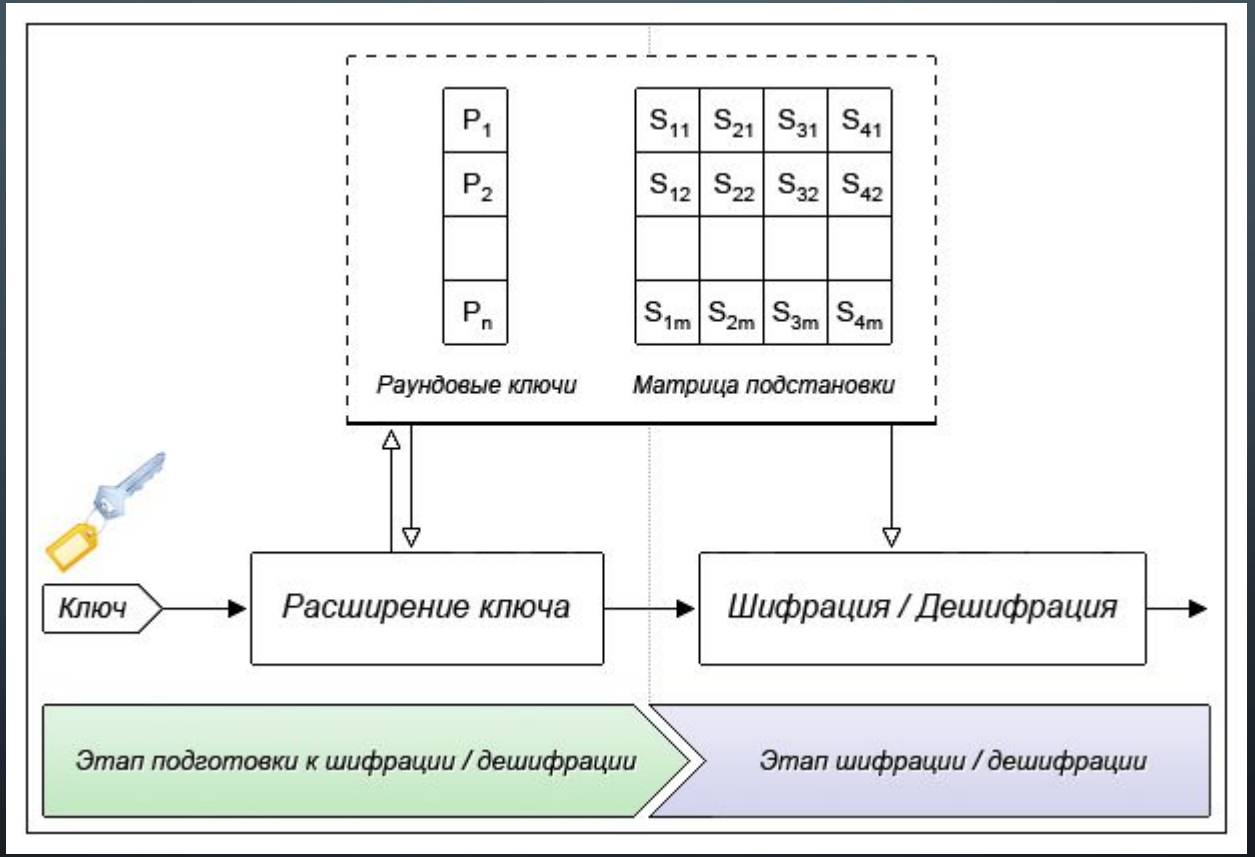
Алгоритм BlowFish

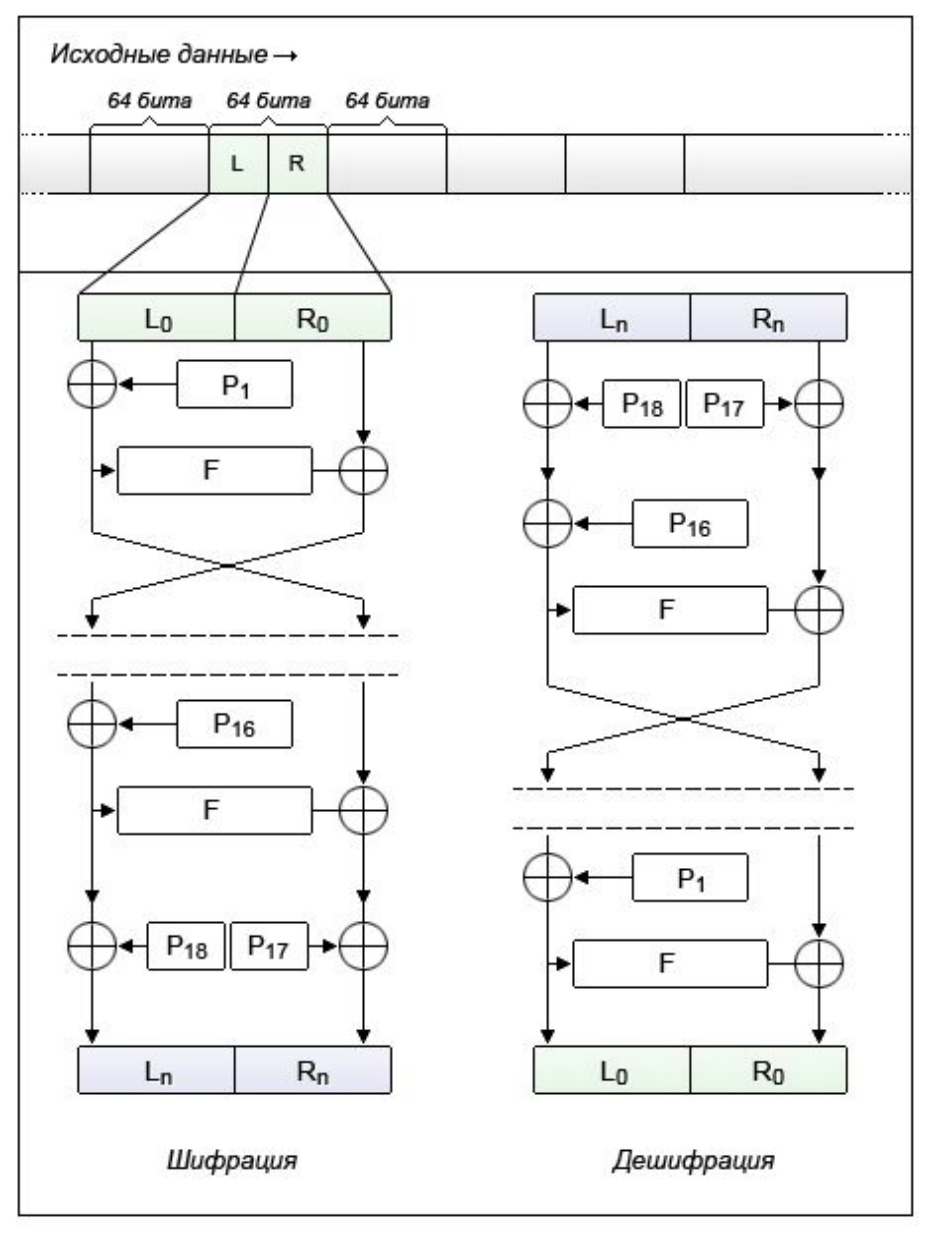
В 1994 году, на семинаре Fast Software Encryption в Кембридже, Брюс Шнайер презентовал свой алгоритм блочного шифра, который был назван Blowfish.

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.

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

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



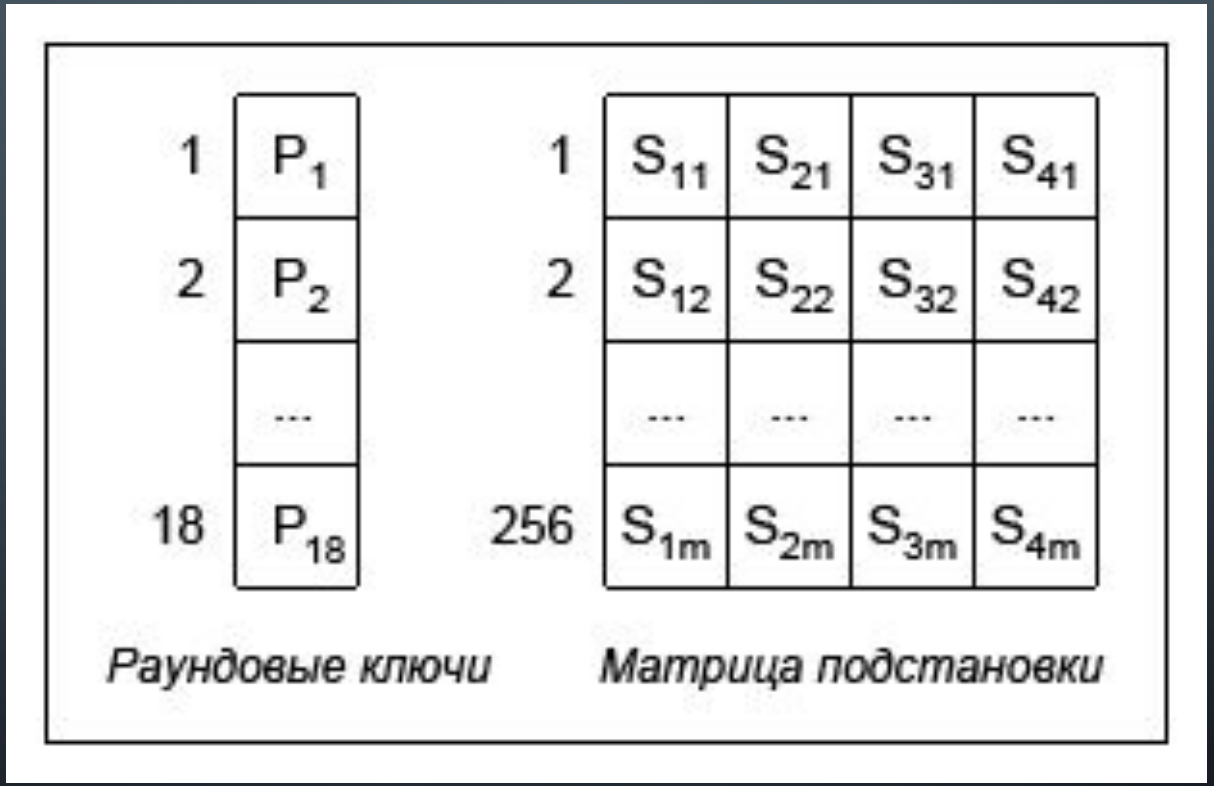


В общем случае, алгоритм шифрования Blowfish представляет собой сеть Фейстеля, но с некоторыми особенностями генерации и использования раундовых ключей.

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.

Расширение ключа

Подготовительным этапом алгоритма Blowfish является этап расширения ключей. 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.



Достоинства и недостатки алгоритма

В своей книге "Прикладная криптография" Брюс Шнайер отметил следующие ограничения алгоритма Blowfish. Во-первых, "...алгоритм Blowfish не годится для применения в случаях, где требуется частая смена ключей". Процедура расширения ключа ресурсоемка, поэтому одно из достоинств алгоритма Blowfish - высокая скорость шифрования - проявляется только в тех случаях, если на одном ключе шифруется достаточно большой объем информации.

Алгоритм Blowfish имеет и достаточно важные преимущества, в частности:

- высокая скорость шифрования на развернутом ключе (как уже упоминалось выше);
- простота алгоритма, снижающая вероятность ошибок при его реализации;
- отсутствие успешных атак на полнораундовую версию алгоритма.