5.7. Хэш-функция SHA-1

Алгоритм получает на входе сообщение максимальной длины 2^{64} бит и создает в качестве выхода дайджест сообщения длиной 160 бит.

Алгоритм состоит из следующих шагов:

Рисунок 5.4 – Логика выполнения SHA-1

Шаг 1: добавление недостающих битов

Сообщение добавляется таким образом, чтобы его длина была кратна 448 по модулю 512 (длина≡448 mod 512).

Шаг 2: добавление длины

К сообщению добавляется блок из 64 битов. Этот блок трактуется как беззнаковое 64-битное целое и содержит длину исходного сообщения до добавления.

Шаг 3: инициализация SHA-1 буфера

Используется 160-битный буфер для хранения промежуточных и окончательных результатов хэш-функции. Буфер может быть представлен как пять 32-битных регистров A, B, C, D и E. Эти регистры инициализируются следующими шестнадцатеричными числами:

A = 67452301; B = EFCDAB89; C = 98BADCFE; D = 10325476; E = C3D2E1F0

Шаг 4: обработка сообщения в 512-битных (16-словных) блоках

Основой алгоритма является модуль, состоящий из 80 циклических обработок, обозначенный как H_{SHA}. Все 80 циклических обработок имеют одинаковую структуру.

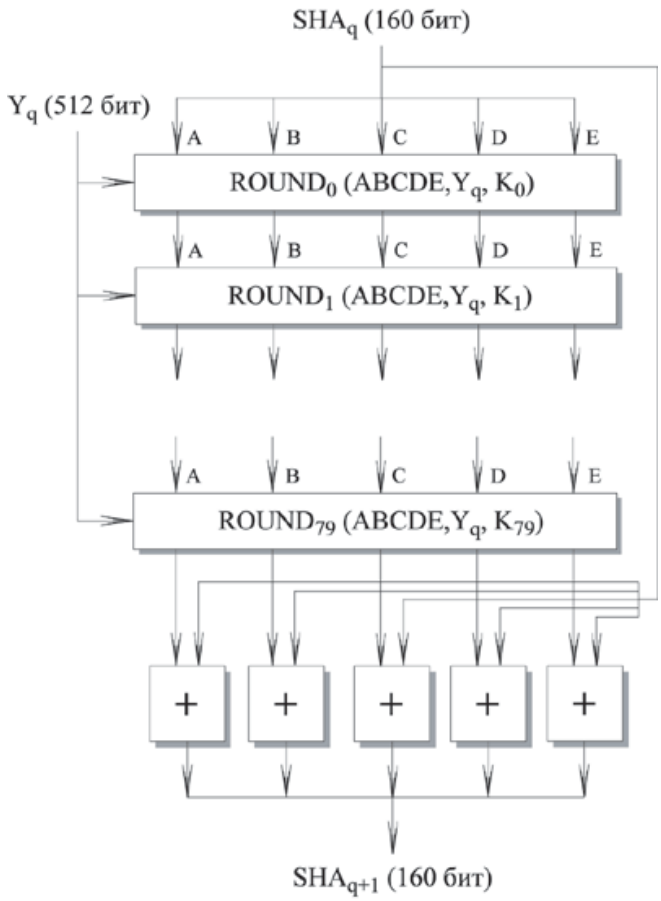

Рисунок 5.5 – Обработка очередного 512-битного блока

Каждый цикл получает на входе текущий 512-битный обрабатываемый блок Y_q и 160-битное значение буфера ABCDE, и изменяет содержимое этого буфера.

В каждом цикле используется дополнительная константа K_t, которая принимает только четыре различных значения:

$0 \le t \le 19$ K_t = 5A827999 (целая часть числа $[2^{30} \times 2^{1/2}]$)

$20 \le t \le 39$ K_t = 6ED9EBA1 (целая часть числа $[2^{30} \times 3^{1/2}]$)

$40 \le t \le 59$ K_t = 8F1BBCDC (целая часть числа $[2^{30} \times 5^{1/2}]$)

$60 \le t \le 79$ K^t = CA62C1D6 (целая часть числа $[2^{30} \times 10^{1/2}]$)

Для получения SHA_{q+1} выход 80-го цикла складывается со значением SHA_q. Сложение по модулю 2^{32} выполняется независимо для каждого из пяти слов в буфере с каждым из соответствующих слов в SHA^q.

Шаг 5: выход

После обработки всех 512-битных блоков выходом L-ой стадии является 160-битный дайджест сообщения.

Рассмотрим более детально логику в каждом из 80 циклов обработки одного 512-битного блока. Каждый цикл можно представить в виде:

A, B, C, D, E (CLS_5 (A) + f_t (B, C, D) + E + W_t + K_t), A, CLS_{30} (B), C, D

Где

A, B, C, D, E - пять слов из буфера.

t - номер цикла, $0 \le t \le 79$.

f_t – элементарная логическая функция.

CLS_s - циклический левый сдвиг 32-битного аргумента на s битов.

W_t - 32-битное слово, полученное из текущего входного 512-битного блока.

K_t - дополнительная константа.

+ - сложение по модулю 2^{32}.

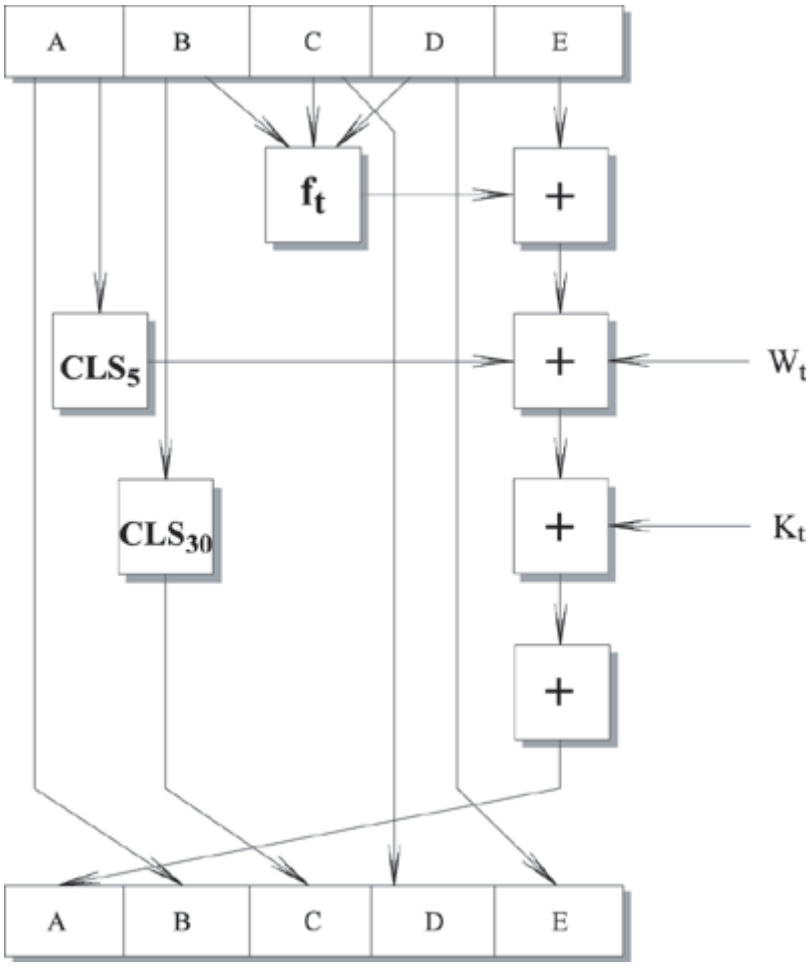

Рисунок 5.6 – Логика выполнения отдельного цикла

Каждая элементарная функция получает на входе три 32-битных слова и создает на выходе одно 32-битное слово. Элементарная функция выполняет набор побитных логических операций, т.е. n-ый бит выхода является функцией от n-ых битов трех входов. Функции следующие:

Номер цикла	**ft (B, C, D)**
$(0 \le t \le 19)$	$(B \wedge C) \vee (\neg B \wedge D)$
$(20 \le t \le 39)$	$B \oplus C \oplus D$
$(40 \le t \le 59)$	$(B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
$(60 \le t \le 79)$	$B \oplus C \oplus D$

32-битные слова W_t получаются из очередного 512-битного блока сообщения следующим образом.

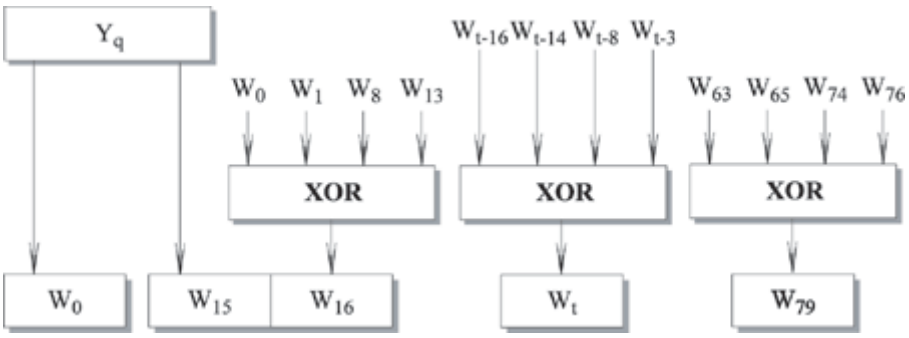

Рисунок 5.7 – Логика выполнения отдельного цикла

Получение входных значений каждого цикла из очередного блока

Первые 16 значений W_t берутся непосредственно из 16 слов текущего блока. Оставшиеся значения определяются следующим образом:

$W_t = W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3}$

В первых 16 циклах вход состоит из 32-битного слова данного блока. Для оставшихся 64 циклов вход состоит из XOR нескольких слов из блока сообщения.

Алгоритм *SHA-1* можно суммировать следующим образом:

$SHA_0 = IV$

$SHA_{q+1} = \Sigma 32\ (SHA_q\ ,\ ABCDE_q\ )$

$SHA = SHA_{L-1}$

Где

IV - начальное значение буфера ABCDE.

$ABCDE_q$ - результат обработки q-того блока сообщения.

L - число блоков в сообщении, включая поля добавления и длины.

$\Sigma 32$ - сумма по модулю 2^{32}, выполняемая отдельно для каждого слова буфера.

SHA - значение дайджеста сообщения.