Файлы (количество строк кода)

- **streebog_generic.c**
  - Строк кода: 1076.
- **streebog.h**
  - Строк кода: 32.

Константы (тип, смысл)

- **buffer0**
  - Тип: **streebog_uint512**.
  - Смысл: Представление числа 0.
- **Ax**
  - Тип: Двумерный массив элементов типа **unsigned long long**.
  - Смысл: В массиве 8 строк и 256 столбцов, и он используется в функции **streebog_xlps**. По своей сути **Ax** — матрица A (преобразование L), неким образом “дополненная” перестановками π (преобразование S) и τ (преобразование P) для оптимизированного выполнения операции LPS (сначала S, потом P, потом L).

$$X[k]\colon V_{512} \to V_{512},\ X[k](a) = k \oplus a,\ k,\ a \in V_{512};$$

$$S\colon V_{512} \to V_{512},\ S(a) = S(a_{63}||\dots||a_0) = \pi(a_{63}||\dots||\pi(a_0),$$

где $a = a_{63}||\dots||a_0 \in V_{512}$, $a_i \in V_8$, $i = 0,\dots, 63$;

$$P\colon V_{512} \to V_{512},\ P(a) = P(a_{63}||\dots||a_0) = a_{\tau(63)}||\dots||a_{\tau(0)},$$

где $a = a_{63}||\dots||a_0 \in V_{512}$, $a_i \in V_8$, $i = 0,\dots, 63$;

$$L\colon V_{512} \to V_{512},\ L(a) = L(a_7||\dots||a_0) = l(a_7)||\dots||l(a_0),$$

где $a = a_7||\dots||a_0 \in V_{512}$, $a_i \in V_{64}$, $i = 0,\dots, 7$.

- **buffer512**
  - Тип: **streebog_uint512**.
  - Смысл: Представление числа 512.
- **C**
  - Тип: Массив величин типа **streebog_uint512**.
  - Смысл: В массиве двенадцать элементов, соответствующих итерационным константам. Константы записаны в виде little-endian по техническим причинам.

C_1 = b1085bda1ecadae9ebcb2f81c0657c1f2f6a76432e45d016714eb88d7585c4fc4b7ce09192676901a2422a08a460d31505767436cc744d23dd806559f2a64507;

C_2 = 6fa3b58aa99d2f1a4fe39d460f70b5d7f3feea720a232b9861d55e0f16b501319ab5176b12d699585cb561c2db0aa7ca55dda21bd7cbcd56e679047021b19bb7;

C_3 = f574dcac2bce2fc70a39fc286a3d843506f15e5f529c1f8bf2ea7514b1297b7bd3e20fe490359eb1c1c 93a376062db09c2b6f443867adb31991e96f50aba0ab2;

C_4 = ef1fdfb3e81566d2f948e1a05d71e4dd488e857e335c3c7d9d721cad685e353fa9d72c82ed03d675d8b71333935203be3453eaa193e837f1220cbebc84e3d12e;

C_5 = 4bea6bacad4747999a3f410c6ca923637f151c1f1686104a359e35d7800fffbdbfcd1747253af5a3dfff00b723271a167a56a27ea9ea63f5601758fd7c6cfe57;

C_6 = ae4faeae1d3ad3d96fa4c33b7a3039c02d66c4f95142a46c187f9ab49af08ec6cffaa6b71c9ab7b40af21f66c2bec6b6bf71c57236904f35fa68407a46647d6e;

C_7 = f4c70e16eeaac5ec51ac86febf240954399ec6c7e6bf87c9d3473e33197a93c9 0992abc52d822c3706476983284a05043517454ca23c4af38886564d3a14d493;

C_8 = 9b1f5b424d93c9a703e7aa020c6e41414eb7f8719c36de1e89b4443b4ddbc49af4892bcb929b069069d18d2bd1a5c42f36acc2355951a8d9a47f0dd4bf02e71e;

C_9 = 378f5a541631229b944c9ad8ec165fde3a7d3a1b258942243cd955b7e00d0984800a440bdbb2ceb17b2b8a9aa6079c540e38dc92cb1f2a607261445183235adb;

C_{10} = abbedea680056f52382ae548b2e4f3f38941e71cff8a78db1fffe18a1b336103 9fe76702af69334b7a1e6c303b7652f43698fad1153bb6c374b4c7fb98459ced;

C_{11} = 7bcd9ed0efc889fb3002c6cd635afe94d8fa6bbbebab07612001802114846679 8a1d71efea48b9caefbacd1d7d476e98dea2594ac06fd85d6bcaa4cd81f32d1b;

C_{12} = 378ee767f11631bad21380b00449b17acda43c32bcdf1d77f82012d430219f9b5d80ef9d1891cc86e71da4aa88e12852faf417d5d9b21b9948bc924af11bd720.

Структуры (поля, смысл)

- **streebog_uint512**
  - Поля: **qword**, массив из восьми величин типа **__le64** (64-битное беззнаковое целое в виде little endian).
  - Смысл: Структура создана для представления 512-мерных двоичных векторов (а также двоичных представлений “512-битных чисел”) путём разбиения их на восемь “64-битных чисел”.

$\mathrm{Vec}_n : \mathbb{Z}_{2^n} \to V_n$ биективное отображение, сопоставляющее элементу кольца $\mathbb{Z}_{2^n}$ его двоичное представление, т. е. для любого элемента z кольца $\mathbb{Z}_{2^n}$, представленного вычетом $z_0 + 2z_1 + \ldots + 2^{n-1} z_{n-1}$, где $z_i \in \{0, 1\}$, $j = 0, \ldots, n-1$, выполнено равенство $\mathrm{Vec}_n(z) = z_{n-1} \| \ldots \| z_1 \| z_0$;

$\mathrm{Int}_n : V_n \to \mathbb{Z}_{2^n}$ отображение, обратное отображению Vec_n, т. е. $\mathrm{Int}_n = \mathrm{Vec}_n^{-1}$;

V_n множество всех n-мерных двоичных векторов, где n — целое неотрицательное число; нумерация подвекторов и компонент вектора осуществляется справа налево, начиная с нуля;

- **streebog_state**
  - Поля: **hash**, **h**, **N**, **Sigma** — все величины типа **streebog_uint512**.
  - Смысл: Структура создана для хранения глобального контекста (значений величин h, N, Σ) и для результата вычисления хеша.

8.1 Этап 1

Присвоить начальные значения текущих величин:

1.1 $h := IV$;

1.2 $N := 0^{512} \in V_{512}$;

1.3 $\Sigma := 0^{512} \in V_{512}$;

Функции (количество строк кода в теле, количество циклов, смысл)

- **streebog_add512**:
  - Вход — 2 константных указателя x и y на **streebog_uint512**, то есть слагаемые, указатель r на **streebog_uint512**, то есть результат суммирования.
  - Выход отсутствует.
  - В конечном счёте r “становится равным сумме x и y”.
  - Строк кода: 12.
  - Смысл: сложение двух “512-битных чисел”, то есть в кольце $\mathbb{Z}_2^{512}$.
  - Циклы: один цикл на восемь проходов.
  - Вызываемые “внутренние” функции отсутствуют.

операция сложения в кольце $\mathbb{Z}_{2^n}$;

- **streebog_finup**:
  - stage3, **выбирает режим работы (shash_desc) и пишет в digest(результат) результат работы**
- **streebog_g**:
  - Вход: struct **streebog_uint512** * h (величина h, которая по исполнении программы станет искомым хешем), const struct **streebog_uint512** *N (параметр N), const struct **streebog_uint512** *m (512-битный блок входных данных).
  - Выход отсутствует.
  - Строк кода: 18.
  - Смысл: реализация функции сжатия, data — вспомогательная переменная. Разберём существенно важные строки.
    #948: *streebog_xlps(h, N, &data);* ~ data := LPS(h XOR N)
    #951: *Ki = data;* ~ K_1 := data
    #952: *streebog_xlps(&Ki, m, &data);* ~ data := LPSX[K_1](m)
    #955: *streebog_round(i, &Ki, &data);* ~ K_i := LPS(C_{i-1} XOR K_{i-1}) и data := LPSX[K_i](data).
    #957: *streebog_xlps(&Ki, &C[11], &Ki);* ~ K_{13} := LPS(C_{12} XOR K_{12}).
    #958: *streebog_xor(&Ki, &data, &data);* ~ data := K_{13} XOR data.
    Здесь преобразование E кончается.
    #961: *streebog_xor(&data, h, &data);* ~ data := data XOR h.
    #962: *streebog_xor(&data, m, h);* ~ h := data XOR m.
  - В конечном счёте выполнили шаг $2.3\ h := g_N(h, m).$
  - Вызываемые “внутренние” функции: **streebog_xlps** три раза, **streebog_round** 11 раз, **streebog_xor** три раза.

Значение хэш-кода сообщения $M \in V^*$ вычисляется с использованием итерационной процедуры. На каждой итерации вычисления хэш-кода используется функция сжатия:

$$g_N: V_{512} \times V_{512} \to V_{512}, N \in V_{512}, \quad (7)$$

значение которой вычисляется по формуле

$$g_N(h, m) = E(LPS(h \oplus N), m) \oplus h \oplus m, \quad (8)$$

где $E(K, m) = X[K_{13}] LPSX[K_{12}] ... LPSX[K_2]LPSX[K_1](m)$.

Значения $K_i \in V_{512}$, $i = 1,...,13$, вычисляются следующим образом:

$$K_1 = K; \quad (9)$$

$$K_i = LPS(K_{i-1} \oplus C_{i-1}), i = 2,...,13. \quad (10)$$

Для краткости вместо $g_{0^{512}}$ будем использовать обозначение g_0.

- **streebog_round**:
  - Вход — целое число i, то есть номер предыдущего раунда, указатель на **streebog_uint512** Ki, то есть ключ (изначально предыдущего раунда), указатель на **streebog_uint512** data, то есть результат раунда.
  - Выход — отсутствует.
  - Строк кода: 2.
  - Вызываемые “внутренние” функции: **streebog_xlps** два раза.
  - Циклы отсутствуют.
  - Смысл: при вызове вычисляет ключ текущего раунда (*streebog_xlps(Ki, &C[i], Ki);*, то есть $K_i := XLPS(C_{i-1}, K_{i-1})$), проводит раунд (data := XLPS(data, K_i), при данном вызове — один, всего — со второго по двенадцатый, потому что первый и тринадцатый проводятся в **streebog_g**) преобразования E (*streebog_xlps(Ki, data, data);*, то есть data := XLPS(K_i, data)).

$$E(K, m) = X[K_{13}] LPSX[K_{12}] ... LPSX[K_2]LPSX[K_1](m).$$

- **streebog_stage2**:
  - Вход —
  - Выход —
  - Смысл:.
  - Строк кода: 8.
  - Вызываемые “внутренние” функции: **streebog_g** один раз, **streebog_add512** два раза.
  - Циклы отсутствуют.
- **streebog_stage3**:
  - **вход - ctx, u8, src, len**
  - **смысл - g, add , add, memzero, g, g, memcpy**
- **streebog_update**:
  - Вход — величина desc типа “указатель на величину типа **shash-desc**”, хранящая информацию о состоянии процесса, величина data типа “указатель на константное целое типа **u8**”, простыми словами — массив байтов, входные данные, которые необходимо хешировать, переменная len вида **unsigned int**, то есть длина входных данных.
  - Выход — величина типа **int**, а именно len.
  - Смысл: разбивает входные данные на блоки по 512 бит.
  - Строк кода: 9.

    - Вызываемые “внутренние” функции: **streebog_stage2**.
    - Циклы: один цикл, проходов столько, сколько 512-битных блоков в исходных данных, на каждом проходе один раз вызывается функция **streebog_stage2**.
- **streebog_xlps**:
    - Вход — 2 константных указателя x и y на **streebog_uint512**, то есть операнды, указатель data на **streebog_uint512**, то есть результат операции
    - Выход - void
    - Смысл: xor x, y, затем порционное применение матрицы итеративное композиции преобразований 8 раз и сдвиг порции в цикле for
- **streebog_xor**:
    - Вход — 2 константных указателя x и y на **streebog_uint512**, то есть операнды, указатель z на **streebog_uint512**, то есть результат операции XOR
    - Выход — величина типа **void**, то есть ничего.
    - В конечном счёте присваивает z->**qword** значение XOR(x -> **qword**, y -> **qword**).
    - Строк кода: 8.
    - Циклы отсутствуют.
    - Смысл: обыкновенный XOR двух “512-битных чисел”.
    - Вызываемые “внутренние” функции отсутствуют.
- **streebog_init**:
    - Вход — переменная desc типа “указатель на величину типа **shash-desc**”, хранящая информацию о состоянии процесса.
    - Выход — величина типа **int**, а именно число 0.
    - Смысл: выбирает режим работы (256 или 512 бит) и в соответствие с этим меняет начальное значение h (вектор IV).
    - Строк кода: 10.
    - Циклы: один цикл на восемь проходов.
    - Вызываемые “внутренние” функции отсутствуют.