Алгоритм шифрования BlowFish

Арсентьев Владислав
Исхаков Анас
Мингазов Рамиль
Миндубаев Ильфат

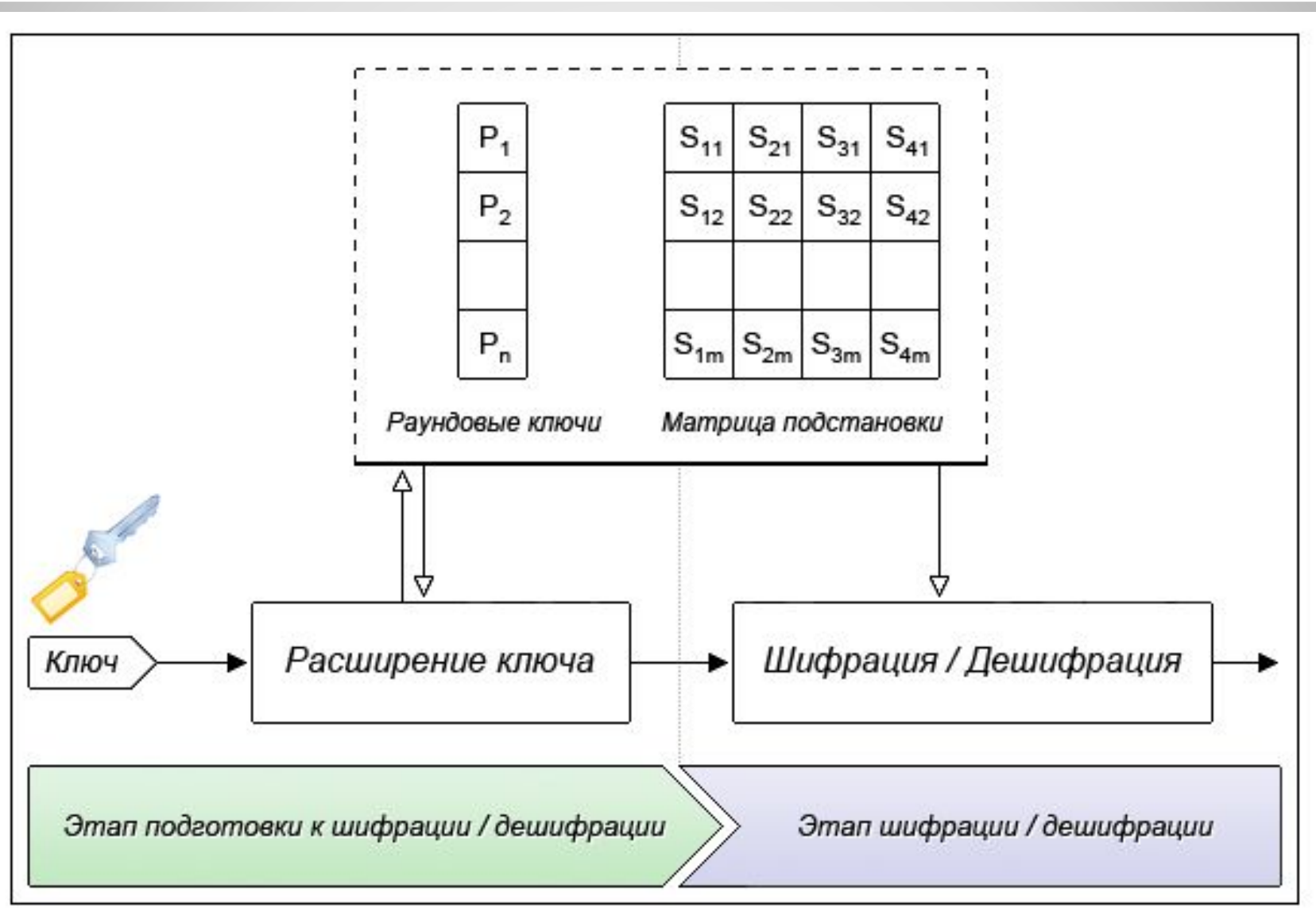
История создания

В конце 1993 года остро возникла необходимость создания криптостойкого ключа, без каких либо ограничений на право свободного использования.

В 1994 году Брюс Шнайер презентовал свой алгоритм блочного шифра, который был назван **Blowfish**.

Что это такое?

BlowFish — алгоритм 64-битного блочного шифра с ключом переменной длины. В общем случае алгоритм состоит из двух этапов — расширение ключа и шифрация/дешифрация исходных данных.

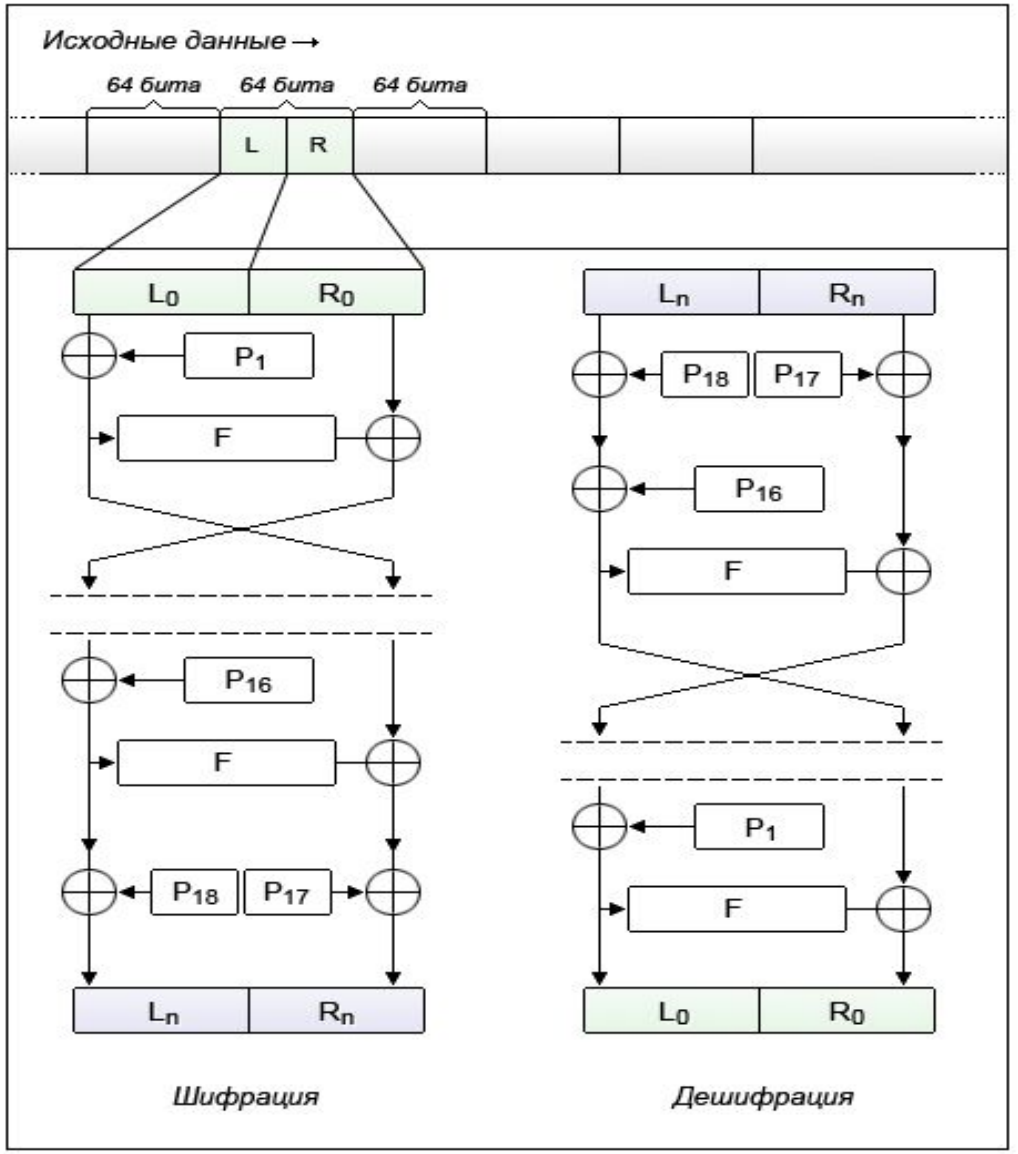

P1
P2
Pn
S11 S21 S31 S41
S12 S22 S32 S42
S1m S2m S3m S4m
Раундовые ключи
Матрица подстановки
Ключ
Расширение ключа
Шифрация / Дешифрация
Этап подготовки к шифрации / дешифрации
Этап шифрации / дешифрации

Как работает?

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

Сеть Фейстеля

1. Исходные данные разбиваются на блоки фиксированной длины.
2. Блок делится на два равных подблока.
3. Видоизменения блоков.
4. Повторение операции n-1 раз.



Алгоритм шифрования Blowfish

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.

Вопрос!

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). Почему же длина исходного ключа в Blowfish изначально ограничена 448 битами?

Правильный ответ:

Длина не ограничена. Можно использовать ключи до 576 бит. Но! Ограничение было сделано исходя из требований к соблюдению безопасности и криптостойкости алгоритма.

Этапы шифрования.

1. Выделяем массив из 18 элементов для раундовых ключей сети Фейстеля и 4 матриц подстановки по 256 элементов в каждой.

2. Заполняем выделенный массив значением мантиссы числа PI.

Этапы шифрования.

3. Делаем итеративный XOR: Pi = Pi XOR Ki
(где Pi — раундовый ключ, а Ki — исходный ключ).

4. Шифруем раундовые ключи и матрицы подстановки с помощью сети Фейстеля.

5. Шифруем/дешифруем блоки исходных данных по 64 бита также с помощью сети Фейстеля.

Достоинства:

- Высокая скорость шифрования на развернутом ключе;
- Простота алгоритма, снижающая вероятность ошибок при его реализации;
- Отсутствие успешных атак на полнораундовую версию алгоритма.

Вывод:

Процедура расширения ключа ресурсоемка, поэтому алгоритм шифрования Blowfish не подходит для применения в случаях, где требуется частая смена ключей. В связи с этим, одно из достоинств алгоритма - высокая скорость шифрования - проявляется только в тех случаях, если на одном ключе шифруется достаточно большой объем информации.

Спасибо!