МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский технологический университет»

МИРЭА

Институт кибернетики

Кафедра информационной безопасности

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

по дисциплине

«Криптографические протоколы»

На тему:

«Реализация хеш-функции. Хеш-функция SHA-512»

Подготовил

студент группы ККСО–01–14 А.С. Першин

Руководитель работы

А.П. Никитин

Москва, 2019

Оглавление

1. Описание

Рассмотрим стандарт FIPS (Федеральный стандарт обработки информации) PUB 180-4, объединяющий все семейство хеш-функций SHA1 и SHA2. Остановимся на хеш-функциях семейства SHA2.

Хеш-функции SHA-2 разработаны Агентством национальной безопасности США и опубликованы Национальным институтом стандартов и технологий в федеральном стандарте обработки информации FIPS PUB 180-2 в августе 2002 года. В этот стандарт также вошла хеш-функция SHA-1, разработанная в 1995 году. В феврале 2004 года в FIPS PUB 180-2 была добавлена SHA-224. В октябре 2008 года вышла новая редакция стандарта – FIPS PUB 180-3. В августе 2015 года вышла последняя на данный момент редакция FIPS PUB 180-4, в которой были добавлены функции SHA-512/256 и SHA-512/224, основанные на SHA-512 (поскольку на 64-битных архитектурах SHA-512 работает быстрее, чем SHA-256). Длина хеш-значения сообщения алгоритма SHA-512 равна 512 бит.

Хеш-алгоритмы, указанные в этом стандарте FIPS PUB 180-4 называются безопасными потому, что по заданному алгоритму невозможно вычислить следующее:

1) восстановить сообщение по конкретному хеш-значению сообщения;

2) найти два различных сообщения, у которых одно и тот же хеш-значение сообщения (найти коллизию). Любые изменения в сообщении, с очень высокой вероятностью, приводят к различным хеш-значениям.

2. Основные операции

Помимо основных (базовых) операций, используемых в ЭВМ: *XOR*, *AND*, *OR*, *NE* ($\neg$), сложение по модулю 2^{64}, вводятся операции правого поворота (*ROTR*) и правого сдвига (которое также присутствует в ЭВМ) (*SHR*).

$ROTR^n(x)$ – поворот вправо (циклический правый сдвиг) операция, где x это 64 битное слово и n целое число, которое $0 \le n < 64$, операция математически определена как: $ROTR^n(x)=(x >> n) \vee (x << 64 - n)$.

$SHR^n(x)$ – операция правого сдвига, где x это 64 битное слово и n целое число, которое $0 \le n < 64$, операция математически определена как: $SHR^n(x)=x >> n$.

3. Функции и константы

3.1 Функции

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\sum_0^{\{512\}}(x) = ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x)$$

$$\sum_1^{\{512\}}(x) = ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x)$$

$$\sigma_0^{\{512\}}(x) = ROTR^{1}(x) \oplus ROTR^{8}(x) \oplus SHR^{7}(x)$$

$$\sigma_1^{\{512\}}(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^{6}(x)$$

3.2 Константы

SHA-512 использует последовательности из 80 констант по 64 битных слов $K_0^{\{512\}}, K_1^{\{512\}}, ..., K_{79}^{\{512\}}$.

```
428a2f98d728ae22 7137449123ef65cd b5c0fbcfec4d3b2f e9b5dba58189dbbc
3956c25bf348b538 59f111f1b605d019 923f82a4af194f9b ab1c5ed5da6d8118
d807aa98a3030242 12835b0145706fbe 243185be4ee4b28c 550c7dc3d5ffb4e2
72be5d74f27b896f 80deb1fe3b1696b1 9bdc06a725c71235 c19bf174cf692694
e49b69c19ef14ad2 efbe4786384f25e3 0fc19dc68b8cd5b5 240ca1cc77ac9c65
2de92c6f592b0275 4a7484aa6ea6e483 5cb0a9dcbd41fbd4 76f988da831153b5
983e5152ee66dfab a831c66d2db43210 b00327c898fb213f bf597fc7beef0ee4
c6e00bf33da88fc2 d5a79147930aa725 06ca6351e003826f 142929670a0e6e70
27b70a8546d22ffc 2e1b21385c26c926 4d2c6dfc5ac42aed 53380d139d95b3df
650a73548baf63de 766a0abb3c77b2a8 81c2c92e47edaee6 92722c851482353b
a2bfe8a14cf10364 a81a664bbc423001 c24b8b70d0f89791 c76c51a30654be30
d192e819d6ef5218 d69906245565a910 f40e35855771202a 106aa07032bbd1b8
19a4c116b8d2d0c8 1e376c085141ab53 2748774cdf8eeb99 34b0bcb5e19b48a8
391c0cb3c5c95a63 4ed8aa4ae3418acb 5b9cca4f7763e373 682e6ff3d6b2b8a3
748f82ee5defb2fc 78a5636f43172f60 84c87814a1f0ab72 8cc702081a6439ec
90befffa23631e28 a4506cebde82bde9 bef9a3f7b2c67915 c67178f2e372532b
ca273eceea26619c d186b8c721c0c207 eada7dd6cde0eb1e f57d4f7fee6ed178
06f067aa72176fba 0a637dc5a2c898a6 113f9804bef90dae 1b710b35131c471b
28db77f523047d84 32caab7b40c72493 3c9ebe0a15c9bebc 431d67c49c100d4c
4cc5d4becb3e42b6 597f299cfc657e2a 5fcb6fab3ad6faec 6c44198c4a475817
```

4. Подготовка к вычислению хеш-значения

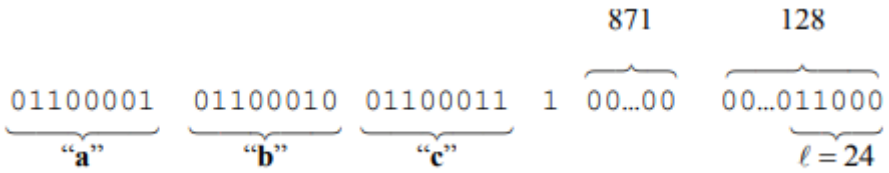
4.1 Дополнение сообщения

Предположим, что длина сообщения *M*, измеренное в битах, равно λ бит.

Добавим в конец бит равный «*1*», за ним последующие *k* нулевых битов, где *k* есть маленькое неотрицательное решение сравнения $\lambda + 1 + k \equiv 896 (mod\ 1024)$.

Затем после всех предыдущих операций добавим в конец *128* битный блок, который есть двоичное представление длины сообщения λ.

Например, сообщение составленное из байт «*abc*» (*1* байт равен *8* бит *ASCII*) в битах имеет длину $8 \times 3 = 24$ бит, добавим к сообщению бит «*1*», затем добавим $896 - (24 + 1) = 871$ нулевых битов, и только после этого добавим *128* битное представление длины сообщения. На выходе получаем *1024* битное дополненное сообщение.



Длина дополненного сообщения составляет *1024* бита.

4.2 Получение сообщения

Для *SHA-512* сообщение и его дополнение представляется как *N 1024* битных блоков, $M^{(1)}, M^{(2)}, ..., M^{(N)}$. Так, полученный 1024 битный блок может быть представлен как 16 64 битных слов, первый 64 битный блок сообщения *i* обозначается $M_0^{(i)}$, следующий блок в 64 бита обозначается как $M_1^{(i)}$,и так далее до $M_{15}^{(i)}$.

4.3 Настройка инициализации начальных хеш-значений $H^{(0)}$

Для *SHA-512* инициализация начальных хеш-значений $H^{(0)}$ представляет собой восемь *64* битных слов в шестнадцатеричной системе счисления:

$$H_0^{(0)} = 6a09e667f3bcc908$$
$$H_1^{(0)} = bb67ae8584caa73b$$
$$H_2^{(0)} = 3c6ef372fe94f82b$$
$$H_3^{(0)} = a54ff53a5f1d36f1$$
$$H_4^{(0)} = 510e527fade682d1$$
$$H_5^{(0)} = 9b05688c2b3e6c1f$$
$$H_6^{(0)} = 1f83d9abfb41bd6b$$
$$H_7^{(0)} = 5be0cd19137e2179$$

5. Хеш-функция SHA-512

SHA-512 может быть использован для вычисления хеш-значения сообщения *M*, имеющего длину λ бит, где $0 \le \lambda < 2^{128}$.

Алгоритм использует:

1) Схему представления блока сообщения как восемь *64* битных слов;

2) Восемь инициализационных хеш-значений по *64* бита каждое;

3) Хеш-значение сообщения представляет собой восемь *64* битных слов. Конечный результат работы алгоритма *SHA-512* – это *512* битное хеш-значение сообщения.

Слова в схеме сообщения маркируются как $W_0, W_1, ..., W_{79}$.

Восемь рабочих переменных обозначаются как *a*, *b*, *c*, *d*, *e*, *f*, *g*, и *h*.

Слова хеш-значений маркируются как $H_0^{(i)}, H_1^{(i)}, ..., H_7^{(i)}$, которые начинаются с начальных хеш-значений $H^{(0)}$, затем итеративно вычисляются их промежуточные хеш-значения обозначаемые как $H^{(i)}$, закачиваются они хеш-значениями $H^{(N)}$.

SHA512 также использует две временные переменные T_1 и T_2.

5.1 Подготовка к алгоритму SHA-512

1. Установка начальных хеш-значений $H^{(0)}$, описанных в секции 4.3

2. Принимаемое сообщение дополняется и преобразуется по правилам в секциях 4.1 и 4.2.

5.2 Вычисление хеш-значения сообщения по алгоритму SHA-512

При вычислении алгоритмом *SHA-512* хеш-значения сообщения используются функции и константы, определенные в разделах 3.1 и 3.2. Сложение «+» производится по модулю 2^{64} .

Каждый блок сообщения $M^{(1)}, M^{(2)}, ..., M^{(N)}$ обрабатываются в порядке, используя следующие шаги:

For i=1 to N:
{

1. Подготовка схемы сообщения $W_0, W_1, ..., W_{79}$, $\{W_t\}$

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{\{512\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{512\}}(W_{t-15}) + W_{t-16} & 16 \le t \le 79 \end{cases}$$

2. Инициализация восьми рабочих переменных: *a*, *b*, *c*, *d*, *e*, *f*, *g*, и *h*, а также $H^{(i-1)}$ вычисленными хеш-значениями:

$$a = H_0^{(i-1)}$$
$$b = H_1^{(i-1)}$$
$$c = H_2^{(i-1)}$$
$$d = H_3^{(i-1)}$$
$$e = H_4^{(i-1)}$$
$$f = H_5^{(i-1)}$$
$$g = H_6^{(i-1)}$$
$$h = H_7^{(i-1)}$$

3. For t=0 to 79:

{

$$T_1 = h + \sum_1^{\{512\}}(e) + Ch(e,f,g) + K_t^{\{512\}} + W_t$$

$$T_2 = \sum_0^{\{512\}}(a) + Maj(a,b,c)$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$b = a$$

$$a = T_1 + T_2$$

}

4. Вычисление следующих хеш-значений $H^{(i)}$:

$$H_0^{(i)} = a + H_0^{(i-1)}$$
$$H_1^{(i)} = b + H_1^{(i-1)}$$
$$H_2^{(i)} = c + H_2^{(i-1)}$$
$$H_3^{(i)} = d + H_3^{(i-1)}$$
$$H_4^{(i)} = e + H_4^{(i-1)}$$
$$H_5^{(i)} = f + H_5^{(i-1)}$$
$$H_6^{(i)} = g + H_6^{(i-1)}$$
$$H_7^{(i)} = h + H_7^{(i-1)}$$

}

После повторения данных этапов один за другим, на протяжении *N* шагов, итоговое *512* битовое хеш-значение сообщения *M* будет иметь вид:

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}$$

6. Результаты реализации алгоритма SHA-512

Разработка производилась в IDE Microsoft Visual Studio 15 Pro. Для реализации задания лабораторной работы было создано общее решение с именем CryptoProtocols. Реализация алгоритма SHA-512 входит в проект SHA512_Hash решения CryptoProtocols.

Для тестирования корректности разрабатываемых проектов в решении CryptoProtocols был создан отдельный проект GoogleTestingSolutionProject модульного тестирования gtest (для unit testing) и gmock (для проверки корректности вызовов методов). Данные пакеты устанавливались через менеджер пакетов NuGet для Visual Studio.

Результат выполнения тест кейсов (значения взяты из [ссылка]) для проверки корректности работы функции хеширования SHA-512 и фиксации времени выполнения для подсчета производительности работы (т.к. gtest замеряет работу вызовов кейсов в микросекундах, то для повышения точности была использована библиотека <chrono> c++11 с точностью до микросекунд) приведены на Рис. 1.

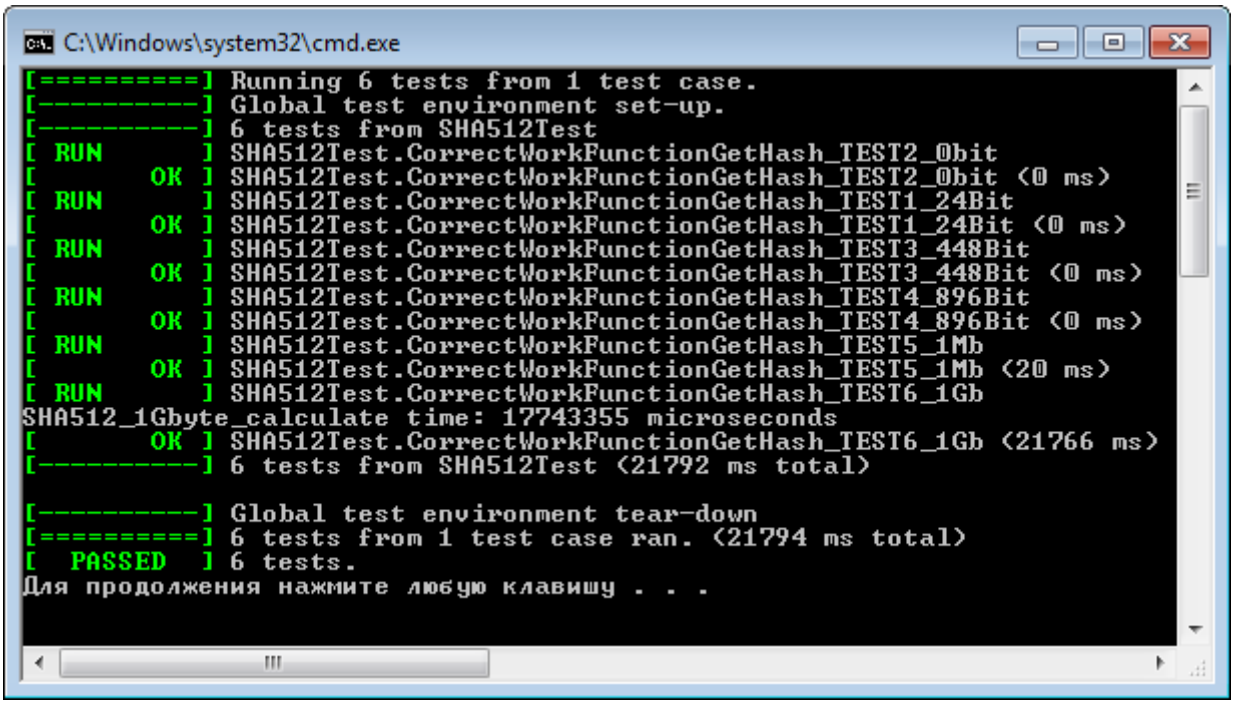


Рис. 1. Результат тестирования реализованного алгоритма SHA-512

Запускался тест на ЦП AMD A6-3410MX (4 ядра, 4 потока) на Рис.2. По полученным данным посчитаем скорость хеширования для данного ЦП. Данные приведены в Табл. 1.

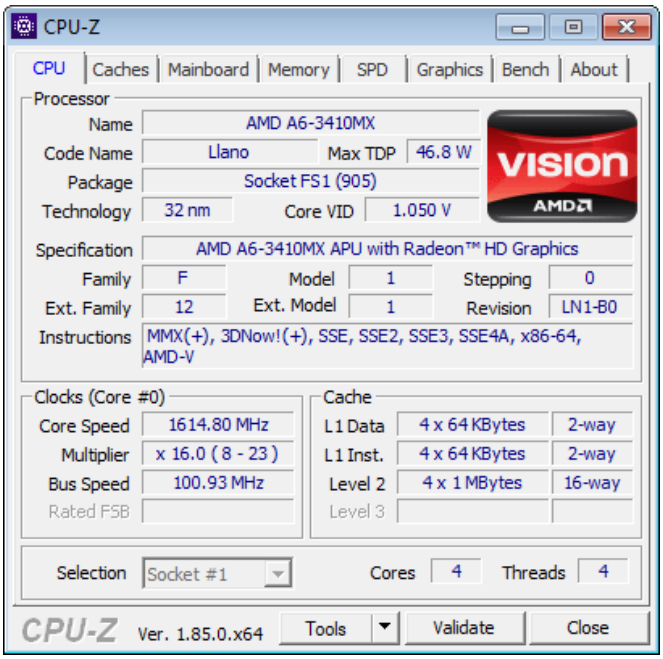


Рис. 2. ЦП AMD A6-3410MX (4 ядра, 4 потока)

Табл. 1. Скорость выполнения хеширования алгоритма SHA-512

Алгоритм и размер хеш-значения	Размер данных [Мбайт]	Скорость [Мбайт/с]
SHA-512/512 бит	1024	57,711746172

Литература

1. FIPS PUB 180-4 «Secure Hash Standard (SHS)» [Интернет ресурс], ссылка https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
2. Test vectors for SHA-1, SHA-2 and SHA-3 [Интернет ресурс], ссылка https://www.di-mgt.com.au/sha_testvectors.html