

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 3](#_Toc2978800)

[2. Основные операции 3](#_Toc2978801)

[3. Функции и константы 3](#_Toc2978802)

[3.1 Функции 3](#_Toc2978803)

[3.2 Константы 4](#_Toc2978804)

[4. Подготовка к вычислению хеш-значения 4](#_Toc2978805)

[4.1 Дополнение сообщения 4](#_Toc2978806)

[4.2 Получение сообщения 5](#_Toc2978807)

[4.3 Настройка инициализации начальных хеш-значений  5](#_Toc2978808)

[5. Хеш-функция SHA-512 5](#_Toc2978809)

[5.1 Подготовка к алгоритму SHA-512 6](#_Toc2978810)

[5.2 Вычисление хеш-значения сообщения по алгоритму SHA-512 6](#_Toc2978811)

[6. Результаты реализации алгоритма SHA-512 7](#_Toc2978812)

[Литература 9](#_Toc2978813)

# 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.

# Основные операции

Помимо основных (базовых) операций, используемых в ЭВМ: *XOR*, *AND*, *OR*, *NE *, сложение по модулю *264*, вводятся операции правого поворота (*ROTR*) и правого сдвига (которое также присутствует в ЭВМ) (*SHR*).

*ROTRn(x)* – поворот вправо (циклический правый сдвиг) операция, где *x* это 64 битное слово и *n* целое число, которое *0 ≤ n < 64*, операция математически определена как: *ROTR n (x)=(x >> n) ∨ (x << 64 - n)*.

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

# Функции и константы

## Функции













## Константы

SHA-512 использует последовательности из 80 констант по 64 битных слов  .



# Подготовка к вычислению хеш-значения

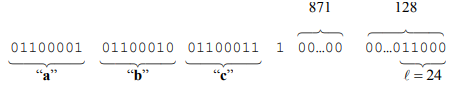
## Дополнение сообщения

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 *λ +1+ k ≡ 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 *8×3 = 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.

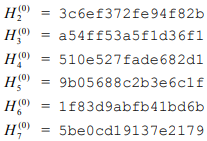
## Получение сообщения

Для *SHA-512* сообщение и его дополнение представляется как *N 1024* битных блоков, . Так, полученный 1024 битный блок может быть представлен как 16 64 битных слов, первый 64 битный блок сообщения *i* обозначается , следующий блок в 64 бита обозначается как ,и так далее до.

## Настройка инициализации начальных хеш-значений

Для *SHA-512* инициализация начальных хеш-значений  представляет собой восемь *64* битных слов в шестнадцатеричной системе счисления:





# Хеш-функция SHA-512

*SHA-512* может быть использован для вычисления хеш-значения сообщения *M*, имеющего длину *λ* бит, где *0 ≤ λ < 2128* .

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 .

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

Слова хеш-значений маркируются как  , которые начинаются с начальных хеш-значений , затем итеративно вычисляются их промежуточные хеш-значения обозначаемые как , закачиваются они хеш-значениями .

*SHA512* также использует две временные переменные T1 и T2.

## Подготовка к алгоритму SHA-512

1. Установка начальных хеш-значений , описанных в секции 4.3

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

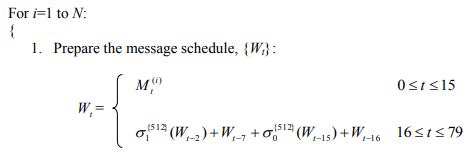
## Вычисление хеш-значения сообщения по алгоритму SHA-512

При вычислении алгоритмом *SHA-512* хеш-значения сообщения используются функции и константы, определенные в разделах 3.1 и 3.2. Сложение «*+*» производится по модулю *264* .

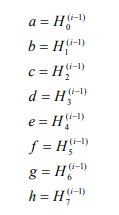
Каждый блок сообщения  обрабатываются в порядке, используя следующие шаги:

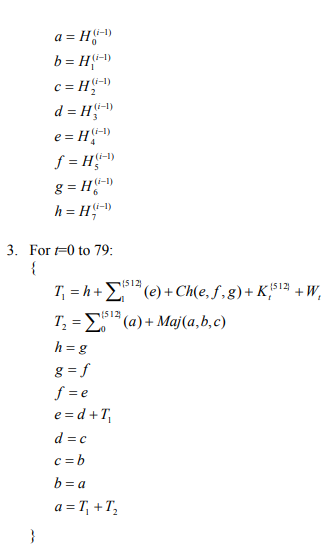


1. Подготовка схемы сообщения , 

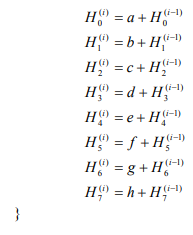


2. Инициализация восьми рабочих переменных: *a*, *b*, *c*, *d*, *e*, *f*, *g*, и *h*, а также  вычисленными хеш-значениями:





4. Вычисление следующих хеш-значений :



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

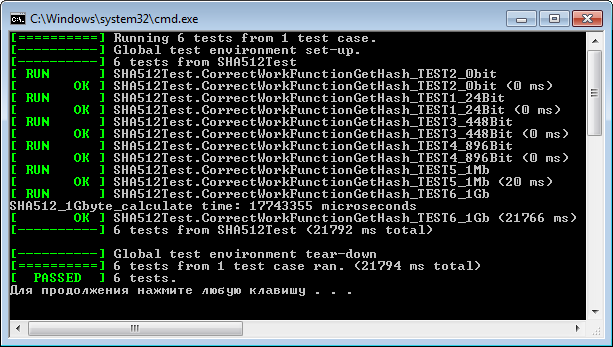


# Результаты реализации алгоритма SHA-512

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.

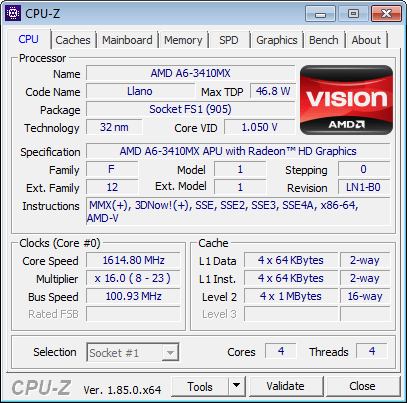
Для тестирования корректности разрабатываемых проектов в решении CryptoProtocols был создан отдельный проект GoogleTestingSolutionProject модульного тестирования gtest (для unit testing) и gmock (для проверки корректности вызовов методов). Данные пакеты устанавливались через менеджер пакетов NuGet для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

Результат выполнения тест кейсов (значения взяты из [[ссылка](http://www.di-mgt.com.au/sha_testvectors.html)]) 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