Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ

РОССИЙСКОЙ ФЕДЕРАЦИИ»

(Финансовый университет)

**Колледж информатики и программирования**

Самостоятельная работа № 1 по дисциплине

МДК.02.02 «Криптографические средства и методы защиты информации»

На тему: «Программная реализация симметричного криптографического алгоритма Blowfish»

Студент группы 3ОИБАС*-№ группы*

|  |  |
| --- | --- |
| *Королев Тимофей Андреевич, Орлов Александр Султандекович, Дроздов Кирилл Николаевич* | «14» 02 2023 г. |

Основная профессиональная образовательная программа по специальности

10.02.05 Обеспечение информационной безопасности автоматизированных систем

Форма обучения очная

Проверили: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рой А.В.,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рой О.В.

Москва

2020

Оглавление

[Введение 3](#_Toc127276713)

[Основная часть 4](#_Toc127276714)

[Теоретические сведенья 4](#_Toc127276715)

[Программная реализация симметричного криптографического алгоритма Blowfish 4](#_Toc127276716)

[Выводы 9](#_Toc127276717)

[Список литературы 10](#_Toc127276718)

# Введение

# Основная часть

## Теоретические сведенья

## Программная реализация симметричного криптографического алгоритма Blowfish

Blowfish (произносится [бло́уфиш]) — криптографический алгоритм, реализующий блочное симметричное шифрование с переменной длиной ключа. Разработан Брюсом Шнайером в 1993 году. Представляет собой сеть Фейстеля[1][2]. Выполнен на простых и быстрых операциях: XOR, подстановка, сложение[2]. Является незапатентованным и свободно распространяемым.

До появления Blowfish существовавшие алгоритмы были либо запатентованными, либо ненадёжными, а некоторые и вовсе держались в секрете (например, Skipjack). Алгоритм был разработан в 1993 году Брюсом Шнайером в качестве быстрой и свободной альтернативы устаревшему DES и запатентованному IDEA. По заявлению автора, критериями проектирования Blowfish были[1][2]:

* скорость (шифрование на 32-битных процессорах происходит за 26 тактов);
* простота (за счёт использования простых операций, уменьшающих вероятность ошибки реализации алгоритма);
* компактность (возможность работать в менее, чем 5 Кбайт памяти);
* настраиваемая безопасность (изменяемая длина ключа).

Алгоритм состоит из двух частей: расширение ключа и шифрование данных. На этапе расширения ключа исходный ключ (длиной до 448 бит) преобразуется в 18 32-битовых подключей и в 4 32-битных S-блока, содержащих 256 элементов. Общий объём полученных ключей равен (18+256\*4)\*32=33344 бит или 4168 байт

Параметры

* секретный ключ K (от 32 до 448 бит)
* 32-битные ключи шифрования
* 32-битные таблицы замен :

Функция F(x) принимает на вход блок размером в 32 бита и проделывает с ним следующие операции:

1. 32-битный блок делится на четыре 8-битных блока, каждый из которых является индексом массива таблицы замен
2. значения и  складываются по модулю , после складываются по модулю  с  и, наконец, складываются с  по модулю .
3. Результат этих операций — значение .

Blowfish представляет собой Сеть Фейстеля, состоящую из 16 раундов. Алгоритм шифрования блока  размером 64 бит выглядит следующим образом[[1]](https://ru.wikipedia.org/wiki/Blowfish#cite_note-_917806ba3e43fd37-1)[[2]](https://ru.wikipedia.org/wiki/Blowfish#cite_note-_0fc697232ab5c38f-2):

1. Разделение входного блока  на 2 32-битных блока
2. Для
3. После 16 раунда  меняются местами:
4. К получившимся блокам прибавляются и
5. Выходной блок  равен конкатенации (объединению)  и .

Blowfish разделён на 2 этапа:

1. Подготовительный — формирование ключей шифрования по секретному ключу.
   * Инициализация массивов P и S при помощи секретного ключа K
     1. Инициализация  фиксированной строкой, состоящей из шестнадцатеричных цифр мантиссы числа пи Архивная копия от 3 сентября 2008 на Wayback Machine.
     2. Производится операция XOR над  с первыми 32 битами ключа , над  со вторыми 32-битами и так далее.  
        Если ключ короче, то он накладывается циклически.
   * Шифрование ключей и таблиц замен
     1. Алгоритм шифрования 64-битного блока, используя инициализированные ключи  и таблицу замен , шифрует 64 битную нулевую (0x0000000000000000) строку. Результат записывается в , .
     2. и  шифруются изменёнными значениями ключей и таблиц замен. Результат записывается в  и .
     3. Шифрование продолжается до изменения всех ключей  и таблиц замен .
2. Шифрование текста полученными ключами и F(x), с предварительным разбиением на блоки по 64 бита. Если невозможно разбить начальный текст точно на блоки по 64 бита, используются различные режимы шифрования для построения сообщения, состоящего из целого числа блоков. Суммарная требуемая память 4168 байт.

Дешифрование происходит аналогично, только применяются в обратном порядке.

Выбор начального значения P-массива и таблицы замен

Нет ничего особенного в цифрах числа пи. Данный выбор заключается в инициализации последовательности, не связанной с алгоритмом, которая могла бы быть сохранена как часть алгоритма или получена при необходимости (Пи (число)). Как указывает сам Шнайер: «Подойдёт любая строка из случайных битов — цифры числа e, RAND-таблицы или биты с выхода генератора случайных чисел.»

Криптостойкость

S-блоки называются **слабыми**, если существуют такие . Ключ, генерирующий **слабые** S-блоки, тоже называется **слабым**. Серж Воденэ указал на наличие небольшого класса слабых ключей (генерирующих слабые S-блоки). Вероятность появления слабого S-блока равна . Он также рассмотрел упрощенный вариант Blowfish, с известной функцией F(x) и слабым ключом. Для этого варианта требуется  выбранных открытых текстов (t — число раундов, а символы [ ] означают операцию получения целой части числа). Эта атака может быть использована только для алгоритма с . Для  требуется  открытых текстов, причём для варианта с известным F(x) и случайным ключом требуется  открытых текстов. Но данная атака неэффективна для Blowfish с 16 раундами ().

Невозможно заранее определить, является ли ключ слабым. Проводить проверку можно только после генерации ключа.

Криптостойкость можно настраивать за счёт изменения количества раундов шифрования (увеличивая длину массива P) и количества используемых S-блоков. При уменьшении используемых S-блоков возрастает вероятность появления слабых ключей, но уменьшается используемая память. Адаптируя Blowfish для 64-битной архитектуры, можно увеличить количество и размер S-блоков (а следовательно и память для массивов P и S), а также усложнить F(x), причём для алгоритма с такой функцией F(x) невозможны вышеуказанные атаки.

Модификация F(x): на вход подается 64-битный блок, который делится на восемь 8-битных блоков (X1-X8). Результат вычисляется по формуле , где  — это операция сложения по модулю

Использование Blowfish 64-битного блока (в отличие, например, от 128-битного блока AES) делает его уязвимым для атаки дней рождения, в частности, в контекстах типа HTTPS. В 2016 году атака SWEET32 продемонстрировала, как использовать атаку дней рождения для восстановления открытого текста (то есть расшифровки) из 64-битных блоков. Проект GnuPG рекомендует не использовать Blowfish для файлов с размером, превышащим 4 ГБ из-за малого размера блока.

Известно, что вариант Blowfish с уменьшенным количеством раундов является уязвимым для атак на основе открытых текстов на сравнительно слабых ключах. Реализации Blowfish с 16 раундами шифрования не подвержены подобным атакам.

# Выводы

# Список литературы