-аппаратные средства защиты информации : учебное пособие / Л. Х. Мифтахова, А. Р. Касимова, В. Н. Красильников [и др.]. — Санкт-Петербург : Интермедия, 2018. — 408 с.

Краковский Юрий Мечеславович

МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Методическое пособие