5.5. Хэш-функция MD5

Алгоритм MD5 получает на входе сообщение произвольной длины и создает в качестве выхода *дайджест сообщения* длиной 128 бит. Алгоритм состоит из следующих шагов:

Рисунок 5.1 – Логика выполнения MD5

Шаг 1: добавление недостающих битов

Сообщение дополняется таким образом, чтобы его длина стала равна 448 по модулю 512 (длина≡448 mod 512).

Шаг 2: добавление длины

64-битное представление длины исходного (до добавления) сообщения в битах присоединяется к результату первого шага. Если первоначальная длина больше, чем 2^{64}, то используются только последние 64 бита.

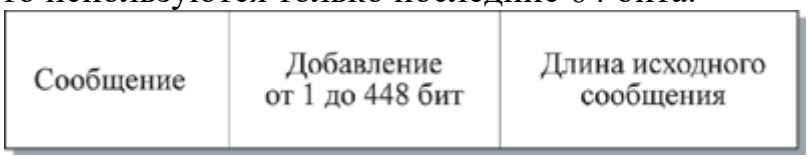

Рисунок 5.2 – Структура расширенного сообщения

Шаг 3: инициализация MD-буфера

Используется 128-битный буфер для хранения промежуточных и окончательных результатов *хэш-функции*. Буфер может быть представлен как четыре 32-битных регистра (A, B, C, D). Эти регистры инициализируются следующими шестнадцатеричными числами:

A = 01234567; B = 89ABCDEF; C = FEDCBA98; D = 76543210

Шаг 4: обработка последовательности 512-битных (16-словных) блоков

Основой алгоритма является модуль, состоящий из четырех циклических обработок, обозначенный как HMD5. Четыре цикла имеют похожую структуру, но каждый цикл использует свою элементарную логическую функцию, обозначаемую f_F, f_G, f_H и f_I соответственно.

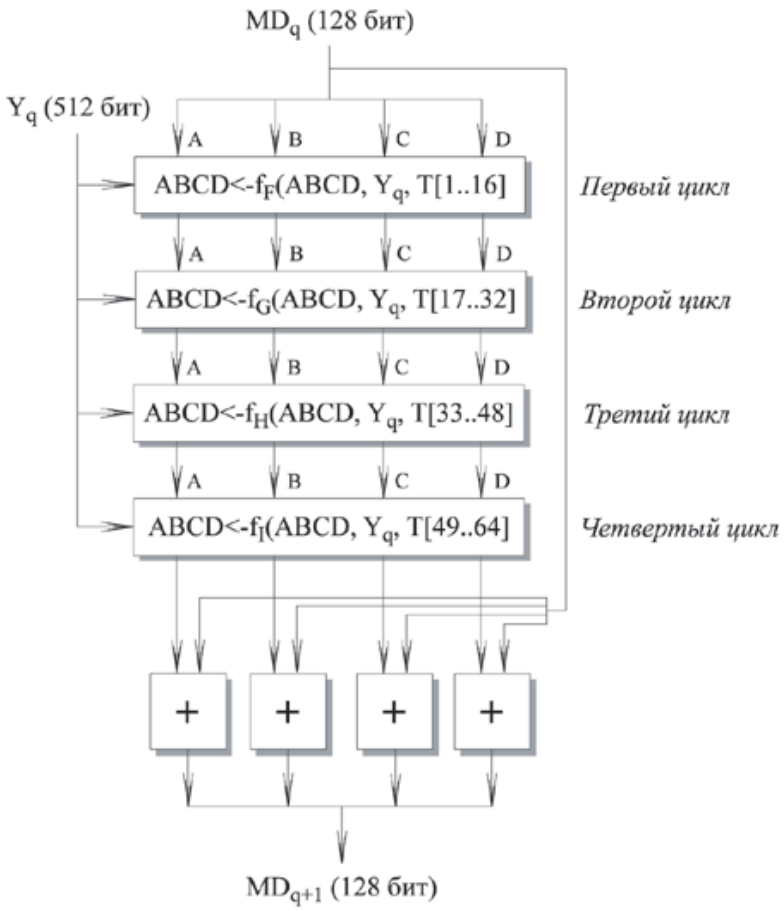

Рисунок 5.3 – Обработка очередного 512-битного блока

Каждый цикл принимает в качестве входа текущий 512-битный блок Y_q, обрабатывающийся в данный момент, и 128-битное значение буфера ABCD, которое является промежуточным значением *дайджеста*, и изменяет содержимое этого буфера. Каждый цикл также использует четвертую часть 64-элементной таблицы T[1 ... 64], построенной на основе функции sin. i-ый элемент T, обозначаемый T[i], имеет значение, равное целой части от 2^{32} * abs (sin (i)), i задано в радианах. Так как abs (sin (i)) является числом между 0 и 1, каждый элемент T является целым, которое может быть представлено 32 битами. Таблица обеспечивает "случайный" набор 32-битных значений, которые должны ликвидировать любую регулярность во входных данных.

Для получения MD_{q+1} выход четырех циклов складывается по модулю 2^{32} с MD_q. Сложение выполняется независимо для каждого из четырех слов в буфере.

Шаг 5: выход

После обработки всех *L* 512-битных блоков выходом *L*-ой стадии является 128-битный *дайджест сообщения*.