3.5. Алгоритм Blowfish

Blowfish является сетью Фейштеля, у которой количество итераций равно 16. Длина блока равна 64 битам, ключ может иметь любую длину в пределах 448 бит. Хотя перед началом любого шифрования выполняется сложная фаза инициализации, само шифрование данных выполняется достаточно быстро.

Алгоритм предназначен в основном для приложений, в которых ключ меняется нечасто, к тому же существует фаза начального рукопожатия, во время которой происходит аутентификация сторон и согласование общих параметров и секретов. Классическим примером подобных приложений является сетевое взаимодействие. При реализации на 32-битных микропроцессорах с большим кэшем данных *Blowfish* значительно быстрее DES.

Алгоритм состоит из двух частей: расширение ключа и шифрование данных. Расширение ключа преобразует ключ длиной, по крайней мере, 448 бит в несколько массивов *подключей* общей длиной 4168 байт.

В основе алгоритма лежит сеть Фейштеля с 16 итерациями. Каждая итерация состоит из перестановки, зависящей от ключа, и подстановки, зависящей от ключа и данных. Операциями являются XOR и сложение 32-битных слов.

Blowfish использует большое количество *подключей*. Эти ключи должны быть вычислены заранее, до начала любого шифрования или дешифрования данных. Элементы алгоритма:

1. P - массив, состоящий из восемнадцати 32-битных *подключей*: P_1, P_2, ..., P_{18}.
2. Четыре 32-битных *S-boxes* с 256 входами каждый. Первый индекс означает номер *S-box*, второй индекс - номер входа.
3. $S_{1,0}$, $S_{1,1}$, … $S_{1,255}$;
4. $S_{2,0}$, $S_{2,1}$, … $S_{2,255}$;
5. $S_{3,0}$, $S_{3,1}$, … $S_{3,255}$;
6. $S_{4,0}$, $S_{4,1}$, … $S_{4,255}$;

Метод, используемый для вычисления этих *подключей*, будет описан ниже.

Шифрование

Входом является 64-битный элемент данных X, который делится на две 32-битные половины, X_l и X_r.

$X_l = X_l$ XOR P_i

$X_r = F(X_l)$ XOR X_r

Swap X_l and X_r

Функция F

Разделить X_l на четыре 8-битных элемента A, B, C, D.

$F(X_l) = ((S_{1,A} + S_{2,B} \bmod 2^{32})$ XOR $S_{3,C}) + S_{4,D} \bmod 2^{32}$

Дешифрование отличается от шифрования тем, что P_i используются в обратном порядке.

Генерация подключей

Подключи вычисляются с использованием самого *алгоритма Blowfish*.

1. Инициализировать первый P-массив и четыре *S-boxes* фиксированной строкой.
2. Выполнить операцию XOR P_1 с первыми 32 битами ключа, операцию XOR P_2 со вторыми 32 битами ключа и т.д. Повторять цикл до тех пор, пока весь P-массив не будет побитово сложен со всеми битами ключа. Для коротких ключей выполняется конкатенация ключа с самим собой.
3. Зашифровать нулевую строку *алгоритмом Blowfish*, используя *подключи*, описанные в пунктах (1) и (2).
4. Заменить P_1 и P_2 выходом, полученным на шаге (3).
5. Зашифровать выход шага (3), используя *алгоритм Blowfish* с модифицированными *подключами*.
6. Заменить P_3 и P_4 выходом, полученным на шаге (5).
7. Продолжить процесс, заменяя все элементы P-массива, а затем все четыре *S-boxes*, выходами соответствующим образом модифицированного *алгоритма Blowfish*.

   Для создания всех *подключей* требуется 521 итерация.