1. Цель работы

Реализовать алгоритм хеширования SHA-3 и провести эксперимент на «нахождение второго прообраза» и «нахождение коллизий».

2. Описание алгоритма

SHA-3 – алгоритм хеширования, также известный как Keccak. 5 августа 2015 года алгоритм утверждён и опубликован в качестве стандарта FIPS202.

Keccak был выбран в качестве официального алгоритма для SHA-3 в 2012 году, построен он по принципу криптографической губки, в которой данные сначала «впитываются» в губку, при котором исходное сообщение *M* подвергается много раундовым перестановкам *f*, затем результат *Z* «отжимается» из губки.

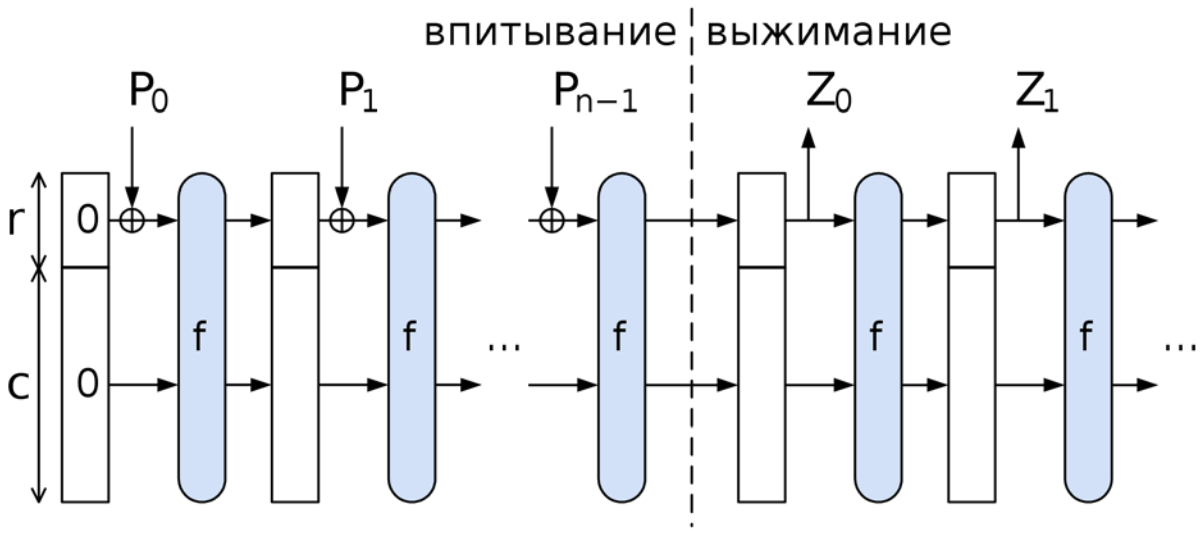

Рисунок 1. Схема губки, используемой в SHA-3

3. Описание реализации

Для удобства проверка реализации будет проводиться на SHA-3-256 и использованы методы

- void add(string data) – добавление сообщения в алгоритм.
- vector<byte> digest() – возвращает вектор хешированных байт.
- string data_to_string(vector<byte> hash) – преобразует байты в шестнадцатеричную.

4. Пример работы программы

Пример 1.
Исходное сообщение: “The quick brown fox jumps over the lazy dog”
Хеш: 69070dda01975c8c120c3aada1b282394e7f032fa9cf32f4cb2259a0897dfc04

Пример 2.
Исходное сообщение: “The quick brown fox jumps over the lazy dog.”
Хеш: a80f839cd4f83f6c3dafc87feae470045e4eb0d366397d5c6ce34ba1739f734d

Пример 3.
Исходное сообщение: “”
Хеш: a7ffc6f8bf1ed76651c14756a061d662f580ff4de43b49fa82d80a4b80f8434a

5. Результаты эксперимента

Для проведения эксперимента были придуманы следующие 3 слова-пароля: “Hello”, “24.11.2001”, “000000000”.

Таблица 1. Результаты экспериментов

	Hello	24.11.2001	000000000
2-й прообраз 8 бит	236.649	219.615	230.314
2-й прообраз 10 бит	1280.38	1286.44	1321.26
2-й прообраз 12 бит	2096.25	2087.77	2089.18
2-й прообраз 14 бит	5062.97	5085.46	5068.94
2-й прообраз 16 бит	11995.6	11796.2	11894.3
Коллизия 8 бит	241.47	229.821	239.878
Коллизия 10 бит	2102.6	2107.61	2117.42
Коллизия 12 бит	4845.14	4767.05	4852.68
Коллизия 14 бит	10050.1	10043.8	10038.7
Коллизия 16 бит	53343.8	53397.5	53458.4

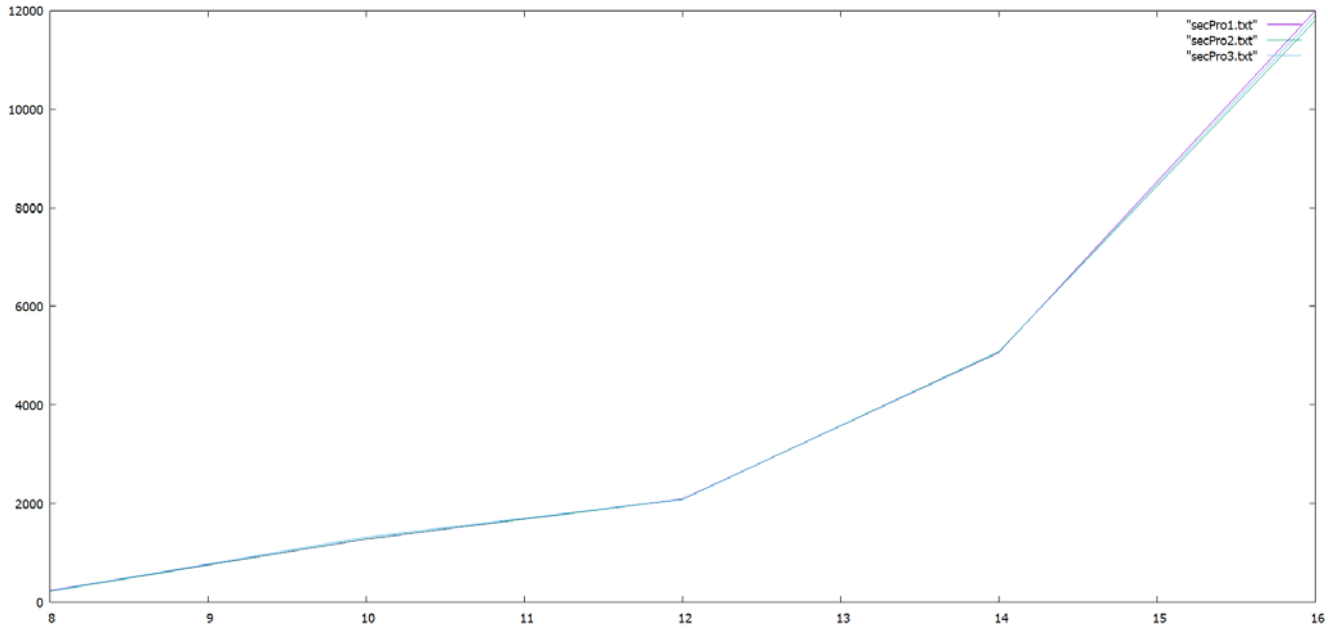

Рисунок 2. График зависимости второго прообраза от количества взятых бит

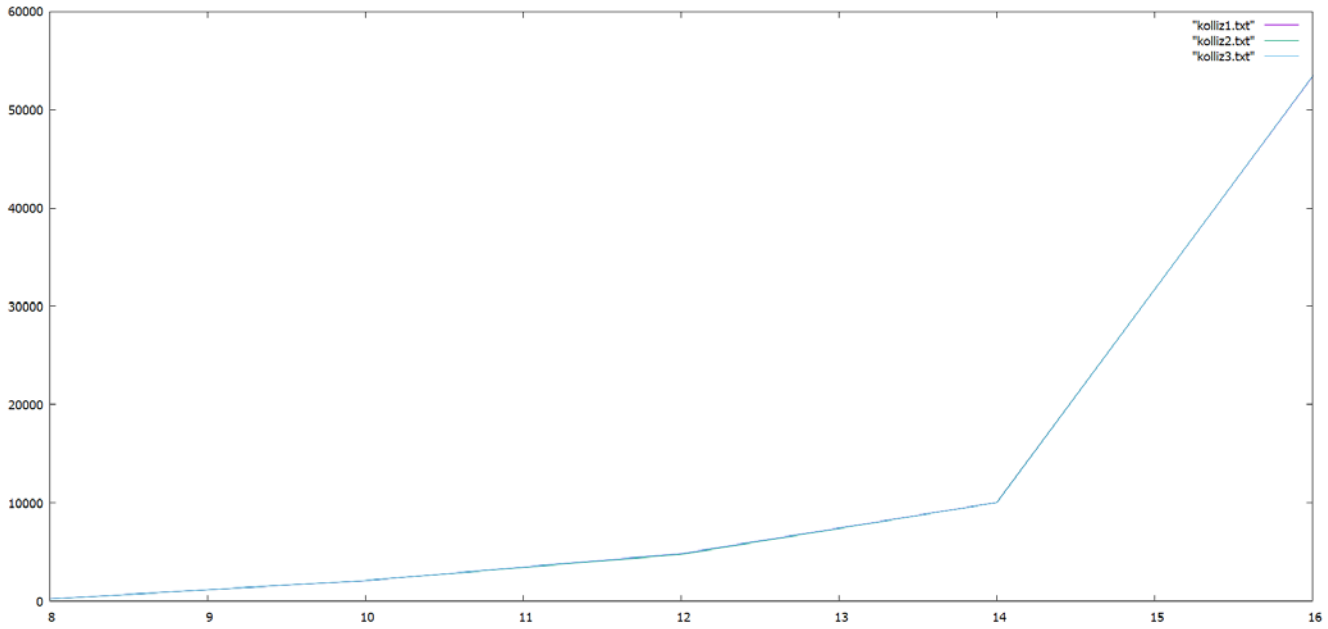

Рисунок 3. График зависимости среднего значения сложности коллизии от количества взятых бит

6. Вывод

В ходе выполнения данной лабораторной работы был реализован алгоритм хеширования SHA3, а также проведен эксперимент (таблица 1, рис. 2-3). Данный алгоритм хеширования в настоящее время используется в различных приложениях, таких как: цифровые подписи, проверка пароля, проверка подлинности хеш-кода, защита от взлома и блокчейн.

По уровню криптостойкости на сегодняшний день алгоритм SHA-3/Keccak является одним из самых безопасных и эффективных алгоритмов хеширования.