Федеральное агентство по рыболовству
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Астраханский государственный технический университет»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ

КАФЕДРА ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

«ЗАЩИТА ИНФОРМАЦИИ»

Методические указания для выполнения лабораторных работ
для обучающихся по направлению подготовки
09.03.01 «Информатика и вычислительная техника,
профиль подготовки
«Автоматизированные системы обработки информации и управления»
и направлению подготовки
*09.03.04 «Программная инженерия», п*рофиль подготовки
«Разработка программно-информационных систем»

Астрахань- 2018

Составители: Белов С.В., к.т.н., доцент кафедры «Информационная безопасность»,
Давидюк Н.В., к.т.н., доцент кафедры «Информационная безопасность»

Рецензент: Попов Г.А., д.т.н., проф., зав. кафедрой «Информационная безопасность»

Утверждены на заседании кафедры «Информационная безопасность» «14» июня 2018 г., протокол №14.

СОДЕРЖАНИЕ

Лабораторная работа №1 «Расчет вероятности проникновения злоумышленника на объект хранения информации»

Теоретические сведения

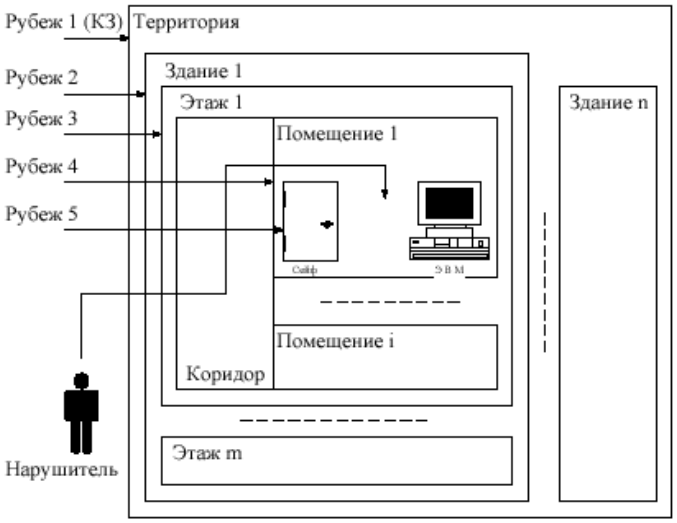
Физическая защита – относительно статическая область, состоящая из систем разграничения доступа, систем сигнализации и, возможно, оборудования для наблюдения, позволяющих идентифицировать и предотвратить физическое вторжение в защищаемую область.

В составе любого объекта защиты можно выделить некоторые ограниченные замкнутые пространства или совокупность тесно связанных между собой по определенному признаку (единый пропускной режим, место расположения, технология рабочих процессов, перечень сотрудников и т. д.) объектов в составе общего объекта - так называемые *зоны безопасности*. Зоны безопасности могут располагаться одна внутри другой, и по отношению к ним должны обеспечиваться следующие требования:

1. **эшелонирование средств защиты**, что предполагает расположение более важных зон безопасности внутри менее важных;
2. **равнопрочность**, т. е. одинаковая степень противостояния одной и той же угрозе и отсутствие незащищенных мест для всех участков зоны.

Обычно выделяют пять типов зон (рис 1.):

1. Зона окружающей среды(не охраняемая зона). Это внешняя среда, окружающая объект. Она является началом пути движения злоумышленника.
2. Территория объекта(мало охраняемая зона). Это зона от забора до зданий.
3. Зона зданий.
4. Зона комнат.
5. Средство хранения информации.



Каждый тип зоны имеет свои особенности при использовании методов обеспечения безопасности.

В каждой зоне можно рассчитать сложность (вероятность) прохождения пути между двумя точками следующим образом:

$$P_{AB} = \frac{k_ç}{l_{AB}}$$

где

l_{AB} - расстояние между точками АВ

$k_ç$ - коэффициент, определяющий степень защищенности данной зоны.

Соединив точки, получается граф, вершинами которых являются точки соприкосновения зон (могут быть окна, двери и т.д.), а дугами сами зон. Граф взвешенный: вершины и дуги имеют веса. Так появляется задача поиска самого надежного пути R_{AB} на графе следующим образом:

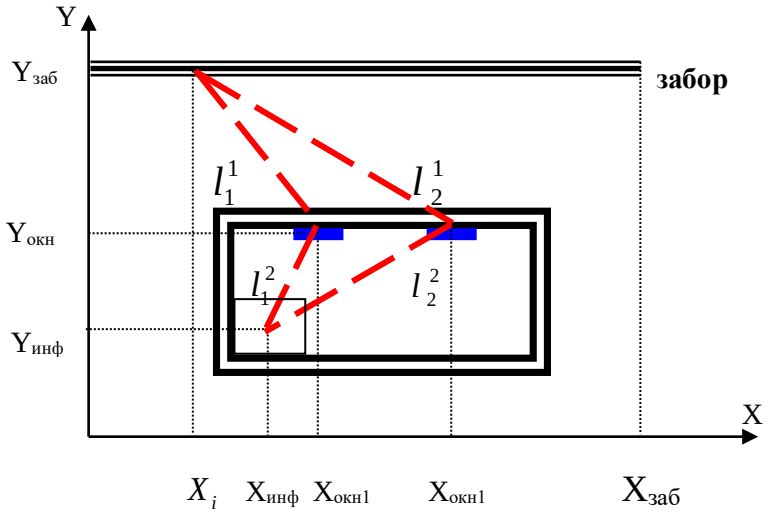
$$P_A\left[\prod_{i:P_{i,i+1}\in R_{AB}} P_{i,i+1}P_{i+1}^{i}\right]P_B \to \min$$

где

$P_{i,j}$ - вес дуги между вершинами i и j

P_i - вес i-ой вершины

Выполнение работы



Исходные данные:

План расположения объекта: $Y_{заб}<50$, $X_{заб}<50$, $Y_{ок}$, $X_{оку1}$, $X_{окн}$, $Y_{ин}$, $X_{ин}$

Вероятность проникновения через окна: $P_{ок1}$, $P_{ок2}$

Забор $X_{заб}$ разбивается на n=100 участков.

Для каждого I-го участка (0<i<100) рассчитывается вероятность проникновения через первое окошко P_{i1} и через второе окошко P_{i2}, следующим образом:

$$P_{i1} = \frac{k_1}{l_1^1} P_{ок1} \frac{k_2}{l_1^2}$$

$$P_{i2} = \frac{k_1}{l_2^1} P_{ок2} \frac{k_2}{l_2^2}$$

где k_1= 2 и k_2=0.5, а расстояния $l_1^1, l_2^1, l_1^2, l_2^2$ рассчитать самостоятельно из графика.

В полученном массиве P Nx2 найти минимум, что и является результатом проникновения.

Лабораторная работа №2 «Преобразование аналоговых телефонных сообщений (скремблеры)»

1. Цель работы

Ознакомиться с основными понятиями преобразования аналоговых сообщений. Реализация простого скремблера.

2. Основные сведения

Наиболее простым и распространенным способом криптографического преобразования аналоговых сообщений является разбиение сообщений X(t) на части и выдача этих частей в определенном порядке в канал связи.

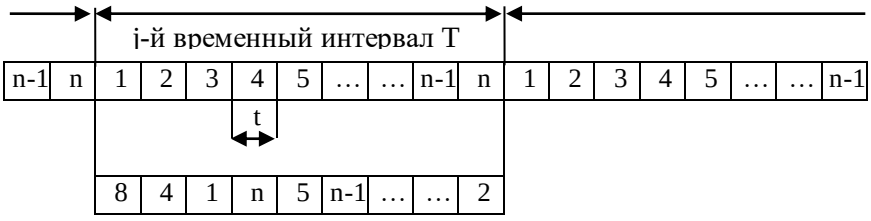


Рис. 1. Временные перестановки частей сообщения X(t)

Этот способ заключается в следующем. Длительность сообщения X(t) (см. рис.1) делится на определенные, равные по длительности временные интервалы T. Каждый такой временной интервал дополнительно делится на более мелкие временные интервалы длительностью t . При этом для величины n=T/t , как правило, выполняется условие n = m ...10m , где m - некоторое целое число, m <10. Части сообщения X(t) на интервалах времени t записываются в запоминающее устройство, “перемешиваются” между собой в соответствие с правилом, определяемым ключом криптографического преобразования k, и в виде сигнала Y(t) выдаются в канал связи. На приемной стороне канала связи, где правило перемешивания известно, т.к. имеется точно такой же ключ криптографического преобразования k, осуществляется “сборка” из сообщения Y(t) открытого сообщения X(t).

К преимуществам этого способа криптографического преобразования относится его сравнительная простота и возможность передачи зашифрованного телефонного сообщения по стандартным телефонным каналам. Однако этот способ позволяет обеспечить лишь временную стойкость. Это обусловлено следующим. Поскольку открытое телефонное сообщение X(t) является непрерывным, то у злоумышленника после записи сообщения Y(t) и выделения интервалов длительностью t (последнее достаточно легко сделать, т. к. в канале связи присутствует синхронизирующий сигнал) появляется принципиальная возможность дешифрования сообщения Y(t) даже без знания используемого ключа k. С этой целью необходимо осуществить выбор интервалов таким образом, чтобы обеспечивалась непрерывность получаемого сообщения на стыках этих интервалов. Очевидно, что при тщательной и кропотливой работе с использованием специальной техники можно достаточно быстро обеспечить такую непрерывность, выделив тем самым открытое сообщение X(t).

Поэтому такой способ криптографического преобразования открытых телефонных сообщений целесообразно применять только в тех случаях, когда информация не представляет особой ценности или когда ее ценность теряется через относительно небольшой промежуток времени.

Более высокую защиту от несанкционированного доступа можно обеспечить, если идею рассмотренного способа распространить на частотный спектр сообщения X(t). В этом случае полоса пропускания телефонного канала F делится с помощью системы полосовых фильтров на n частотных полос шириной D f, которые перемешиваются в соответствии с некоторым правилом, определяемым ключом криптографического преобразования k. Перемешивание частотных полос осуществляется со скоростью V циклов в секунду, т.е. одна перестановка полос длится 1/V с, после чего она заменяется следующей.

Для повышения защиты от несанкционированного доступа после перемешивания частотных полос может осуществляться инверсия частотного спектра сообщения Y(t).

Рис.2 иллюстрирует рассмотренный способ криптографического преобразования. В верхней части рис.2 приведен частотный спектр сообщения X(t), а в нижней - спектр сообщения Y(t) на одном из циклов перемешивания при n = 5.

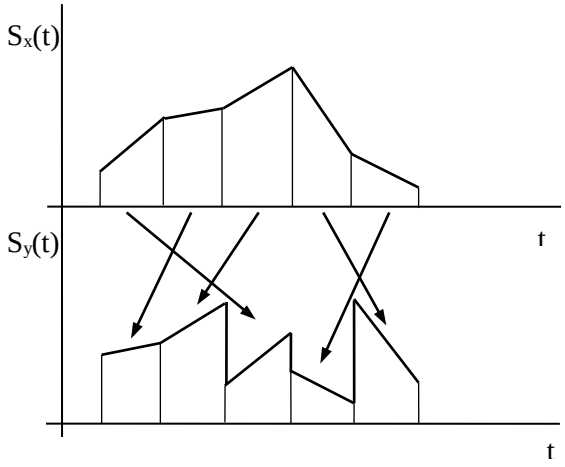


Рис. 2. Частотные спектры сообщений X(t) и Y(t)

Рассмотренный способ позволяет обеспечить более высокую защиту телефонных сообщений от несанкционированного доступа по сравнению с предыдущим способом. Для восстановления открытого сообщения X(t) в этом случае злоумышленнику необходимо иметь дополнительные данные по относительным частотам появления звуков и их сочетаний в разговорной речи, частотным спектрам звонких и глухих звуков, а также формантной структуре звуков. В табл.1 приведены данные по относительным частотам появления некоторых звуков и границам формантных областей звуков русской речи, которые могут быть использованы злоумышленником при восстановлении перехваченных телефонных сообщений.

Таблица 1.Данные по относительным частотам появления некоторых звуков и границам формантных областей

Звук	Относительная частота появления, Гц	1-ая формантная область,	2-ая формантная область, Гц
Гласный			
а	0,079	1100 - 1400	-
и	0,089	2800 - 4200	-
о	0,11	400 - 800	-
у	0,026	200 - 600	-
ы	0,022	200 - 600	1500 - 2300
э	0,002	600 - 1000	1600 - 2500
Согласный			
з	0,016	0 - 600	4200 - 8600
ж	0,008	200 - 600	1350 - 6300
л	0,04	200 - 500	700 - 1100
м	0,031	0 - 400	1600 - 1850
н	0,069	0 - 400	1500 - 3400
р	0,05	200 - 1500	-
с	0,054	4200 - 8600	-
ф	0,001	7000 - 12000	-
х	0,012	400 - 1200	-
ш	0,008	1200 - 6300	-

Очевидно, что наиболее высокую защиту телефонных сообщений от несанкционированного доступа представляется возможным обеспечить путем объединения рассмотренных способов. При этом временные перестановки будут разрушать смысловой строй, а частотные перемешивать гласные звуки.

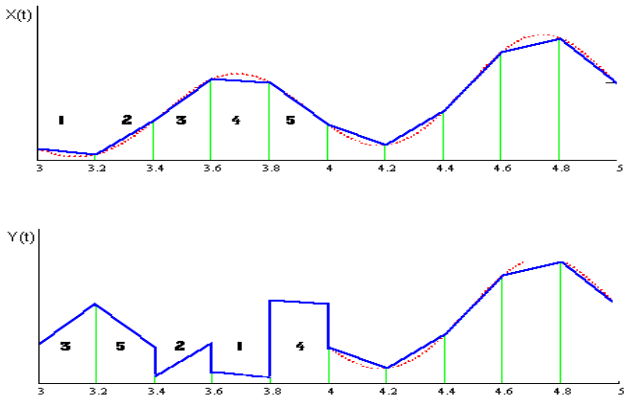
Устройства, реализующие рассмотренные способы, называются скремблерами.

В этой связи представляет определенный интерес серия скремблеров, в качестве базового для которой был использован скремблер SCR - M1.2. Эти скремблеры реализуют рассмотренные способы криптографического преобразования аналоговых телефонных сообщений и довольно широко используются в различных государственных и коммерческих структурах. В табл.2 приведены основные характеристики некоторых скремблеров этой серии.

2. Выполнение работы

Выбирается функция X (t), описывающая значение входного сигнала в момент t_1. Можно положить X(t) = A sin (α t) + B cos (β t) + t C cos (cos (γ t)), где A, B, C, α, β, γ – константы, (α >0, β > 0, γ > 0).

Длительность сообщения X(t) (см. рис) делится на определенные, равные по длительности временные интервалы T = 1-2. Каждый такой временной интервал дополнительно делится на более мелкие временные интервалы длительностью t . При этом для величины t=T/n , как правило, выполняется условие n = m ...10m , где m - некоторое целое число, m <10. Части сообщения X(t) на интервалах времени t записываются в запоминающее устройство, “перемешиваются” между собой в соответствие с правилом, определяемым ключом криптографического преобразования k, и в виде сигнала Y(t) выдаются в канал связи. Выполняется обратное преобразование.



Лабораторная работа №3 «Модели дискреционного управления доступом Харрисона-Руззо-Ульмана»

1. Цель

Ознакомиться с основными понятиями моделей доступа. Реализация матричной модели доступа.

2. Теоретические сведения

Одна из первых моделей безопасности была модель дискреционного доступа, модель АДЕПТ-50. В модели представлено четыре типа объектов, относящихся к безопасности: пользователи(u), задания(j), терминалы(t) и файлы(f), причем каждый объект описывается четырехмерным кортежем (A, C, F, M), включающим параметры безопасности:

Компетенция A - скаляр - элементы из набора иерархически упорядоченных уровней безопасности, таких как: НЕСЕКРЕТНО, КОНФИДЕНЦИАЛЬНО, СЕКРЕТНО, СОВЕРШЕННО СЕКРЕТНО.

Категория C - дискретный набор рубрик. Категории не зависят от уровня безопасности. Пример набора рубрик: (ОГРАНИЧЕНО, ТАЙНО, ТОЛЬКО ДЛЯ ПРОСМОТРА, ЯДЕРНЫЙ, ПОЛИТИЧЕСКИЙ).

Полномочия F - группа пользователей, имеющих право на доступ к определенному объекту.

Режим M - набор видов доступа, разрешенных к определенному объекту или осуществляемых объектом. Пример: ЧИТАТЬ ДАННЫЕ, ПРИСОЕДИНЯТЬ ДАННЫЕ, ИСПОЛНИТЬ ПРОГРАММУ.

Если U={u} обозначает набор всех пользователей, известных системе, а F(i) - набор всех пользователей, имеющих право использовать объект i, то для модели формулируются следующие правила:

Пользователь u получает доступ к системе $\Leftrightarrow$ $u \in U$.

Пользователь u получает доступ к терминалу t $\Leftrightarrow$ $u \in F(t)$ (т.е. в том и только в том случае, когда пользователь u имеет право использовать терминал t).

Пользователь u получает доступ к файлу j $\Leftrightarrow$ $A(j) \geq A(f)$, $C(j) \supseteq C(f)$, $M(j) \supseteq M(f)$ и $u \in F(f)$, т.е. тогда и только тогда, когда:

привилегии выполняемого задания шире привилегий файла или равны им;

пользователь является членом F(f).

Задавая параметры безопасности A, C, F, M можно сформировать матрицу определения параметров безопасности (Табл. 2.1.).

Таблица 2.1.

Матрица определения параметров безопасности модели АДЕПТ-50.

Объект	A	C	F	M
Пользователь u	Const	Const	{u}	Const
Терминал t	Const	Const	{u(t,i)}	Const
Задание j	min(A(u),A(t))	$C(u) \cap C(t)$	{u(j,i)}	$M(u) \cap M(t)$
Существ. файл f(i)	Const	Const	{u(f,i)}	Const
Нов.файл f=g(f1,f2)	max(A(f1),A(f2))	$C(f1) \cup C(f2)$	{u(f,j)}	$M(f1) \cup M(f2)$

f1, f2 - старые файлы; новый файл f является некоторой их функцией.

Четырехмерный кортеж безопасности, полученный на основе прав задания, а не прав пользователя, используется в модели для управления доступом. Данный подход обеспечивает однородный контроль права на доступ над неоднородным множеством программ и данных, файлов, пользователей и терминалов. Например, наивысшим полномочием доступа к файлу для пользователя "СОВ. СЕКРЕТНО", выполняющего задание с "КОНФИДЕНЦИАЛЬНОГО" терминала будет "КОНФИДЕНЦИАЛЬНО".

Теперь рассмотрим модель, называемую пятимерным пространством безопасности Хартстона[3]. В данной модели используется пятимерное пространство безопасности для моделирования процессов, установления полномочий и организации доступа на их основании. Модель имеет пять основных наборов:

A - установленных полномочий;

U - пользователей;

E - операций;

R - ресурсов;

S - состояний;

Область безопасности будет выглядеть как декартово произведение: $A \times U \times E \times R \times S$. Доступ рассматривается как ряд запросов, осуществляемых пользователями u для осуществления операции e над ресурсами R, в то время, когда система находится в состоянии s. Например, запрос на доступ представляется четырехмерным кортежем $q = (u, e, R, s)$, $u \in U$, $e \in E$, $s \in S$, $r \subseteq R$. Величины u и s задаются системой в фиксированном виде. Таким образом, запрос на доступ - подпространство четырехмерной проекции пространства безопасности. Запросы получают право на доступ в том случае, когда они полностью заключены в соответствующие подпространства.

Процесс организации доступа можно описать алгоритмически следующим образом. Для запроса q, где q(u, e, R, s), набора U' вполне определенных групп пользователей, набора R' вполне определенных единиц ресурсов и набора P правильных (установленных) полномочий, процесс организации доступа будет состоять из следующих процедур.

1. Вызвать все вспомогательные программы, необходимые для предварительного принятия решений.

2. Определить из U те группы пользователей, к которым принадлежит u. Затем выбрать из P спецификации полномочий, которым соответствуют выделенные группы пользователей. Этот набор полномочий F(u) определяет привилегию пользователя u.

3. Определить из P набор F(e) полномочий, которые устанавливают e как основную операцию. Этот набор называется привилегией операции e.

4. Определить из P набор F(R)(привилегию единичного ресурса R) - полномочия, которые определяют поднабор ресурсов из R', имеющего общие элементы с запрашиваемой единицей ресурса R.

Полномочия, которые являются общими для трех привилегий в процедурах 2, 3, 4 образуют D(q), так называемый домен полномочий для запроса

q: D(q)=F(u)∩F(e)F(R)

5. Удостовериться, что запрашиваемый ресурс R полностью включается в D(q), т.е. каждый элемент из R должен содержаться в некоторой единице ресурса, которая определена в домене полномочий D(q).

6. Осуществить разбиение набора D(q) на эквивалентные классы так, чтобы два полномочия попадали в эквивалентный класс тогда и только тогда, когда они специфицируют одну единицу ресурса. Для каждого такого класса логическая операция ИЛИ (или И) выполняется с условиями доступа элементов каждого класса.

Новый набор полномочий - один на каждую единицу ресурса, указанную в D(q), есть F(u, q) - фактическая привилегия пользователя u по отношению к запросу q.

7. Вычислить EAC - условие фактического доступа, соответствующего запросу q, осуществляя логическое И (или ИЛИ) над условиями доступа членов F(u, q).Это И(или ИЛИ) выполняется над всеми единицами ресурсов, которые перекрывают единицу запрошенного ресурса.

8. Оценить EAC и принять решение о доступе:

разрешить доступ к R, если R перекрывается;

отказать в доступе в противном случае.

9. Произвести запись необходимых событий.

10. Вызвать все программы, необходимые для организации доступа после "принятия решения".

11. Выполнить все вспомогательные программы, вытекающие для каждого случая из условия 8.

12. Если решение о доступе было положительным - завершить физическую обработку.

Автор модели, Хартстон, отмечает, что приведенная последовательность шагов не всегда необходима в полном объеме. Например, в большинстве реализаций шаги 2 и 6 осуществляются во время регистрации пользователя в системе.

3.1 Модель дискреционного доступа, продемонстрированная Харрисоном, Руззо и Ульманом.

Рассмотрим типичную модель системы защиты, состоящую из следующих частей.

1. Конечный набор общих прав $A=\{a_1, ..., a_n\}$.

2. Конечный набор исходных субъектов S_0 и конечный набор исходных объектов O_0.

Конечный набор команд C формы $\alpha(X_1, ..., X_n)$, где α - имя; $X_1, ..., X_n$ - формальные параметры, указывающие на объекты.

Элементами матрицы доступа являются права доступа, взятые из набора общих прав. Состояния системы изменяются при изменении элементов матрицы доступа M. Запросы к системе можно выразить в форме:

if a_1 in $M[s_1, o_1]$ and

a_2 in $M[s_2, o_2]$ and

...

a_m in $M[s_m, o_m]$

then

op_1,

op_2,

...

op_n.

Причем $\forall a_i \in A$ и операция op является одной из следующих примитивных операций:

enter a into (s, 0);
delete a from (s, 0);
create subject s;
create object o;
destroy subject o;
destroy object o.

Семантика данных операций очевидна. Для системы с начальной конфигурацией Q_0 и права a, можно сказать, что система безопасна для a, если не существует последовательности запросов к системе в состоянии Q_0 таких, что в результате них право a будет записано в ячейку, не содержащую ее. Существует две теоремы о безопасности данного типа систем. Первая относится к безопасности моно-операционных систем. Под моно-операционной системой понимается система, в которой каждый запрос имеет только одну операцию.

Теорема. Существует алгоритм для определения, является или нет моно-операционная система безопасной для данного права a.

Вторая теорема указывает на то, что проблема безопасности для системы с запросами общего вида является неразрешимой.

Теорема. Проблема определения безопасности для данного права a в системе с запросами общего вида является неразрешимой.

Доказательство данных теорем приведено в приложении 1. Харрисон, Руззо и Ульман показали, что безопасными являются монотонные системы (системы, не содержащие операции destroy и delete), системы не содержащие операций create и моно-условные системы (системы, запрос к которым содержит только одно условие).

К достоинствам моделей дискреционного доступа можно отнести хорошую гранулированность защиты и относительно простую реализацию. В качестве примера реализаций данного типа моделей можно привести так называемую матрицу доступа, строки которой соответствуют субъектам системы, а столбцы - объектам; элементы матрицы характеризуют права доступа. Проблемы, возникающие в системах, синтезированных на их основании показаны в следующем параграфе.

3. Выполнение работы

Исходные данные:

N – количество пользователей

M – количество объектов

Матрица Users[N] – матрица пользователей

Матрица Object[M] – матрица значений объектов типа String

Матрица P[N, M+1] – матрица прав доступа,

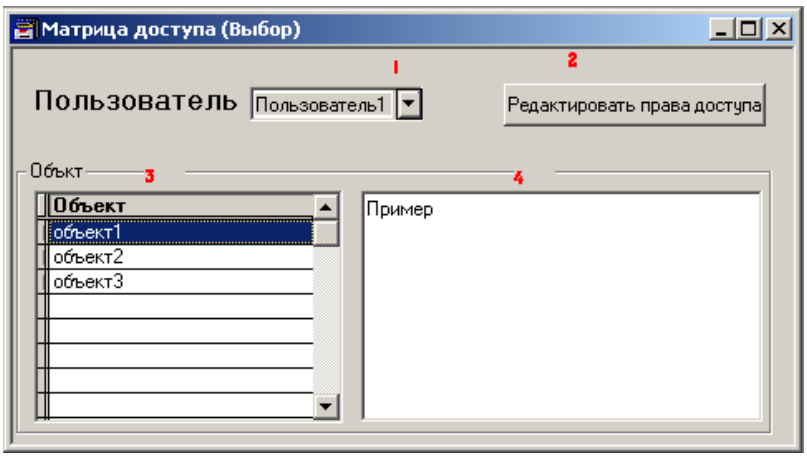
Где

P[I, 0] – административные права по редактированию матрицы доступа для I-го пользователя (1, 0)

P[I, J] (1<=J<=m) – права I-го пользователя к J-му объекту (0 – нет прав, 1 – чтение, 2 – модификация)

Выполнение работы:

Разработайте следующую форму.



В списке 1 выбирается пользователь **I** и согласно матрицы доступа **P**:
Если P[I, 0] = 0, то кнопка 2 недоступна.
В списке 3 выбирается объект J,
если P[I, J] = 0, то поле 4 недоступно, и значение объекта J не выводиться.
если P[I, J] = 1, то поле 4 недоступно, и значение объекта J выводиться.
если P[I, J] = 2, то поле 4 доступно, и значение объекта J выводиться. При редактировании поля 4, необходимо записывать результат в Object[J].

При выборе кнопке 2 происходит вызов формы редактирования матрицы доступа в которой выполняются следующее операции:
Добавление удаление пользователей.
Добавление удаление объектов.
Назначение и изменение прав доступа.

Лабораторная работа №4 «Модель доступа Белла-ЛаПадула»

Цель
Ознакомиться с основными понятиями многоуровневой модели доступа. Реализация модели Белла-ЛаПаддула.

Основные сведения

Расширением матричной модели доступа является многоуровневая модель доступа. Объекты в многоуровневой модели имеют различные уровни доступа (например, уровни секретности), а субъекты - степени допуска. Разрешение допуска субъекта к объекту является функцией от степени допуска конкретного субъекта и уровня допуска конкретного объекта.

Многоуровневая модель доступа создана на основе теории алгебраических решеток. Данные могут передаваться между субъектами, если выполняются следующие правила (здесь буквами **a**, **b** и **c** будем обозначать идентификаторы субъектов, а буквами **x**, **y** и **z**, соответственно, их уровни доступа).

1. Данные могут передаваться субъектом самому себе:
   $x <= x$
2. Данные могут передаваться от субъекта a к субъекту c, если они могут передаваться от субъекта a к субъекту b и от субъекта b к субъекту c:
   если $x <= y$ и $y <= z$, то $x <= z$,
3. Если $x <= y$ и $y <= x$, то $x = y$.

Отметим, что рассмотренные правила представляют, соответственно, свойства рефлективности, транзитивности и антисемметричности.

Примером использования многоуровневой модели доступа является система контроля доступа, принятая в военном ведомстве США. Уровнями доступа выступают уровни

секретности: НЕСЕКРЕТНО, КОНФИДЕНЦИАЛЬНО, СЕКРЕТНО, СОВЕРШЕННО СЕКРЕТНО.

Внутри отдельных уровней секретности для выделения разделов данных, требующих специального разрешения на доступ к ним, определены категории: АТОМНЫЙ, НАТО и ДРУГИЕ. Для получения доступа к данным определенной категории субъект должен иметь не только доступ к данным соответствующего уровня (по секретности), но и разрешение на доступ по категории. Например, субъект, имеющий доступ к данным с уровнем СОВЕРШЕННО СЕКРЕТНО и категории НАТО, не может получить доступ к данным с категориями АТОМНЫЙ и ДРУГИЕ уровня СОВЕРШЕННО СЕКРЕТНО.

2.1. Модель Белла-Лападулы

Модель Белла и Лападулы состоит из следующих элементов:

- множества субъектов S;
- множества объектов O;
- множества уровней защиты L;
- множества прав доступа G;
- списка текущего доступа b;
- списка запросов Z.

Каждому субъекту s пр. S сопоставляются два уровня защиты: базовый Ls(Si) пр. L, задаваемый в начале работы и остающийся неизменным, и текущий It(Si) пр. L, зависящий от уровней защиты тех объектов, к которым субъект Si имеет доступ в настоящий момент времени.

Множество объектов O наделяется структурой дерева таким образом, что каждому объекту Oj соответствует список объектов, непосредственно следующих за ним (объектов-сыновей) и, если Oj отличен от корня дерева, то существует единственный объект O(j), непосредственно предшествующий ему (отец объекта Oj). Каждому объекту Oj приписывается уровень защиты I(Oj) пр. L.

Множество L, является конечным частично упорядоченным множеством, обладавшим свойством алгебраической решетки. Возможно представление каждого уровня защиты Lr пр. L, в виде вектора из двух компонент: классификации и множества категорий. Будем говорить, что уровень защиты I1 больше уровня защиты I2, если классификация I1 больше или равна классификации I2, и множество категорий I1 содержит множество категорий I2(в формализованном виде: I1>=I2).

Множество прав доступа G имеет вид

G={r, a, w, e},

где

r- чтение объектом субъекта (получение субъектом данных, содержащихся в объекте) ;

a- модификация данных объекта субъектом без их предварительного прочтения;

w- запись-модификация данных объекта после их предварительного прочтения субъектом;

e- исполнение субъектом объекта (действие, не связанное ни с чтением, ни с модификацией данных).

Белл и Лападула сформулировали два условия защиты для модели:

простое условие защиты;

свойство ограничения

Простое условие защиты предложено для исключения прямой утечки секретных данных и состоит в следующем. Если субъекту Sj запрещен доступ:

по чтению r объекта Oj, тогда Is(Si)<=I(Oj);

по записи w в объект Oj, тогда Is(Si)<=I(Oj).

Простое условие защиты накладывает ограничения на базовые уровни защищенных объектов.

Свойство ограничения защиты предназначено для предотвращения косвенной утечки данных. Это условие накладывает ограничения на уровни защиты тех объектов, к которым субъект может иметь доступ одновременно. Если субъект S имеет доступ X1 к объекту O1 и

доступ X2 к объекту O2, то, в зависимости от вида доступа, должны выполняться соотношения между уровнями защиты, приведенные в таблице.

Введение свойства ограничения предназначено для предотвращения потока данных вида "чтение объекта для переписи данных в объект с меньшим либо несравнимым уровнем защиты" (напомним, что доступ w предполагает предварительное чтение).

Состояние системы считается безопасным, если соотношения между уровнями защиты объектов и субъектов удовлетворяют как простому условию защиты, так и условию ограничения.

Таблица 1

Доступ к O1	Доступ к O2	Соотношение
Чтение	Дополнение	I(S1)<=I(O2)
Чтение	Запись	I(S1)<=I(O2)
Запись	Запись	I(S1)<=I(O2)
Запись	Запись	I(S1)<=I(O2)

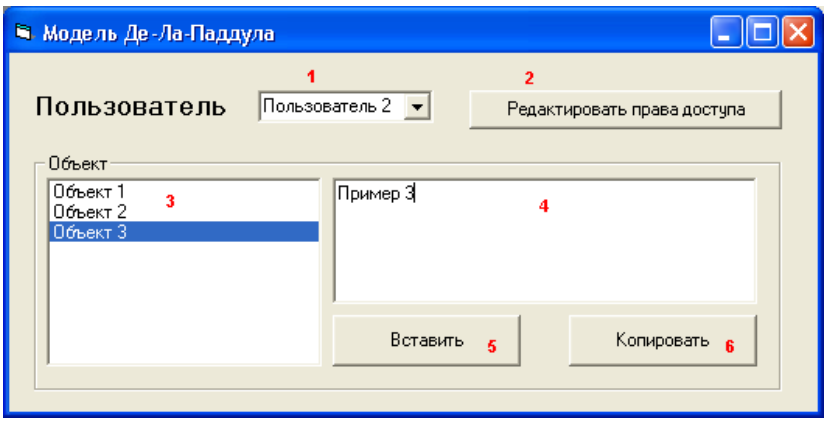
Система защиты должна обеспечивать безопасность данных, если на не допускает перехода из безопасного состояния в состояние, не являвшееся безопасным.

Для обеспечения безопасности данных необходимо и достаточно, чтобы изменение состояний системы приводило только к безопасным состояниям, если исходное состояние было безопасным.

Выполнение работы

Исходные данные:
N – количество пользователей
M – количество объектов
Матрица Users[N] – матрица пользователей
Матрица Object[M] – матрица значений объектов типа String
Матрица P[N, M+1] – матрица прав доступа,
Где
P[I, 0] – административные права по редактированию матрицы доступа для I-го пользователя (1, 0)
P[I, J] (1<=J<=m) – права I-го пользователя к J-му объекту (0 – нет прав, 1 – чтение, 2 – модификация)
Дополнительные данные:
PrivilegesUser[N] – номер привилегии для пользователя (0 - 3)
PrivilegesObject[M] – номер привилегии для объекта (0 - 3)
Выполнение работы:

Разработайте следующую форму.



В списке 1 выбирается пользователь **I** и согласно матрицы доступа **P**:
Если P[I, 0] = 0, то кнопка 2 недоступна.

В списке 3 выбирается объект J,

если P[I, J] = 0 или PrivilegesUser[I] < PrivilegesObject[J], то поле 4 недоступно, и значение объекта J не выводиться.

если P[I, J] = 1, то поле 4 недоступно, и значение объекта J выводиться.

если P[I, J] = 2, то поле 4 доступно, и значение объекта J выводиться. При редактировании поля 4, необходимо записывать результат в Object[J].

Согласно матрицам привилегий при выборе **I** пользователя и **j** объекта выполняются следующие действия:

Если PrivilegesUser[I] > PrivilegesObject[J], то кнопка 6 доступна. С помощью которой происходит копирование в буфер фрагмента текста j объекта, при этом необходимо запомнить номер привилегии объекта (PrivilegesBuffer).

Если PrivilegesBuffer < PrivilegesObject[J], то кнопка 5 доступна. С помощью которой происходит копирование из буфера фрагмента текста в j объект.

При выборе кнопке 2 происходит вызов формы редактирования матрицы доступа и матрицы привилегий в которой выполняются следующее операции:

Добавление удаление пользователей.

Добавление удаление объектов.

Назначение и изменение прав доступа.

Лабораторная работа №5 «Шифры перестановки»

Цель работы: изучить приемы шифрования методом перестановки.

Введение

Шифр, преобразования из которого изменяют только порядок следования символов исходного текста, но не изменяют их самих, называется шифром перестановки (ШП).

Рассмотрим преобразование из ШП, предназначенное для зашифрования сообщения длиной *n* символов. Его можно представить с помощью таблицы

1	2	…	*n*
i_1	i_2	…	i_n

где i_1 - номер места шифротекста, на которое попадает первая буква исходного сообщения при выбранном преобразовании, i_2 - номер места для второй буквы и т.д. В верхней строке таблицы выписаны по порядку числа от 1 до *n*, а в нижней – те же числа, но в произвольном порядке. Такая таблица называется подстановкой степени *n*.

Зная подстановку, задающую преобразование, можно осуществить как зашифрование, так и расшифрование текста. Например, если для преобразования используется подстановка

1	2	3	4	5	6
5	*2*	3	4	1	6

И в соответствии с ней зашифровывается слово МОСКВА, то получится КОСВМА. Попробуйте расшифровать сообщение НЧЕИУК, полученное в результате преобразования с помощью указанной выше подстановки.

В качестве упражнения читателю предлагается самостоятельно выписать подстановки, задающие преобразования в описанных ниже трех примерах шифров перестановки. Ответы помещены в конце раздела.

Читатель, знакомый с методом математической индукции, может легко убедиться в том, что существует 1*2*3*…*n (n!) вариантов заполнения нижней строки таблицы (6). Таким образом, число различных преобразований шифра перестановки, предназначенного для зашифрования

сообщения длины n, меньше либо равно n! (заметим, что в это число входит и вариант преобразования, оставляющий все символы на своих местах!).

С увеличением числа n значение n! растет очень быстро. Приведем таблицу значений n! для первых 10 натуральных чисел:

n	1	2	3	4	5	6	7	8	9	10
n!	1	2	6	24	120	720	5040	40320	362880	3628800

При больших n для приближенного вычисления n! можно пользоваться известной формулой Стирлинга

$$n! \approx \sqrt{2\pi n}(n/e)^n,$$

где $e = 2{,}717281828\ldots$

Примером ШП, предназначенного для зашифрования сообщений длины n, является шифр, в котором в качестве множества ключей взято множество всех перестановок степени n, а соответствующие им преобразования шифра задаются, как было описано выше. Число ключей такого шифра равно n! .

Для использования на практике такой шифр не удобен, так как при больших значениях n приходится работать с длинными таблицами.

Широкое распространение получили шифры перестановки, использующие некоторую геометрическую фигуру. Преобразования из этого шифра состоят в том, что в фигуру исходный текст вписывается по ходу одного «маршрута», а затем походу другого выписывается с нее. Такой шифр называют ***маршрутной перестановкой***. Например, можно вписывать исходное сообщение в прямоугольную таблицу, выбрав такой маршрут: по горизонтали, начиная с левого верхнего угла поочередно слева направо и справа налево. Выписывать же сообщение будем по другому маршруту: по вертикали, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

Зашифруем, например, указанным способом фразу: **ПРИМЕРМАРШРУТНОЙПЕРЕСТАНОВКИ**

Используя прямоугольник размера 4x7:

П	Р	И	М	Е	Р	М
Н	Т	У	Р	Ш	Р	А
О	Й	П	Е	Р	Е	С
И	К	В	О	Н	А	Т

Зашифрованная фраза выглядит так: **МАСТАЕРРЕШРНОЕРМИУПВКЙТРПНОИ**

Теоретически маршруты могут быть значительно более изощренными, однако запутанность маршрутов усложняет использование таких шифров.

Ниже приводятся описания трех разновидностей шифров перестановки.

Шифр «Сцитала»

Одним из самых первых шифровальных приспособлений был жезл («Сцитала»), применявшийся еще во времена войны Спарты против Афин в V веке до н.э. Это был цилиндр, на который виток к витку наматывалась узкая папирусная лента (без просветов и нахлестов), а затем на этой ленте вдоль его оси записывался необходимый для передачи текст. Лента сматывалась с цилиндра и отправлялась адресату, который, имея цилиндр точно такого же диаметра, наматывал ленту на него и прочитывал сообщение. Ясно, что такой способ шифрования осуществляет перестановку местами букв сообщения.

Шифр «Сцитала» реализует не более n перестановок (n, по-прежнему, - длина сообщения). Действительно, этот шифр, как не трудно видеть, эквивалентен следующему шифру маршрутной перестановки: в таблицу, состоящую из n столбцов, построчно записывают

сообщение, после чего выписывают буквы по столбцам. Число задействованных столбцов в таблице не может превосходить длины сообщения.

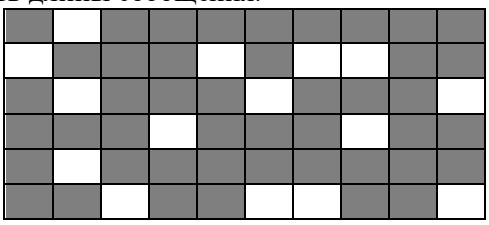


Рис. 1

Имеются еще чисто физические ограничения, накладываемые реализацией шифра «Сцитала». Естественно предположить, что диаметр жезла не должен превосходить 10 см. При высоте строки в 1 см на одном витке такого жезла уместится не более 32 букв ($10\pi<32$). Таким образом, число перестановок, реализуемых «Сциталой» вряд ли превосходит 32.

Шифр «Поворотная решетка»

Для использования шифра, называемого поворотная решетка, изготовляется трафарет из прямоугольного листа клетчатой бумаги размера 2m x 2k клеток. В трафарете вырезано mk клеток так, чтобы при наложении его на чистый лист бумаги того же размера четырьмя возможными способами его вырезы полностью покрывают всю площадь листа.

Буквы из сообщения последовательно вписываются в вырезы трафарета (по строкам, в каждой строке слева направо) при каждом из четырех его возможных положений заранее в установленном порядке.

Поясним процесс шифрования на примере. Пусть в качестве ключа используется решетка 6x10, приведенная на рисунке 1.

Зашифруем с ее помощью текст:

ШИФРРЕШЕТКАЯВЛЯЕТСЯЧАСТНЫМСЛУЧАЕМШИФРАМАРШРУТНОЙПЕРЕСТАНОВКИ

	Ш								
И				Ф		Р	Р		
	Е				Ш				Е
			Т				К		
	А								
		Я			В	Л			Я

Рис. 2

Е	Ш		Т	С			Я		
И				Ф		Р	Р	Ч	
	Е	А			Ш	С			Е
Т			Т	Н			К	Ы	
	А	М	С		Л				У
		Я			В	Л		Ч	Я

Рис. 3

Е	Ш	А	Т	С	Е	М	Я		Ш
И	И			Ф		Р	Р	Ч	
	Е	А	Ф		Ш	С	Р		Е
Т	А		Т	Н	М		К	Ы	А
Р	А	М	С	Ш	Л	Р	У		У
	Т	Я			В	Л		Ч	Я

Рис.4

Е	Ш	А	Т	С	Е	М	Я	Н	Ш
И	И	О	Й	Ф	П	Р	Р	Ч	Е
Р	Е	А	Ф	Е	Ш	С	Р	С	Е
Т	А	Т	Т	Н	М	А	К	Ы	А
Р	А	М	С	Ш	Л	Р	У	Н	У
О	Т	Я	В	К	В	Л	И	Ч	Я

Рис.5

Наложив решетку на лист бумаги, вписываем первые 15 (по числу вырезов) букв сообщения: **ШИФРРЕШЕТКАЯВЛЯ**. Сняв решетку, мы увидим текст, представленный на рисунке 2. Поворачиваем решетку на 180°. В окошечках появятся новые, еще не заполненные клетки. Вписываем в них следующие 15 букв. Получится запись, приведенная на рис.3. Затем переворачиваем решетку на другую сторону и зашифровываем остаток текста аналогичным образом (рис. 4,5).

Получатель сообщения, имеющий точно такую же решетку, без труда прочтет исходный текст, наложив решетку на шифротекст по порядку четырьмя способами.

Можно доказать, что число возможных трафаретов, то есть количество ключей шифра «решетка», составляет T=4mk. Этот шифр предназначен для сообщений длины n=4mk. Число всех перестановок в тексте такой длины составит (4mk)!,что во много раз больше числа T. Однако уже при размере трафарета 8 x 8 число возможных решеток превосходит 4 миллиарда.

Шифр вертикальной перестановки

Широко распространена разновидность шифра маршрутной перестановки, называемая ***«шифром вертикальной перестановки»*** (ШВП). В нем снова используется прямоугольник, в который сообщение вписывается обычным способом (по строкам слева направо). Выписываются буквы по вертикали, а столбцы при этом берутся в порядке, определяемом ключом. Пусть, например, этот ключ таков: (5,4,1,7,2,6,3), и с его помощью надо зашифровать сообщение:

ВОТПРИМЕРШИФРАВЕРТИКАЛЬНОЙПЕРЕСТАНОВКИ

Впишем сообщение в прямоугольник, столбцы которого пронумерованы в соответствии с ключом:

5	1	4	7	2	6	3
В	О	Т	П	Р	И	М
Е	Р	Ш	И	Ф	Р	А
В	Е	Р	Т	И	К	А
Л	Ь	Н	О	Й	П	Е
Р	Е	С	Т	А	Н	О
В	К	И	-	-	-	-

Теперь, выбирая столбцы в порядке, заданном ключом, и выписывая последовательно буквы каждого из них сверху вниз, получаем такую криптограмму:

ОРЕЬЕКРФИЙА-МААЕО-ТШРНСИВЕВЛРВИРКПН-ПИТОТ-

Число ключей ШВП не более m!, где m – число столбцов таблицы. Как правило, m гораздо меньше, чем длина текста n (сообщение укладывается в несколько строк по m букв), а, значит, и m! много меньше n!.

В случае, когда ключ ШВП не рекомендуется записывать, его можно извлекать из какого-то легко запоминающегося слова или предложения. Для этого существует много способов. Наиболее распространенный состоит в том, чтобы приписывать буквам числа в соответствии с обычным алфавитным порядком букв. Например, пусть ключевым словом будет **ПЕРЕСТАНОВКА**. Присутствующая в нем буква А получает номер 1. Если какая-то буква входит несколько раз, то ее появления нумеруются последовательно слева направо. Поэтому второе вхождение буквы А получает номер 2. Поскольку буквы Б в этом слове нет, то буква В получает номер 3 и так далее. Процесс продолжается до тех пор, пока все буквы не получат номера таким образом мы получаем следующий ключ:

П	Е	Р	Е	С	Т	А	Н	О	В	К	А
9	4	10	5	11	12	1	7	8	3	6	2

Для обеспечения дополнительной скрытности можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой метод шифрования называется двойной перестановкой. В случае двойной перестановки столбцов и строк таблицы, перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Пример выполнения шифрования методом двойной перестановки показан на рис.6. Если считывать шифртекст из правой таблицы построчно, то получим следующее: **ТЮАЕООГМРЛИПОЬСВ**

Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (в нашем примере последовательности 4,1,3,2 и 3,1,4,2 соответственно).

Число вариантов двойной перестановки быстро возрастаем при увеличении размера таблицы:

для таблицы 3 х 3 36 вариантов;
для таблицы 4 х 4 576 вариантов;

для таблицы 5 х 5 14400 вариантов.

Однако двойная перестановка не отличается высокой стойкостью и сравнительно просто взламывается при любом размере таблицы шифрования.

	4	1	3	2
3	П	Р	И	Л
1	Е	Т	А	Ю
4	В	О	С	Ь
2	М	О	Г	О

Исходная таблица

	1	2	3	4
3	Р	Л	И	П
1	Т	Ю	А	Е
4	О	Ь	С	В
2	О	О	Г	М

Перестановка столбцов

	1	2	3	4
1	Т	Ю	А	Е
2	О	О	Г	М
3	Р	Л	И	П
4	О	Ь	С	П

Перестановка строк

Рис.6

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

О	И	Р	М
Е	О	С	Ю
В	Т	А	Ь
Л	Г	О	П

Рис.7

Применение магических квадратов

В средние века для шифрования перестановкой применялись и магические квадраты.

Магическими квадратами называют квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают в

сумме по каждому столбцу, каждой строке, и каждой диагонали одно и то же число.

Шифруемый текст вписывали в магические квадраты в соответствии с нумерацией их клеток. Если затем выписать содержимое такой таблицы по строкам, то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения. В те времена считалось, что созданные с помощью магических квадратов шифртексты охраняет не только ключ, но и магическая сила.

Пример магического квадрата и его заполнение сообщением **ПРИЛЕТАЮ ВОСЬМОГО** показан на рис.7.

Шифртекст, получаемый при считывании содержимого правой таблицы по строкам, имеет вполне загадочный вид: **ОИРМЕОСЮВТАЬЛГОП**

Число магических квадратов быстро возрастает с увеличением размера квадрата. Существует только один магический квадрат размером 3х3 (если не учитывать его повороты). Количество магических квадратов 4х4 составляет уже 880, а количество магических квадратов 5х5 – около 250000.

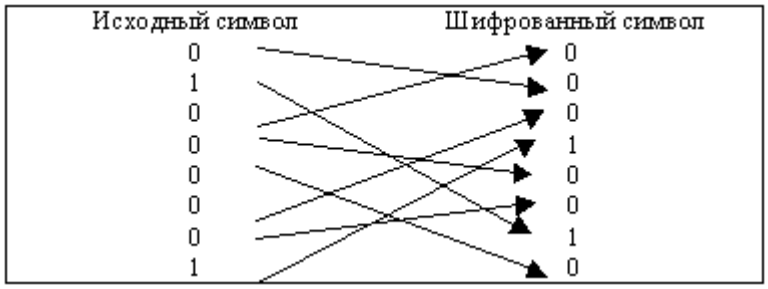
Магические квадраты средних и больших размеров могли служить хорошей базой для обеспечения нужд шифрования того времени, поскольку практически нереально выполнить вручную перебор всех вариантов для такого шифра.

Перестановка бит

Использованием компьютеров в процессе шифрования продиктован следующий метод – ***перестановки бит*** в символе.

Пусть сообщение состоит из символов, любой из которых в машинном представлении представляет собой последовательность из восьми бит. Переставив некоторым образом биты в каждом символе, мы получим зашифрованную последовательность.

Чтобы расшифровать ее, нам будет необходимо лишь произвести обратную перестановку.



Вскрытие шифра перестановки

Прейдем к вопросу о методах вскрытия шифра перестановки. Проблема, возникающая при восстановлении сообщения, зашифрованного ШП, состоит не только в том, что число возможных ключей велико даже при небольших длинах текста. Если и удастся перебрать все допустимые варианты перестановок, не всегда ясно, какой из этих вариантов истинный. Например, пусть требуется восстановить исходный текст по криптограмме АОГР, и нам ничего не известно, кроме того, что применялся шифр перестановки. Какой вариант «осмысленного» исходного текста признать истинным: ГОРА или РОГА? А может быть АРГО? Приведем пример еще более запутанной ситуации. Пусть требуется восстановить сообщение по криптограмме **ААНИНК-ТЕОМЛ,З.ЬЬЗИВТЛП-ЬЯО**, полученной шифром перестановки.

Возможны, как минимум, два варианта исходного сообщения:

КАЗНИТЬ,- НЕЛЬЗЯ- ПОМИЛОВАТЬ.

КАЗНИТЬ- НЕЛЬЗЯ,- ПОМИЛОВАТЬ.

Эти варианты имеют прямо противоположный смысл, и в имеющихся условиях у нас нет возможности определить, какой из вариантов истинный.

Иногда, за счет особенностей реализации шифра, удается получить информацию об используемом преобразовании перестановки.

Рассмотрим шифр «Сцитала». Выше уже рассматривался вопрос о количестве перестановок, реализуемых «Сциталой». Их оказалось не более 32. Это число невелико, поэтому можно осуществить перебор всех вариантов. При достаточной длине сообщения мы, скорее всего, получим единственный читаемый вариант текста.

Рассмотрим пример с ШВП. По условию пробелы между словами при записи текста в таблицу опускались, поэтому заключаем, что все столбцы, содержащие пробел в последней строке, должны стоять в конце текста. Таким образом, возникает разбиение столбцов на две группы (содержащие 6 букв, и содержащие 5 букв). Для завершения восстановления исходного текста достаточно найти порядок следования столбцов в каждой из групп в отдельности, что гораздо проще.

Аналогичная ситуация возникает и при неполном использовании шифра «решетка».

Пусть имеется решетка размера m x r и зашифрованное с ее помощью сообщение длины mr-k, не содержащая пробелов. Не заполненные k мест в решетке при условии, что $mr/4 \geq k$, соответствует вырезам в четвертом положении решетки. На основе такой информации происходит резкое уменьшение числа допустимых решеток (их будет $4^{mr/(4-k)}$).

Еще один подход к вскрытию шифров вертикальной перестановки – лингвистический. Он основан на том, что в естественных языках некоторые комбинации букв встречаются очень часто, другие – гораздо реже, а многие вообще не встречаются (например – «ыьъ»).

Будем подбирать порядок следования столбцов друг за другом так, чтобы во всех строках этих столбцов получались «читаемые» отрезки текста.

Сочетание лингвистического метода с учетом дополнительной информации довольно быстро может привести к вскрытию сообщения.

Контрольные вопросы

В чем заключается метод шифрования перестановкой?

Что такое маршрутная перестановка?

Какой «маршрут» можно использовать для реализации шифра «Сцитала»?

Что называется «поворотной решеткой»?

Оцените количество ключей шифра вертикальной перестановки. Во сколько раз это количество ключей возрастает при использовании двойной перестановки?

Приведите пример использования магического квадрата для шифрования сообщения **‘ЯУЕЗЖАЮВНОВГОРОД’**.

Что такое шифрование перестановкой бит?

Предложите путь вскрытия шифра перестановки. Какие сложности возникают при этом и какие «оплошности» шифровальщиков можно использовать?

Варианты заданий

Реализовать алгоритмы шифрования и дешифрования файлов с помощью метода, указанного в варианте. Предусмотрите выбор ключа шифрования.

1. Входную последовательность разбейте на группы по четыре символа, далее в каждой группе символы переставьте с использованием подстановки, которую выберите самостоятельно.
2. Реализуйте маршрутную перестановку с использованием шифрующей таблицы 5x8. Маршрут выберите самостоятельно.
3. Реализуйте процедуру, моделирующую использование «Сцитала». Число столбцов шифрующей таблицы выберите самостоятельно.
4. Выберите поворотную решетку и смоделируйте ее использование.
5. Реализуйте шифрование двойной перестановкой. Размерность шифрующей таблицы выбрать самостоятельно.
6. Смоделируйте использование магических квадратов. Предусмотрите их генерацию.
7. Реализуйте метод перестановки бит.

Определите ключ (варианты ключей, при неоднозначности толкования), использованный для шифрования в варианте 1. Исходные данные: шифртекст, длина групп разбиения (4 символа).

Определите ключ, использованный при шифровании в варианте 5. Исходные данные - размерность шифрующей таблицы – 3x3.

Лабораторная работа №6 «Шифры простой замены»

Цель работы: изучение алгоритмов зашифрования и расшифрования сообщений с помощью шифров замены.

Шифрами замены называют такие шифры, которые осуществляются путем замены каждого символа открытого сообщения на другие символы – шифрообозначения, причем порядок следования шифрообозначений совпадает с порядком следования соответствующих им символов открытого сообщения.

Пусть, например, зашифровывается сообщение на русском языке и при этом замене подлежит каждая буква сообщения. Формально в этом случае шифр замены можно описать следующим образом. Для каждой буквы α исходного алфавита строится некоторое множество символов $M\alpha$, которое называется множеством шифрообозначений для буквы α.

Таблица является ключом шифра замены. Зная ее можно осуществить как зашифрование, так и расшифрование.

При зашифровании каждая буква α открытого сообщения, начиная с первой, заменяется любым символом из множества M_α. За счет этого можно получить различные варианты зашифрованного сообщения для одного и того же открытого сообщения.

В простейшем случае множество шифрообозначений M_α состоит из одного элемента. Такой шифр называется шифром простой замены.

А	Б	…	Я
$M_А$	$M_Б$	…	$M_Я$

Шифры простой замены

Система шифрования Цезаря

В качестве ключа шифра Цезаря используют таблицу, первая строка которой содержит буквы исходного алфавита, а вторая строка – последовательность букв, записанных в алфавитном порядке, но начинающаяся не с буквы А, а с какой-либо другой:

А	Б	…	Э	Ю	Я
Д	Е	…	Б	В	Г

При шифровании букву исходного сообщения находят в первой строке и заменяют ее на соответствующую букву второй строки. Запомнить ключ достаточно просто – надо лишь запомнить первую букву второй строки.

Серьезный недостаток данного шифра – ограниченное число ключей, которое равно числу букв в алфавите.

Афинная система подстановок Цезаря

Здесь буквы исходного сообщения преобразуются следующим образом:

$T_1=AT+B \pmod m$,

где T – порядковый номер буквы исходной последовательности,

T_1 - порядковый номер соответствующей буквы зашифрованной последовательности,

m – размер алфавита,

A,B – целые числа, причем A и m взаимно простые.

Пример.

Зашифруем фразу КОРАБЛИ ОТПЛЫВАЮТ ВЕЧЕРОМ, используя афинную систему подстановок при A=13, B=5. Размер алфавита m=32 (будем считать, что в исходном алфавите в качестве буквы Й используется И, а в качестве Ё – Е, и добавим 32-ым символом пробел). В результате преобразований получим:

Сообщение К О Р А Б Л И О Т П Л Ы В А Ю Т В Е Ч Е Р О М

Шифртекст Ы П И Е У З О Щ П В Ь З Ш Е Я В Щ Ж Г Ж И П Х

Лозунговый шифр

В данном шифре запоминание ключа основано на лозунге – легко запоминающемся слове или фразе. Например, выберем слово – лозунг “заявление” и заполним вторую строку таблицы по следующему правилу: *сначала вписывается слово - лозунг, причем повторяющиеся буквы отбрасываются, затем эта таблица дополняется не вошедшими в нее буквами алфавита*. Ключ будет иметь вид:

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
З	А	Я	В	Л	Е	Н	И	Б	Г	Д	Ж	К	М	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	

Полибианский квадрат

Шифр изобрел греческий писатель и историк Полибий. Прямоугольная таблица заполняется буквами алфавита в случайном порядке. Каждая буква открытого сообщения заменяется буквой, расположенной ниже в том же столбце. Если буква находится на последней строке таблицы, то для ее шифрования берут самую верхнюю букву столбца. Например, слово "**АЛФАВИТ**" после зашифрования преобразуется в "**УЪ УТСЗ**".

У	К	В	Ъ	М	Ю	Ь	Д
И	Б	Т	Л	Э	Г	Щ	Н
С	Ф	З	Ы	П	Ц	Е	Я
А		Р	Х	Ж	Ш	О	Ч

Шифрующая таблица Трисемуса

Для получения такого шифра используется таблица для записи букв и ключевое слово (или фраза). В таблицу сначала вписывается ключевое слово, причем повторяющиеся буквы отбрасываются. Затем эта таблица дополняется не вошедшими в нее буквами алфавита. На рисунке изображена таблица Трисемуса с ключевым словом "БАНДЕРОЛЬ". Применение таблицы аналогично применению полибианского квадрата.

Достоинством этого и других рассмотренных шифров простой замены является простота реализации, недостатком – легкая раскрываемость, учитывая то, что данные шифры сохраняют информацию о частоте встречаемости букв исходного текста. Это позволяет применять методы подсчета частот для расшифровки сообщений. С целью устранения данного недостатка применяют следующие методы:

Биграммный шифр Плейфейра.

Этот шифр основан на таблице, аналогичной таблице Трисемуса.

Процедура шифрования включает следующие шаги:

1) Открытый текст разбивается на пары букв (биграммы). Текст должен иметь четное число букв, и в нем не должно быть биграмм, содержащих две одинаковые буквы.

2) Последовательность биграмм открытого текста преобразуется в последовательность биграмм с помощью шифрующей таблицы по следующим правилам:

Если обе буквы биграммы открытого сообщения не попадают в одну строку или столбец, тогда для замены находят буквы в углах прямоугольника, определяемого данной парой букв.

Если обе буквы биграммы открытого сообщения принадлежат одному столбцу таблицы, то их заменяют на буквы, которые лежат под ними. Если при этом буква открытого текста находится в нижней строке, то для шифрования берется буква из верхней строки того же столбца.

Если обе буквы биграммы открытого сообщения принадлежат одной строке таблицы, то они заменяются на буквы, которые лежат справа от них. Если при этом буква открытого текста находится в крайнем правом столбце, то для шифрования берется буква из крайнего левого столбца той же строки.

Контрольные вопросы

Какие шифры называют шифрами замены?
Что такое ключ шифра замены?
Что называют множеством шифрообозначений?
Приведите примеры шифров простой замены. Опишите алгоритм одного из них.
Каковы основные недостатки шифров простой замены?

Варианты заданий

Реализовать алгоритмы шифрования и дешифрования файлов с помощью метода, указанного в варианте.

1. Система шифрования Цезаря.
2. Афинная система подстановок Цезаря.
3. Лозунговый шифр.
4. Полибианский квадрат.
5. Шифрующая таблица Трисемуса.
6. Биграммный шифр Плейфейра.

Лабораторная работа №7 «Шифры сложной замены»

Цель

Освоить навыки шифрования с применением шифров сложной замены

Теоретические сведения

Шифры сложной замены называют многоалфавитными, так как для шифрования каждого символа исхдного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно циклически меняет используемые алфавиты.

При r-алфавитной подстановке символ x_0 исходного сообщения заменяется символом y_0 из алфавита B_0, символ x_1 - символом y_1 из алфавита B_1, и так далее, символ x_{r-1} заменяется символом y_{r-1} из алфавита B_{r-1}, символ x_r заменяется символом y_r снова из алфавита B_0, и т.д.

Общая схема многоалфавитной подстановки для случая r = 4:

Входной символ	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9
Алфавит подстановки	B0	B1	B2	B3	B0	B1	B2	B3	B0	B1

Эффект использования многоалфавитной подстановки заключается в том, что обеспечивается маскировка естественной статистики исходного языка, так как конкретный символ из исходного алфавита может быть преобразован в несколько различных символов шифровальных алфавитов B_i.

Система омофонов

Данный шифр характеризуется тем, что буквы исходного сообщения имеют несколько замен. Число замен берется пропорциональным вероятности появления буквы в открытом тексте. Данные о распределениях вероятностей букв в русском тексте приведены в таблице.

Буква	Вероят-ность	Буква	Вероят-ность	Буква	Вероят-ность	Буква	Вероят-ность
Пробел	0.175	Р	0.040	Я	0.018	Х	0.009
О	0.090	В	0.038	Ы	0.016	Ж	0.007
Е	0.072	Л	0.035	З	0.016	Ю	0.006
А	0.062	К	0.028	Ъ	0.014	Ш	0.006

И	0.062	М	0.026	Б	0.014	Ц	0.004
Н	0.053	Д	0.025	Г	0.013	Щ	0.003
Т	0.053	П	0.023	Ч	0.012	Э	0.003
С	0.045	У	0.021	Й	0.010	Ф	0.002

Шифруя букву исходного сообщения, выбирают случайным образом одну из ее замен. Замены (часто называемые омофонами) могут быть представлены трехразрядными числами от 000 до 999. Например, букве О присваиваются 90 случайных номеров, буквам Б и Ъ – по 14 номеров. Если омофоны присваиваются случайным образом различным появлениям одной и той же буквы, тогда каждый омофон появляется в шифртексте равновероятно.

Система омофонов обеспечивает простейшую защиту от криптоаналитических атак, основанных на подсчете частот появления букв в шифртексте.

Шифр Гронсфельда

Шифрование осуществляется следующим образом. Под буквами исходного сообщения записывают цифры числового ключа. Если ключ короче сообщения, то его запись циклически повторяют. Для замены выбирают ту букву, которая смещена по алфавиту на соответствующую цифру ключа. Например, применяя в качестве ключа группу из четырех начальных цифр числа *e* (основания натуральных логарифмов), а именно 2718, получаем для исходного сообщения ВОСТОЧНЫЙ ЭКСПРЕСС следующий шифртекст:

Сообщение В О С Т О Ч Н Ы Й Э К С П Р Е С С
Ключ 2 7 1 8 2 7 1 8 2 7 1 8 2 7 1 8 2
Шифртекст Д Х Т Ь Р Ю О Г Л Д Л Щ С Ч Ж Щ У

Шифр Гронсфельда представляет собой частный случай системы шифрования Вижинера.

Система шифрования Вижинера

Этот шифр сложной замены можно описать таблицей шифрования, называемой таблицей (квадратом) Вижинера. Таблица Вижинера используется для зашифрования и расшифрования.

Ключ	А	Б	В	Г	Д	…	Э	Ю	Я
0	А	Б	В	Г	Д	…	Э	Ю	Я
1	Б	В	Г	Д	Е	…	Ю	Я	А
2	В	Г	Д	Е	Ж	…	Я	А	Б
3	Г	Д	Е	Ж	З	…	А	Б	В
…	…	…	…	…	…	…	…	…	…
30	Ю	Я	А	Б	В	…	Ы	Ъ	Э
31	Я	А	Б	В	Г	…	Ъ	Э	Ю

Таблица имеет два входа:

верхнюю строку подчеркнутых символов, используемую для считывания очередной буквы исходного открытого текста;

крайний левый столбец ключа.

Последовательность ключей обычно получают из числовых значений букв ключевого слова.

При шифровании исходного сообщения его выписывают в строку, а под ним записывают ключевое слово или фразу. Если ключ оказался короче сообщения, то его циклически повторяют. В процессе шифрования находят в верхней строке таблицы очередную букву исходного текста и в левом столбце очередное значение ключа. Буква шифртекста находится на пересечении столбца, определяемого шифруемой буквой, и строки, определяемой числовым значением ключа.

Пример.

Зашифруем сообщение **ПРИЛЕТАЮ СЕДЬМОГО** с использованием ключа **АМБРОЗИЯ**.

Сообщение П Р И Л Е Т А Ю С Е Д Ь М О Г О
Ключ АМ Б Р О З И Я А М Б Р О З И Я
Шифртекст ПЪ Й ЫУЩ И Э С С Е К Ь Х Л Н

Шифр “двойной квадрат” Уитстона

В отличие от полибианского шифр “двойной квадрат” использует сразу две таблицы со случайно расположенными буквами, размещенные по одной горизонтали, а шифрование идет биграммами как и в шифре Плейфейра.

Процедура шифрования выполняется следующим образом. Перед шифрованием исходное сообщение разбивается на биграммы. Каждая биграмма шифруется отдельно. Первую букву биграммы находят в левой таблице, а вторую букву – в правой таблице. Затем мысленно строят прямоугольник так, чтобы буквы биграммы лежали в его противоположных вершинах. Другие две вершины этого прямоугольника дают буквы биграммы шифртекста.

Ж	Щ	Н	Ю	Р
И	Т	Ь	Ц	Б
Я	М	Е	.	С
В	Ы	П	Ч	
:	Д	У	О	К
З	Э	Ф	Г	Ш
Х	А	,	Л	Ъ

И	Ч	Г	Я	Т
,	Ж	Ь	М	О
З	Ю	Р	В	Щ
Ц	:	П	Е	Л
Ъ	А	Н	.	Х
Э	К	С	Ш	Д

Если обе буквы биграммы лежат в одной строке, то и буквы шифртекста берут из той же строки. Первую букву биграммы шифртекста берут из левой таблицы в столбце, соответствующем второй букве биграммы сообщения. Вторая же буква биграммы шифртекста берется из правой таблицы в столбце, соответствующем первой букве биграммы сообщения.

Пример:

Сообщение ПР ИЛ ЕТ АЮ _Ш ЕС ТО ГО
Шифртекст ПЕ ОВ ЩН ФМ ЕШ РФ БЖ ДЦ

Контрольные вопросы

В чем отличие шифров простой и сложной замены?
Опишите алгоритмы шифрования, основанные на лозунге.
Какие шифры сложной замены вам известны?
Каким образом для шифрования используют “двойной квадрат” Уитстона?
В чем заключается шифрование с использованием системы Вижинера?

Варианты заданий

Реализовать алгоритмы шифрования и дешифрования файлов с помощью метода, указанного в варианте.

1. Система омофонов
2. Шифр Гронсфельда.

3. Система шифрования Вижинера.
4. Шифр “двойной квадрат” Уитстона.

Лабораторная работа №8 «Шифрование методом гаммирования»

Цель работы: Изучить применение метода гаммирования для шифровки и дешифровки текста.

Описание метода.

Гаммирование является широко применяемым криптографическим преобразованием.

Под *гаммированием* понимают процесс наложения по определенному закону гаммы шифра на открытые данные. *Гамма шифра* - это псевдослучайная последовательность, выработанная по заданному алгоритму для шифровки открытых данных и дешифровки зашифрованных данных.

Процесс *шифровки* заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы на исходный открытый текст обратимым образом, например с использованием операции сложения по модулю 2.

Следует отметить, что перед шифровкой открытые данные разбивают на блоки $T_o^{(i)}$ одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков $Г_ш^{(i)}$ аналогичной длины.

Уравнение шифровки можно записать в виде

$$T_ш^{(i)} = Г_ш^{(i)} \oplus T_o^{(i)},\ i = 1 \ldots M,$$

где $T_ш^{(i)}$ *i*-й блок шифртекста;
$Г_ш^{(i)}$ *i*-й блок гаммы шифра;
$T_o^{(i)}$ *i*-й блок открытого текста;
M количество блоков открытого текста.

Процесс *дешифровки* сводится к повторной генерации гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение дешифровки имеет вид

$$T_o^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)},\ i = 1 \ldots M.$$

Достоинства и недостатки метода.

Получаемый этим методом шифртекст достаточно труден для раскрытия, поскольку ключ здесь является переменным. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока. Если период гаммы превышает длину всего шифруемого текста и злоумышленнику неизвестна никакая часть исходного текста, то такой шифр можно раскрыть только прямым перебором всех вариантов ключа. В этом случае криптостойкость шифра определяется длиной ключа.

Однако, метод гаммирования становится бессильным, если злоумышленник узнает фрагмент исходного текста и соответствующую ему шифрограмму. Простым вычитанием по модулю получается отрезок псевдослучайной последовательности и по нему восстанавливается вся последовательность.

Методы генерации псевдослучайных последовательностей чисел

При шифровании методом гаммирования в качестве ключа используется случайная строка битов, которая объединяется с открытым текстом, также представленным в двоичном виде, с помощью побитового сложения по модулю 2, и в результате получается шифрованный текст.

Генерирование непредсказуемых двоичных последовательностей большой длины является одной из важных проблем классической криптографии. Для решения этой проблемы широко используются генераторы двоичных псевдослучайных последовательностей.

Генерируемые псевдослучайные ряды чисел часто называют гаммой шифра или просто гаммой (по названию буквы γ греческого алфавита, часто используемой в математических формулах для обозначения случайных величин).

Обычно для генерации последовательности псевдослучайных чисел применяют компьютерные программы, которые, хотя и называются генераторами случайных чисел, на самом деле вырабатывают детерминированные числовые последовательности, которые по своим свойствам очень похожи на случайные.

К криптографически стойкому генератору ПСП чисел (гаммы шифра) предъявляются три основных требования:

период гаммы должен быть достаточно большим достаточно большим для шифрования сообщений различной длины;

гамма должна быть практически непредсказуемой, что означает невозможность предсказать следующий бит гаммы, даже если известны тип генератора и предшествующий кусок гаммы;

генерирование гаммы не должно вызывать больших технических сложностей;

Длина периода гаммы является самой важной характеристикой генератора ПСЧ. По окончании периода числа начнут повторяться и их можно будет предсказать.

Один из первых способов генерации ПСЧ на ЭВМ предложил в 1946 г. Джон фон Нейман. Суть этого способа состоит в том, что каждое последующее случайное число образуется возведением в квадрат предыдущего числа с отбрасыванием цифр младших и старших разрядов. Однако этот способ оказался ненадежным и от него вскоре отказались.

Из известных процедур генерации последовательности ПСЧ наиболее часто применяется так называемый *линейный конгруэнтный генератор*. Этот генератор вырабатывает последовательность ПСЧ Y_1, Y_2, ..., Y_{i-1}, Y_i, ..., используя соотношение

$$Y_i = (a \cdot Y_{i-1} + b) \bmod m,$$

где Y_i i-е (текущее) число последовательности;
Y_{i-1} предыдущее число последовательности;
m – модуль;
a – множитель;
b – приращение;
a Y_0 - порождающее число (исходное значение).

Текущее псевдослучайное число Y_i получают из предыдущего числа Y_{i-1} умножением его на коэффициент a, сложением с приращением b и вычислением остатка от деления на m. Данное уравнение генерирует ПСЧ с периодом повторения, зависящим от выбранных значений a и b и может достигать значения m. Значение m обычно устанавливается равным 2^n, где n - длина машинного слова в битах, либо равным простому числу, например $m=2^{31}-1$. Как показано Д. Кнутом, линейный конгруэнтный датчик ПСЧ имеет максимальный период тогда и только тогда, когда b - нечетное, и a mod 4 = 1.

Также для получения последовательности ПСЧ применяются аддитивные и мультипликативные генераторы.

Мультипликативный генератор вырабатывает последовательности чисел с помощью рекуррентного соотношения:

$$Y_i = (a \cdot Y_{i-1}) \bmod m.$$

Требования к значениям констант a и m такие же, как и для линейного конгруэнтного генератора.

Текущее случайное число Y_i *аддитивного датчика* получается из суммы чисел Y_{i-1} и Y_{i-2} вычислением модуля от деления этой суммы на m:

$$Y_i = (Y_{i-1} + Y_{i-2}) \bmod m.$$

Описание алгоритмов.

Алгоритм шифровки.

1. Проинициализировать датчик случайных чисел.
2. Выделить блок открытого текста.
3. Сгенерировать гамму шифра.
4. Получить блок зашифрованного текста, сложив по модулю 2 блок открытого текста с гаммой шифра.
5. Если текст не закончился, перейти к пункту 2, иначе к пункту 6.
6. Конец алгоритма шифровки.

Алгоритм дешифровки.

1. Проинициализировать датчик случайных чисел.
2. Выделить блок зашифрованного текста.
3. Сгенерировать гамму шифра.
4. Получить блок открытого текста, сложив по модулю 2 блок зашифрованного текста с гаммой шифра.
5. Если зашифрованный текст не закончился, перейти к пункту 2, иначе к пункту 6.
6. Конец алгоритма дешифровки.

Контрольные вопросы.

1. Что такое гаммирование? Что понимают под гаммой шифра?

2. Какие операции можно применять при наложении гаммы? В чём заключается процесс шифровки и дешифровки?

4. Какие достоинства и недостатки у метода гаммирования?

5. Какие требования предъявляются к криптографически стойкому генератору ПСП? Почему наиболее важна длина периода гаммы?

6. Опишите линейный конгруэнтный способ генерации ПСП.

7. Опишите аддитивный и мультипликативный генераторы ПСП.

8. Опишите алгоритмы шифровки и дешифровки открытого текста методом гаммирования.

Варианты заданий.

Зашифровать и расшифровать текст указанным методом.

Для генерации гаммы использовать:

1. аддитивный датчик со значениями m = 4096, Y_0 = 4003, Y_1 = 59;
2. линейный конгруэнтный датчик со значениями a = 5, b = 7, m = 4096, Y_0 = 4003, Y_1 = 59;
3. мультипликативный датчик со значениями a = 7, m = 4096, Y_0 = 502;
4. датчик, период у которого наибольший; датчики и значения для них выбрать из предыдущих вариантов. Гамму генерировать с помощью указанного датчика. Подобрать для него такие значения параметров, чтобы период был наибольшим.
5. мультипликативный датчик;
6. линейный конгруэнтный датчик.

Лабораторная работа №9 «Одноразовая система шифрования»

Цель работы: Знакомство с традиционными симметричными криптографическими системами: шифры сложной замены. Изучение алгоритма кодирования одноразовой системы шифрования.

Традиционные симметричные криптосистемы: шифры сложной замены

Под шифром понимают совокупность обратимых преобразований множества открытых данных на множество закрытых данных, задаваемых ключом и алгоритмом криптографического преобразования.

Ключ – это конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор только одного варианта из всех возможных для данного алгоритма.

Основной характеристикой шифра является криптостойкость, которая определяет его стойкость к раскрытию методами криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимым для раскрытия шифра.

К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований:

- достаточная криптостойкость (надежность закрытия данных);
- простота процедур шифрования ирасшифрования;
- незначительная избыточность информации за счет шифрования;
- нечувствительность к небольшим ошибкам шифрования и др.

В той или иной мере этим требованиям отвечают:

- шифры перестановок;
- шифры замены;
- шифры гаммирования;
- шифры, основанные на аналитических преобразованиях шифруемых данных.

Процессы шифрования и расшифрования заменой(подстановкой) осуществляются в рамках некоторой криптосистемы. Характерной особенностью симметричных криптосистем является применение одного и того же секретного ключа как при шифровании, так и при расшифровании сообщений.

Как открытый текст, так и шифротекст образуется из букв, входящих в конечное множество символов, называемых алфавитом.

В общем виде некоторый алфавит А можно представить так:

$$A=\{a_0, a_1, \ldots, a_{m-1}\}$$

При выполнении криптографических преобразований полезно заменить буквы алфавита целыми числами. Это позволяет упростить выполнение необходимых алгебраических манипуляций. Например, можно установить взаимно однозначное соответствие между русским алфавитом

$$A=\{АБВГ\ldots..ЮЯ\}$$

и множеством целых

$$Z_{32} =\{0,1,2,3,\ldots.,31\}$$

Текст с n буквами из алфавита Z_m можно рассматривать как n-грамму

$$x=(x_0 , x_1 , x_2 ,\ldots., x_{n-1}),$$

где $x_i \in Z_m$, $0<=i<n$, для некоторого целого $n=1,2,3,\ldots.$.

Через $Z_{m,n}$ будем обозначать множество n-грамм, образованных из букв множества Z_m .

Криптографическое преобразование E представляет собой совокупность преобразований:

$$E=\{E^{(n)}:1<=n<\infty\}$$

$$E^{(n)}: Z_{m,n} \rightarrow Z_{m,n} .$$

Преобразование $E^{(n)}$ определяет, как каждая n-грамма открытого текста $x\in Z_{m,n}$ заменяется n-граммой шифротекста y, т.е.

$$y= E^{(n)}(x), \text{ причем } x,y \in Z_{m,n};$$

Криптографическая система может рассматриваться как семейство криптографических преобразований

$$\mathbf{E}=\{E_k : K \in \mathbf{K}\},$$

помеченных параметром K, называемым ключом.

Множество значений ключа образует ключевое простр-во **K**.

Шифры сложной замены называют многоалфавитными, так как для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно и циклически меняет используемые алфавиты.

При r-алфавитной подстановке символ x_0 исходного сообщения заменяется символом y_0 из алфавита B_0 , символ x_1 - символом y_1 из алфавита B_1 и так далее.

Например, для r=4:

Входной символ: $X_0\ X_1\ X_2\ X_3\ X_4\ X_5\ X_6\ X_7\ X_8\ X_9$

Алфавит подстановки: $B_0\ B_1\ B_2\ B_3\ B_4\ B_5\ B_6\ B_7\ B_8\ B_9$

Эффект использования многоалфавитной подстановки заключается в том, что обеспечивается маскировка естественной статистики исходного языка, так как конкретный символ из исходного алфавита A может быть преобразован в несколько различных символов шифровальных алфавитов B_j. Степень обеспечиваемой защиты теоретически пропорциональна длине периода r в последовательности используемых алфавитов B_j.

Одноразовая система шифрования

Почти все применяемые на практике шифры характеризуются как условно надежные, поскольку они могут быть в принципе раскрыты при наличии неограниченных вычислительных возможностей. Абсолютно надежные шифры нельзя разрушить даже при использовании неограниченных вычислительных возможностей. Существует единственный такой шифр, применяемый на практике, - одноразовая система шифрования. Характерной особенностью одноразовой системы шифрования является одноразовое использование ключевой последовательности.

Одноразовая система шифрует исходный открытый текст

X = (Xo, Xi, ... Xn-i)

в шифртекст

Y=(Yo.Yi,...,Yn-i)

посредством подстановки Цезаря

$Yi=(Xi+Ki) \bmod m,\ 0<= i < n,$

где K_I - i -и элемент случайной ключевой последовательности.

Ключевое пространство К одноразовой системы представляет собой набор дискретных случайных величин из Z_m и содержит m^n значений.

Процедура расшифрования описывается соотношением

X, = (Yi - К,) mod m,

где K_i - i-й элемент той же самой случайной ключевой последовательности.

Одноразовая система изобретена в 1917 г. американцами Дж.Моборном и Г.Вернамом. Для реализации этой системы подстановки иногда используют одноразовый блокнот. Этот блокнот составлен из отрывных страниц, на каждой из которых напечатана таблица со случайными числами (ключами) K_I. Блокнот выполняется в двух экземплярах: один используется отправителем, а другой - получателем. Для каждого символа X, сообщения используется свой ключ K_I из таблицы только один раз. После того как таблица использована, она должна быть удалена из блокнота и уничтожена. Шифрование нового сообщения начинается с новой страницы.

Этот шифр абсолютно надежен, если набор ключей K_I, действительно случаен и непредсказуем. Если криптоаналитик попытается использовать для заданного шифртекста все возможные наборы ключей и восстановить все возможные варианты исходного текста, то они все окажутся равновероятными. Не существует способа выбрать исходный текст, который был действительно послан. Теоретически доказано, что одноразовые системы являются нераскрываемыми системами, поскольку их шифртекст не содержит достаточной информации для восстановления открытого текста.

Казалось бы, что благодаря данному достоинству одноразовые системы следует применять во всех случаях, требующих абсолютной информационной безопасности. Однако возможности применения одноразовой системы ограничены чисто практическими аспектами.

Существенным моментом является требование одноразового использования случайной ключевой последовательности. Ключевая последовательность с длиной, не меньшей длины сообщения, должна передаваться получателю сообщения заранее или отдельно по некоторому секретному каналу. Это требование не будет слишком обременительным для передачи действительно важных одноразовых сообщений, например, по горячей линии Вашингтон - Москва. Однако такое требование практически неосуществимо для современных систем обработки информации, где требуется шифровать многие миллионы символов.

В некоторых вариантах одноразового блокнота прибегают к более простому управлению ключевой последовательностью, но это приводит к некоторому снижению надежности шифра. Например, ключ определяется указанием места в книге, известной отправителю и получателю сообщения. Ключевая последовательность начинается с указанного места этой книги и используется таким же образом, как в системе Вижинера:

При шифровании исходного сообщения его выписывают в строку, а под ним записывают ключевое слово (фразу). Если ключ оказался короче сообщения, тот его циклически повторяют. В процессе шифрования символ исходного текста заменяют на некоторый другой символ, отстоящий от него на количество букв, соответствующее символу ключа.

Иногда такой шифр называют шифром с бегущим ключом. Управление ключевой последовательностью в таком варианте шифра намного проще, так как длинная ключевая последовательность может быть представлена в компактной форме. Но с другой стороны, эти ключи не будут случайными. Поэтому у криптоалитика появляется возможность использовать информацию о частотах букв исходного естественного языка.

Описание последовательности действий для алгоритма одноразового шифрования.

Для шифрования некоторого сообщения $A_0,\ldots,An-1$ составленного из букв алфавита $A=\{a_0,...,a_{m-1}\}$ необходимо:

Построить схему замещения букв алфавита целыми числами от 1 до m $A \rightarrow X$, позволяющую осуществлять обратное преобразование $X \rightarrow A$. (Например, $a_0 - 1$, $a_1 - 2,\ldots, a_{m-1} - m$)

Преобразовать исходный текст в множество $X=(X_0,\ldots,Xn-1)$

Получить последовательность $K= (X_0,\ldots,Xn-1)$ случайных равномерно распределенных чисел от 0 до m-1 ($0<=K<=m-1$).

Выполнить подстановку Цезаря $Y_i=(X_i + K_i) \bmod m$

Для полученной последовательности $Y=(Y_0,\ldots,Yn-1)$ произвести обратное преобразование из числового представления Y в буквы алфавита $A : Y \rightarrow B$.

Полученная последовательность $B=(B_0,\ldots,Bn-1)$ будет являться шифротекстом, а последовательность $K=(K_0,\ldots,Kn-1)$ – ключом.

Для расшифрования сообщения $B_0,\ldots,Bn-1$ необходимо произвести действия в обратном порядке, заменив подстановку Цезаря на формулу $X_i=(Y_i - K_i) \bmod m$.

Контрольные вопросы:

1. Что является характерной особенностью симметричных систем?
2. Что такое ключ?
3. В чем заключается суть шифрования заменой(подстановкой)?
4. Что позволяет упростить выполнение необходимых алгебраических действий в процессе шифрования?
5. Чем отличаются шифры сложной замены от простых шифров подстановки?
6. Почему данная система шифрования называется одноразовой?
7. На основе каких преобразований осуществляется кодирование?
8. Каковы достоинства и недостатки одноразовой системы шифрования?
9. Можно ли расшифровать закодированное сообщение, не зная ключа ? Если да, то в каких случаях?
10. Какой из случаев хуже с точки зрения дешифровки: при передаче был искажен i-ый символ
    а) закодированного сообщения
    б) ключа ?

11. Целесообразно ли использование одноразовой системы шифрования в современных условиях? Ответ аргументировать.

Задание

Реализовать шифрование/дешифрование любого исходного текста указанным методом.