Задание №2

Шифрование с помощью датчика псевдослучайных чисел

Принцип шифрования заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел (ПСЧ) и наложении полученной гаммы на открытые данные обратимым образом (например, при использовании логической операции «исключающее ИЛИ»).

Процесс дешифрования данных сводится к повторной генерации гаммы шифра при известном ключе и наложению такой гаммы на зашифрованные данные. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, когда гамма шифра не содержит повторяющихся битовых последовательностей. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого слова. Фактически если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Чтобы получить линейные последовательности элементов гаммы, длина которых превышает размер шифруемых данных, используются датчики ПСЧ. На основе теории групп было разработано несколько типов таких датчиков.

В настоящее время наиболее доступными и эффективными являются конгруэнтные генераторы ПСЧ. Для этого класса генераторов ПСЧ можно сделать математически строгое заключение о том, какими свойствами обладают выходные сигналы этих генераторов с точки зрения периодичности и случайности.

Одним из хороших конгруэнтных генераторов является линейный конгруэнтный датчик ПСЧ. Он вырабатывает последовательности псевдослучайных чисел T(i), описываемые соотношением

$$T(i+1) = (A*T(i)+C) \bmod M,$$

где A и C — константы, T(0) — исходная величина, выбранная в качестве порождающего числа.

Такой датчик ПСЧ генерирует псевдослучайные числа с определенным периодом повторения, зависящим от выбранных значений A и C. Значение M обычно устанавливается равным 2^b, где b - длина слова компьютера в битах**.** Датчик имеет максимальный период M до того, как генерируемая последовательность чисел начнет повторяться. По причине, отмеченной ранее, необходимо выбирать числа *A* и C таким образом, чтобы период M был максимальным. Как показано в Д. Кнут Искусство программирования для ЭВМ.— М.: Мир, 1976. — Т.2, линейный конгруэнтный датчик ПСЧ имее у т максимальную длину M тогда и только тогда, когда C — нечетное, и (A)mod4=1.

Выше было сказано, что при определенных условиях криптостойкость растет с увеличением размера ключа. Для шифрования данных с помощью датчика ПСЧ может быть выбран ключ любого размера. Например, пусть ключ состоит из набора чисел X(j) размерностью b, где /=1,2, ..., N. Тогда создаваемую гамму шифра G можно представить как объединение непересекающихся множеств H(j):

$$G = H(1) \cup H(2) \cup \ldots \cup H(N),$$

где H(j*)* — множество соответствующих j-му сегменту данных и полученных на основе порождающего числа Y(j*),* определенного как функция от X(j) (например,

ПСЧ, полученное на основе X(j)).

Разумеется, возможны и другие, более изощренные варианты выбора порождающих чисел для гаммы шифра. Более того, гамму шифра необязательно рассматривать как объединение непересекающихся множеств. Например, гамма шифра может быть представлена в следующем виде:

$$G = L(1) (+) L(2) (+) ... (+) L(.N).$$

Здесь символ (+) обозначает операцию «Исключающее ИЛИ», а множества L(j), для каждого из которых мощность равна мощности гаммы, представляют собой объединение следующих множеств:

$$L(J) = V(j) \cup H(j) \cup W(j),$$

где V(j) и W(j) - множества нулей, H(j) - множество ПСЧ, соответствующих j-сегменту данных. Причем мощности всех трех множеств выбраны на основе ключа, исходя из того, что мощность L(j) равна мощности G.

Пример простейшей программы шифрования области памяти методом гаммирования с использованием датчика псевдослучайных чисел приведен в приложении

Шифрование с помощью датчика ПСЧ является довольно распространенным криптографическим методом. Во многом качество шифра, построенного на основе датчика ПСЧ, определяется не только и не столько характеристиками датчика, сколько алгоритмом получения гаммы. Один из фундаментальных принципов криптологической практики гласит: даже очень грозно выглядящие шифры могут быть чувствительны к простым воздействиям. Кроме этого, шифры могут быть легко раскрыты, когда не применяются меры предосторожности. В качестве иллюстрации данного принципа рассмотрим проблему известного исходного текста.

Перспективный с практической точки зрения шаг на пути раскрытия любого зашифрованного файла — получить часть некоторого исходного текста и соответствующую ему часть зашифрованного. Общеизвестно, что стандартная информация, например, гриф «СОВ. СЕКРЕТНО» часто является уязвимой. Предположим, возможность добавлять записи к файлу и проверять зашифрованный файл, до и после добавления известной записи. Если гамма шифра представляет собой последовательность псевдослучайных чисел, каждое из которых может быть сгенерировано из предыдущего, то весь исходный текст можно легко восстановить из зашифрованного текста. Рассмотрим последовательность P = p(1),..., p(n) из n исходных слов в файле, к которым после y-го слова, $1<y<n$, добавляется новый элемент текста, содержащий w слов. В результате получается обновленный текст

$$p' = p'(1),..., p'(n+w).$$

Очевидно, что

$$p'(i) = p'(i), \quad i = 1, 2, ..., y,$$

$$p(i) = p'(i+w),..., \quad i = y+1, y+2, ..., n.$$

Здесь p'(y+l), ... ,p'(y+w) являются известными словами исходного текста.

Пусть G = g(I), g(2) ...— последовательность слов гаммы шифра, используемых для шифрования как P, так и P′. Тогда зашифрованные тексты для P и P′ можно представить в виде

$$C = c'(1), ... ,c'(n),$$

где c(i) = p(i) (+)gf(i), i= 1, 2, ... л;

C′ = c'(1),..., *c′*(п*),*

где c'(i) = p'(i) (+) g'(i), i = 1, 2, ..., n+w.

Теперь можно вычислить слово гаммы, которое использовалось для закрытия известного исходного текста:

g(y+1)= p'(y+i) (+) c'(y+i), i = 1 ,2 ,..., w.

Но эти слова гаммы использовались для шифрования Р. Следовательно,

p(y+1)= g(y+i*)* (+) c(y+i), *i* = 1, 2,..., w.

Видно, что дешифрование можно повторить, подставив y+w вместо *y.* Таким образом, все сегменты текста после позиции у могут быть дешифрованы. Легкое дешифрование текста стало возможным в связи с тем, что алгоритм шифрования не зависит ни от длины шифруемого файла, ни от содержимого самого файла. Но более или менее серьезное усовершенствование алгоритма получения гаммы шифра приводит к существенному повышению криптостойкости. Хорошие результаты дает метод гаммирования с обратной связью, который заключается в том, что для получения сегмента гаммы используется контрольная сумма определенного участка шифруемых данных. Например, если рассматривать гамму шифра как объединение непересекающихся множеств H(j), то процесс шифрования данных можно представить следующими шагами:

- определение контрольной суммы участка данных, соответствующего сегменту гаммы H(1);
- генерация сегмента гаммы H(1) и наложение его на соответствующий участок шифруемых данных;
- генерация с учетом контрольной суммы уже зашифрованного участка данных следующего сегмента гаммы H(2) (обычно контрольная сумма используется в процессе генерации порождающего числа для очередного сегмента гаммы);
- подсчет контрольной суммы участка данных, соответствующего сегменту гаммы H(2), и наложение этого сегмента гаммы на соответствующий участок шифруемых данных и т. д.

Под контрольной суммой здесь понимается функция f(t(1*),* ..., t(n)), где t(i) - i-е слово шифруемых данных. Разумеется, метод гаммирования с обратной связью может быть реализован с помощью другого алгоритма. Здесь изложены только общие принципы метода обратной связи (использование некоторых характеристик шифруемых данных для генерации гаммы).

ЗАДАНИЕ.

1. Разработать процедуру побайтного шифрования-дешифрования данных в оперативной памяти компьютера.
2. Исследовать характеристики датчика ПСЧ.
3. Провести анализ работы разработанной программы.

Приложение

ПРОГРАММА ШИФРОВАНИЯ. ОБЛАСТИ ПАМЯТИ С ПОМОЩЬЮ ПРОГРАММНОГО ДАТЧИКА ПСЧ

Функции.

```
Область памяти, указанная вызывающей программой, шифруется
```

по специальному алгоритму (с использованием датчика ПСЧ). Поскольку алгоритм шифрования является обратимым (содержимое области памяти складывается с гаммой шифра по команде "Исключающее ИЛИ"), дешифрование закодированной области памяти заключается в его повторном шифровании (повторном подключении данной процедуры с теми же параметрами).

ВХОДНЫЕ ПАРАМЕТРЫ.

Все параметры шифрования и характеристики кодируемой области памяти передаются на следующих регистрах: регистры ES:DI указывают на кодируемую область, регистр CX содержит длину этой области; регистр BP содержит коэффициент A для получения очередного ПСЧ, используемого для шифрования области, регистр SI содержит коэффициент C, регистр BX -начальное значение для запуска датчика ПСЧ.

ВЫХОДНЫЕ ПАРАМЕТРЫ.

После выполнения процедуры регистры ES:DI будут указывать на байт, следующий за обработанной областью, регистр CX будет обнулен, а в регистр BX загрухено ПСЧ. Остальные регистры не изменяются.

```
        push ax; сохранить рабочие  регистры.
        push dx
cme_l:  mov ax,bx; вычислить очередной  коэффициент
        случайного числа.
        mul Ьp
        add ax, si
        mov bx,ax; сохранить очередной коэффициент в регистре
        BX.
        xor BYTE PTR es:[di],a1 ; закодировать один байт
        области.
        inc di; перейти к следующему байту кодируемой области.
        loop cme_l
        pop dx; восстановить рабочие регистры.
        pop ax
        ret
```