Лабораторная работа № 2

Блочные составные шифры. Сеть Фейстеля

Цель работы

Ознакомиться с блочными составными шифрами, освоить криптографические преобразования подстановки и перестановки. Изучить и реализовать шифрование информации при помощи сети Фейстеля.

Блочные составные шифры

Блочный шифр – это криптографическая система, которая делит открытый текст на отдельные блоки, как правило, одинакового размера и независимо оперирует с каждым из них с целью получения последовательности блоков шифрованного текста.

Шеннон предложил рассматривать блочные шифры как наиболее эффективное перспективное средство обеспечения конфиденциальности сообщений в системах секретной связи [1].

В общем случае детерминированный шифр G определяется следующим образом:

$$G = (P,\ C,\ K,\ F),$$

где P – множество входных значений, C – множество выходных значений, K – пространство ключей, F – функция шифрования, т.е. $F\colon\ P \times K \rightarrow C$.

Пусть составной шифр определяется семейством преобразований G_i, имеющим общие пространства входных и выходных значений, т.е. $P_i = C_i = M$, при этом результат действия функции F_i зависит от ключевого элемента $k_i \in K_i$. На основе этого семейства с помощью операции композиции можно построить шифр, задаваемый отображением

$$F\colon\ M \times (K_1 \times K_2 \times \ldots \times K_r) \rightarrow M,$$

причём

$$F = F_r \cdot \ \ldots \ \cdot F_2 \cdot F_1,$$

а ключом является вектор

$$(k_1,\ k_2,\ \ldots,\ k_r) \in K_1 \times K_2 \times \ldots \times K_r.$$

Преобразование F_i называется ***i-м раундом шифрования***, ключ k_i – ***раундовым ключом***. В некоторых случаях раундовые ключи получаются из ключа всей системы с помощью алгоритма выработки раундовых ключей (при этом размер ключа системы существенно меньше суммарного размера всех раундовых ключей). Если ключевые пространства K_i и преобразования F_i для всех раундов совпадают, то такой составной шифр называется ***итерационным***, представляющим собой композицию одной и той же криптографической функции, используемой с разными ключами.

Идея, лежащая в основе составных (или композиционных) блочных шифров, состоит в построении криптостойкой системы посредством многократного применения относительно простых криптографических преобразований, в качестве которых К. Шеннон предложил использовать преобразования *подстановки* (substitution) и *перестановки* (permutation). Схемы, реализующие эти преобразования, называются ***SP-сетями***.

Многократное использование этих преобразований (см. рисунок 1) позволяет обеспечить два свойства, которые должны быть присущи стойким шифрам: *диффузия* (diffusion) и *конфузия* (confusion) (см. рисунок 2). ***Диффузия*** предполагает распространение влияния одного знака открытого текста на значительное количество знаков шифротекста. Наличие у шифра этого свойства:

- позволяет скрыть статистическую зависимость между знаками открытого текста, иначе говоря, перераспределить избыточность исходного языка посредством распространения её на весь текст;
- не позволяет восстанавливать неизвестный ключ по частям.

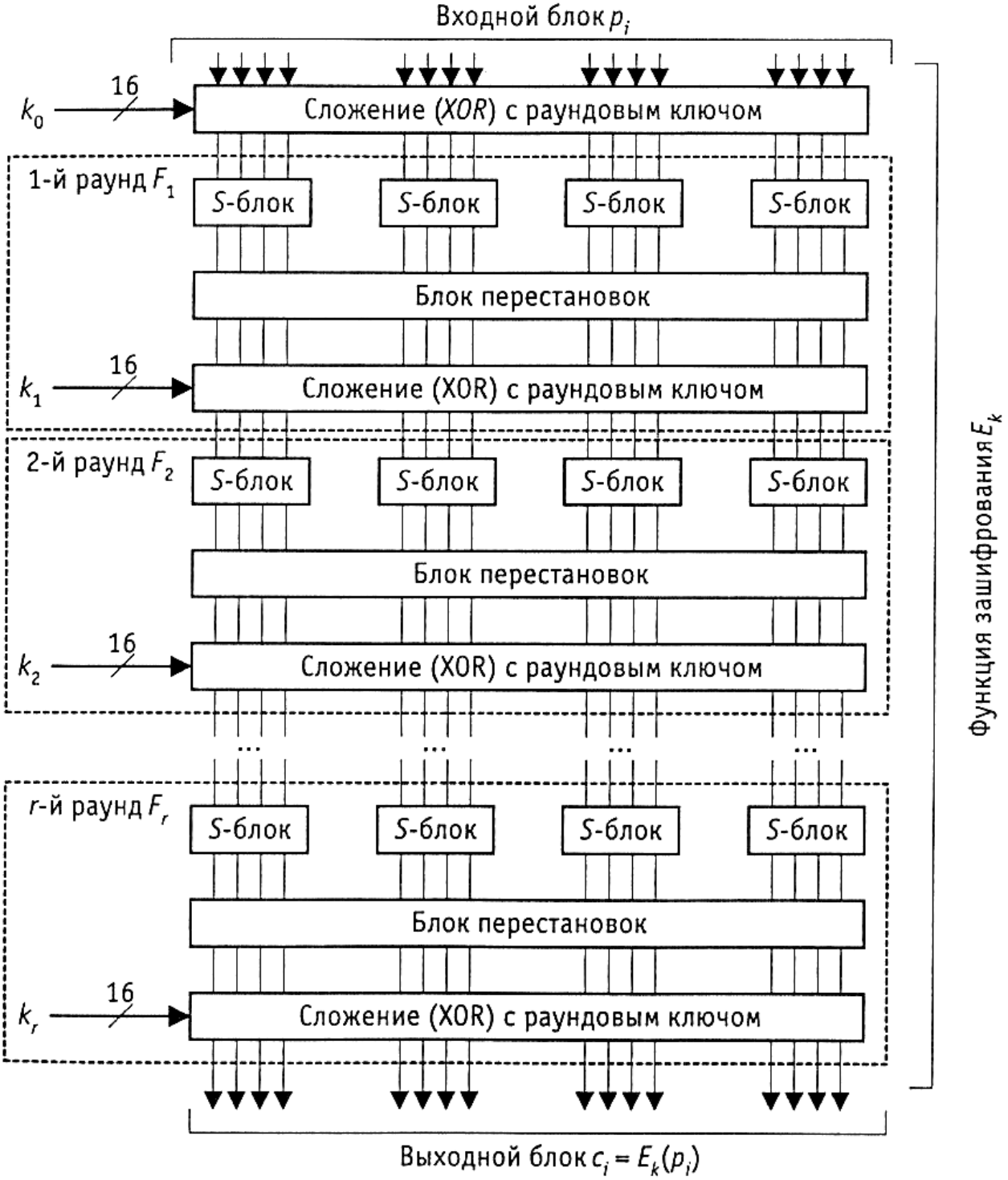

Рисунок 1 – Схема простейшего итерационного шифра

Цель ***конфузии*** – сделать как можно более сложной зависимость между ключом и шифротекстом. Криптоаналитик на основе статистического анализа перемешанного текста не должен получить сколько-нибудь значительного количества информации об использованном ключе.

Применение диффузии и конфузии порознь не обеспечивает необходимую стойкость, надёжная криптосистема получается только в результате их совместного использования.

Лавинный эффект (avalanche) – это число символов, которое изменилось в зашифрованном тексте при изменении одного бита открытого текста или ключа. Чем больше лавинный эффект, тем выше надёжность шифра.

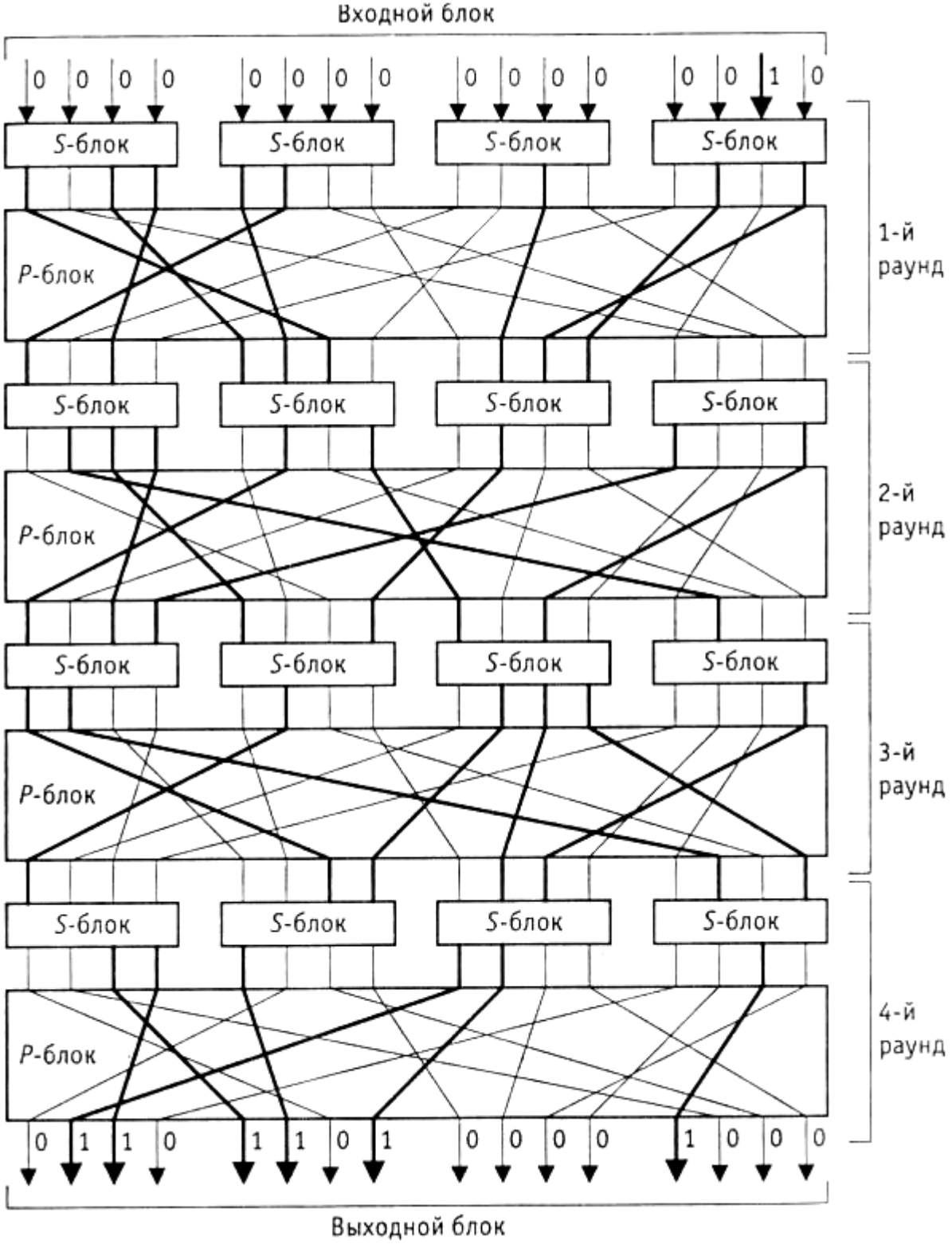

Рисунок 2 – Диффузия и конфузия в SP-сети

Сеть Фейстеля

Наиболее удачным криптографом, первым предложившим элегантный и практичный способ организации свойства рассеивания, оказался Хорст Фейстель, работавший с Шенноном на протяжении длительного времени. Архитектура нового способа шифрования впоследствии была названа в классической литературе *архитектурой Фейстеля* (на данный момент существует более устоявшийся термин: *сеть Фейстеля* или *Feistel's network*) [1].

Сетью Фейстеля называется метод обратимых преобразований текста, при котором значение, вычисленное от одной из частей текста, накладывается на другие части. Часто структура сети выполняется таким образом, что для шифрования и дешифрования используется один и тот же алгоритм – различие состоит только в порядке использования материала ключа.

Независимые потоки информации, порождённые из исходного блока, называются ***ветвями сети***. Величины V_i именуются ***параметрами сети***, обычно это функции от материала ключа. Функция F называется ***образующей***. Действие, состоящее из однократного вычисления образующей функции и последующего наложения её результата на другую ветвь с обменом их местами, называется ***циклом*** или ***раундом сети Фейстеля***.

На рисунке 3 показана структура шифра, предложенного Фейстелем. На вход алгоритма шифрования подаются блок открытого текста длиной 64 бита и ключ V. Блок открытого текста разделяется на две равные части X_1 и X_2, которые последовательно проходят через 16 раундов обработки, а затем объединяются снова для получения блока шифрованного текста соответствующей длины. Для раунда i в качестве входных данных выступают X_1^i и X_2^i, полученные на выходе предыдущего раунда, и подключ V_i, вычисляемый по общему ключу V. Как правило, все подключи V_i отличаются как от общего ключа V, так и друг от друга.

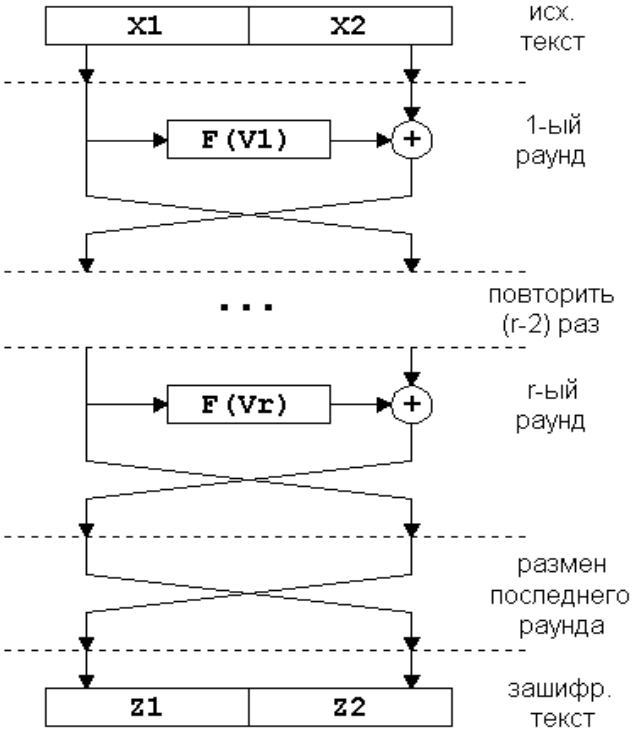

Рисунок 3 – Классическая сеть Фейстеля

Все раунды обработки проходят по одной и той же схеме. Сначала для правой половины блока данных выполняется операция подстановки. Она заключается в применении к левой половине блока данных некоторой функции раунда F и последующем сложении полученного результата с правой половиной блока данных с помощью операции XOR. Для всех раундов функция раунда имеет одну и ту же структуру, но зависит от параметра – подключа раунда V_i. После подстановки выполняется перестановка, представляющая собой обмен местами двух половин блока данных.

Процесс дешифрования шифра Фейстеля принципиально не отличается от процесса шифрования. Применяется тот же алгоритм, но на вход подаётся шифрованный текст, а подключи V_1 используются в обратной последовательности: для первого раунда берётся подключ V_n, для второго – V_{n-1}, и так далее до тех пор, пока не будет введён ключ V_1 для последнего раунда. Это свойство данной схемы шифрования оказывается очень удобным, т.к. для дешифрования не требуется вводить второй алгоритм, отличный от алгоритма шифрования.

Сеть Фейстеля надёжно зарекомендовала себя как криптостойкая схема проведения криптопреобразований, и её можно найти практически в любом современном блочном шифре.

Задание

I. Реализовать приложение для шифрования, позволяющее выполнять следующие действия:
   1. Шифровать данные при помощи сети Фейстеля:
      1) шифруемый текст должен храниться в одном файле, а ключ шифрования – в другом;
      2) приложение должно позволять выбирать способ получения подключей из заданного ключа шифрования:

а) для i-го раунда подключом V_i является цепочка из 32 подряд идущих бит заданного ключа, которая начинается с бита номер i, продолжается до последнего бита ключа и при его достижении циклически повторяется, начиная с 1 бита;
б) для i-го раунда, начиная с бита номер i, берётся цепочка из 8 подряд идущих бит ключа, которая является начальным значением для скремблера вида $0000\ 0011_2$; подключом V_i является сгенерированная этим скремблером последовательность из 32 бит;

3) приложение должно позволять выбирать вид образующей функции:
а) функция F – единичная, т.е. $F(V_i) = V_i$;
б) функция имеет вид $F(V_i, X) = S(X) \oplus V_i$, где $S(X)$ – левая часть шифруемого блока, на которую посредством операции XOR была наложена 32-битная последовательность, сгенерированная 16 разрядным скремблером вида $0100\ 0000\ 0000\ 0011_2$;
4) зашифрованный текст должен сохраняться в файл;
5) в процессе шифрования предусмотреть возможность просмотра и изменения ключа, шифруемого и зашифрованного текстов в шестнадцатеричном и символьном виде.

2. Исследовать лавинный эффект (исследования проводить на одном блоке текста):
1) для бита, который будет изменяться, приложение должно позволять задавать его позицию (номер) в открытом тексте или в ключе;
2) приложение должно уметь после каждого раунда шифрования подсчитывать число бит, изменившихся в зашифрованном тексте при изменении одного бита в открытом тексте либо в ключе;
3) приложение может строить графики зависимости числа бит, изменившихся в зашифрованном тексте, от раунда шифрования, либо графики можно строить в стороннем ПО, но тогда приложение для шифрования должно сохранять в файл необходимую для построения графиков информацию.

II. Реализовать приложение для дешифрования, позволяющее выполнять следующие действия:
1. Дешифровать данные при помощи сети Фейстеля:
1) зашифрованный текст должен храниться в одном файле, ключ – в другом;
2) приложение должно позволять выбирать способ получения подключей из заданного ключа шифрования;
3) приложение должно позволять выбирать вид образующей функции;
4) расшифрованный текст должен сохраняться в файл;
5) в процессе дешифрования предусмотреть возможность просмотра и изменения ключа, зашифрованного и расшифрованного текстов в шестнадцатеричном и символьном виде.

III. С помощью реализованных приложений выполнить следующие задания:
1. Протестировать правильность работы разработанных приложений.
2. Исследовать лавинный эффект при изменении одного бита в открытом тексте и в ключе: построить графики зависимостей числа бит, изменившихся в зашифрованном сообщении, от раунда шифрования при всех возможных комбинациях способов выбора ключа и образующей функции (всего должно быть построено 8 графиков).
3. Сделать выводы о проделанной работе.

Дополнительные критерии оценивания качества работы

1. Наглядность приложений:
1 – приложения позволяют просматривать и изменять ключи, шифруемый и зашифрованный тексты во всех предусмотренных заданием представлениях;
0 – приложения позволяют просматривать ключи, шифруемый и зашифрованный тексты только в каком-то одном представлении;
л.р. не принимается – иначе.

2. Построение графиков:
   1 – программа сама строит графики лавинного эффекта;
   0 – программа только выгружает необходимые для построения графиков данные;
   л.р. не принимается – программа не строит графики и не выгружает данные.

Вопросы для защиты

1. В чём заключается идея составных блочных шифров?
2. Что такое SP-сеть?
3. В чем заключается диффузия и конфузия?
4. Дайте определение сети Фейстеля и лавинообразного эффекта.
5. Что называют раундом и ветвями сети Фейстеля?
6. В чём заключаются процессы шифрования и дешифрования с помощью сети Фейстеля?
7. Что будет, если увеличить или уменьшить число раундов сети Фейстеля?
8. Какой может быть образующая функция F?
9. Каким образом можно вычислять подключи V_i?

Список литературы

1. Столлингс, В. Криптография и защита сетей: принципы и практика : Пер. с англ. / В. Столлингс. – 2-е изд. – М. : Издательский дом "Вильямс", 2001. – 672 с.