1. Аннотация

Алгоритм BlowFish:

BlowFish — алгоритм 64-битного блочного шифра с ключом переменной длины. Был разработан известным специалистом в области криптографии и защиты информации Брюсом Шнайером (Bruce Schneier) в 1993 году.

В общем случае алгоритм состоит из двух этапов — расширение ключа и шифрация/дешифрация исходных данных.

Отличительными особенностями этого алгоритма стала более высокая степень криптостойкости, нежели алгоритма DES (в том числе за счет использования переменной длины ключа, до 448 бит), высокая скорость шифрации/дешифрации (за счет генерации таблиц замены) и конечно — возможность его свободного применение для любых целей.

На этапе расширения ключа, исходный ключ преобразуется в матрицу раундовых ключей (P) и матрицу подстановки (или замены) (S, Substitution-box), общим объемом в 4168 байт. По всей вероятности, этим «расширением» (от 448 бит до 4168 байт) и объясняется выбор названия алгоритма BlowFish.

Шифрация данных, а также создания матрицы раундовых ключей и подстановки, происходит через использование сети Фейстеля, состоящей в свою очередь из 16 раундов.

Сеть Фейстеля:

В 1971 году, «крестный отец» стандарта DES, Хорст Фейстель (Horst Feistel), в стенах корпорации IBM, разработал два устройства, реализовавшие различные алгоритмы шифрования, названные затем общим название «Люцифер». В одной из этих устройств он использовал схему, которую впоследствии назвали Сетью Фейстеля (рис.1). Эта сеть представляет собой определённую многократно итерированную (повторяющуюся) структуру, которою называют ячейкой Фейстеля.

Принцип работы сети:

1. Исходные данные разбиваются на блоки фиксированной длины (как правило кратно степени двойки — 64 бит, 128 бит). В случае если длина блока исходных данных меньше длины разрядности шифра, то блок дополняется каким-либо заранее известным образом.

2. Блок делится на два равных подблока — «левый» L0 и «правый» R0. В случае 64-битной разрядности — на два блока с длиной 32 бита каждый.

3. «Левый подблок» L0 видоизменяется функцией итерации F(L0, P0) в зависимости от ключа P0, после чего он складывается по модулю 2 (XOR) с «правым подблоком» R0.

4. Результат сложения присваивается новому левому подблоку L1, который становится левой половиной входных данных для следующего раунда, а «левый подблок» L0 присваивается без изменений новому правому подблоку R1, который становится правой половиной.

5. Эта операция повторяется n-1 раз, при этом при переходе от одного этапа к другому меняются раундовые ключи (P0, P1, P2 и т.д.), где n — количество раундов для используемого алгоритма.

Процесс расшифрования аналогичен процессу шифрования за исключением того, что раундовые ключи используются в обратном порядке.

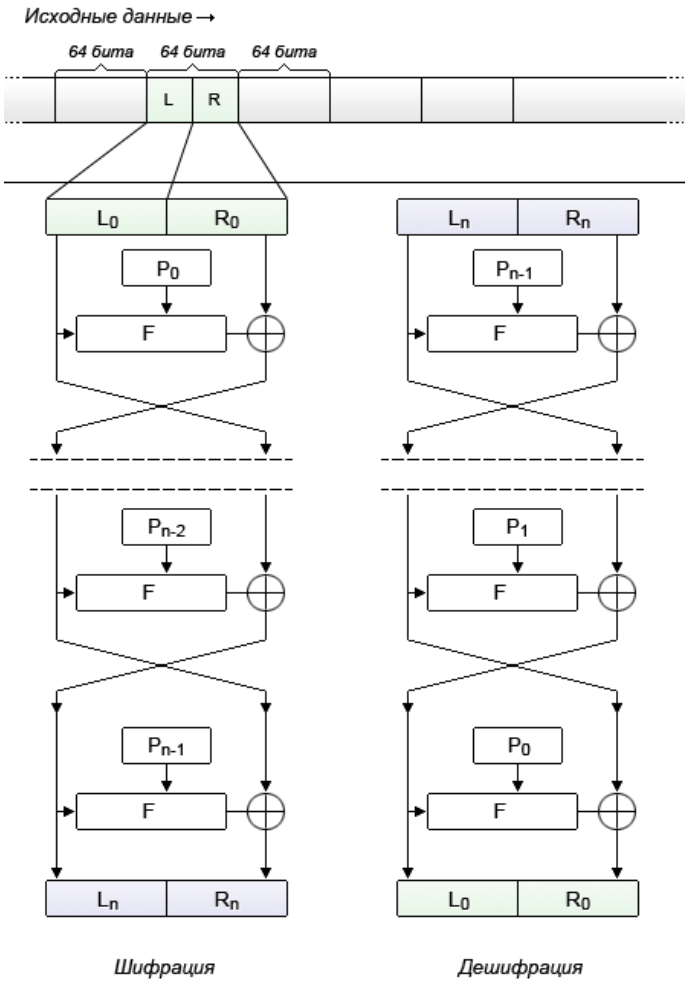

Рисунок 1. Сеть Фейстеля.

Алгоритм Blowfish:

В общем случае, алгоритм шифрования Blowfish представляет собой сеть Фейстеля, но с некоторыми особенностями генерации и использования раундовых ключей (P0, P1 …).

Для начала допустим, что функция итерации F в алгоритме Blowfish это некоторый «черный ящик», который принимает на входе и выдает на выходе 32-битное число (DWORD).

При этом 32-битные раундовые ключи Pn:

1. вычисляются по некоторому правилу от исходного ключа (длиной до 448 бит);
2. не являются аргументами для функции итерации F;

3. непосредственно складываются по модулю 2 (XOR) с «левым блоком». Результат этой операции является входящим 32-битным аргументом для функции F.

В алгоритме Blowfish при шифрации выполняется 16 раундов (внутри сети Фейстеля), а 17-й и 18-й ключи складываются с левым и правым выходным блоком последнего раунда. Такое количество раундов было выбрано, поскольку именно оно определяет длину возможного ключа. Однако, в учебных целях, наша реализация способна выполнять произвольное число раундов, заданных пользователем.

Если используется 18 раундовых ключей, каждый из которых имеет длину 32 бита, то в итоге мы получаем ключ длиной 576 бит (18 ключей × 32 бита). Длина исходного ключа в Blowfish изначально не ограничена 448 битами. Можно использовать ключи до 576 бит. Однако ограничение было сделано исходя из требований к соблюдению безопасности и криптостойкости алгоритма.

Расширение ключа (Blow it up!)

Подготовительным этапом алгоритма Blowfish является этап расширения ключей (рис.2). В процессе этого этапа строится матрица раундовых ключей Pn и матрица подстановки — 4 блока замены S-Box (Substitution-box), каждый из которых состоит из 256 32-х битных элементов.

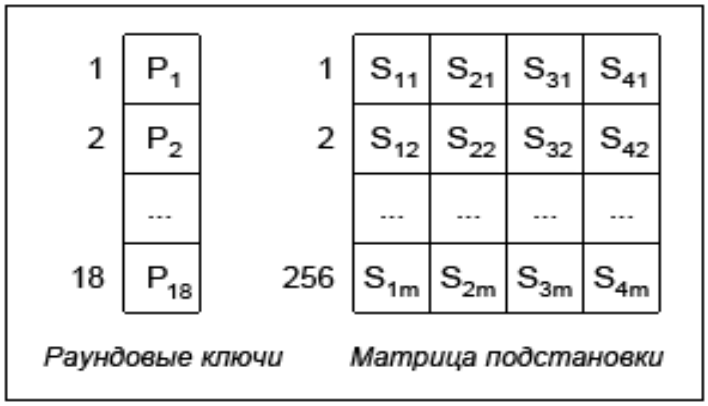

Рисунок 2. Расширение ключа.

Элементы этих матриц шифруются (вычисляются) с помощью рассмотренной выше сети Фейстеля для алгоритма Blowfish. Таким образом, сеть Фейстеля в алгоритме Blowfish используется:

- для шифрации/дешифрации исходных данных;
- для генерации матрицы раундовых ключей и матрицы подстановки (т.е. расширения ключа).

2. Manual (мануал)

При запуске программы открывается консоль и выводится приглашение: «Enter Ur information:». Далее пользователю необходимо ввести строку, которая будет зашифрована. После ввода информации на экране появляется следующее приглашение: «Enter the number of rounds:». Пользователь вводит количество циклов (раундов) шифрования в виде целого числа.

Результат выводится в консоль в следующем виде:

1. Введенная информация.
2. Зашифрованная информация
3. Дешифрованная информация.

3. Контрольный пример

На рис. 3 представлен пример шифрования информации, введенной пользователем (текст зеленого цвета). Количество циклов – 18, 16 раундов из которых выполнение шифрования внутри сети Фейстеля, а 17-й и 18-й ключи складываются с левым и правым выходным блоком последнего раунда. Результат выполнения: строка в 16-ричной системе счисления до шифрования, после шифрования и после расшифрования.

```
Enter Ur information: The weather is cool!
Enter the number of rounds: 18
[ 54 68 65 20 77 65 61 74 68 65 72 20 69 73 20 63 6F 6F 6C 21 ]
[ 2D A9 C6 36 88 4C 58 4 1 71 71 6E 25 54 99 E4 85 EB C0 A3 49 2 58 17 ]
[ 54 68 65 20 77 65 61 74 68 65 72 20 69 73 20 63 6F 6F 6C 21 0 0 0 0 ]
```

Рисунок 3. Контрольный пример при 18 циклах.

На рис. 4 представлено выполнение программы при изменении количества итераций до n=60.

```
Enter Ur information: The weather is cool!
Enter the number of rounds: 60
[ 54 68 65 20 77 65 61 74 68 65 72 20 69 73 20 63 6F 6F 6C 21 ]
[ C4 45 35 C5 FA 1A 64 B2 5B 4B 3C EB 4E 25 50 48 B7 DC B DA BE B3 1D 29 ]
[ 54 68 65 20 77 65 61 74 68 65 72 20 69 73 20 63 6F 6F 6C 21 0 0 0 0 ]
```

Рисунок 4. Контрольный пример при 60 циклах.

На рис. 5 представлено выполнение программы при изменении количества итераций до n=25. Исходная, зашифрованная и расшифрованная информация выводятся в консоль.

```
Enter Ur information: The weather is cool!
Enter the number of rounds: 25
[ 54 68 65 20 77 65 61 74 68 65 72 20 69 73 20 63 6F 6F 6C 21 ]
[ 87 34 C9 7F 8D EC 4A DA 77 88 AD 98 FB D8 4A A3 7F FF 3F D0 5E 2E B 91 ]
[ 54 68 65 20 77 65 61 74 68 65 72 20 69 73 20 63 6F 6F 6C 21 0 0 0 0 ]
```

Рисунок 5. Контрольный пример при 25 циклах.

4. Блок-схема алгоритма

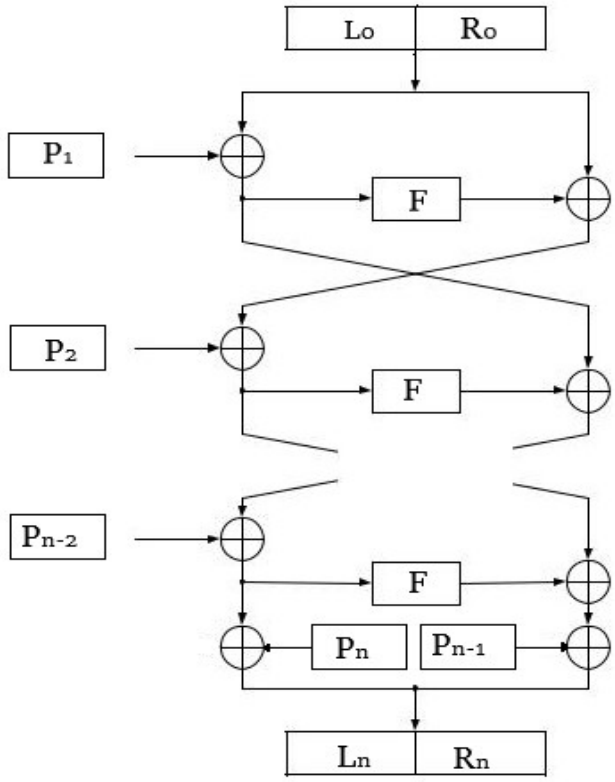

Рисунок 6. Блок-схема алгоритма BlowFish (с n - числом раундов).

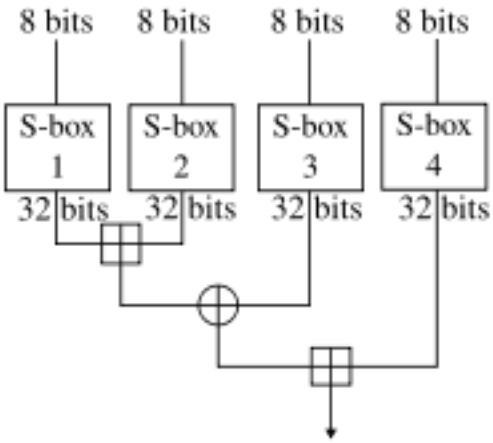

Рисунок 7. Блок-схема реализации функции F.

5. Вывод:

В ходе выполнения лабораторной работы рассмотрен алгоритм симметричного шифрования Blowish. Реализована программа на языке программирования C с возможностью изменения количества циклов-итераций.