мость (пропускная способность), робастность (для защиты от статической атаки и для защиты от целенаправленного повреждения), способность к обнаружению, вид области, независимость от формата.

После проведенных исследований были сделаны выводы относительно каждого стеганографического метода. Все основные форматы графических файлов (методы тестировались именно на них) имеют различные методы сокрытия сообщений со своими сильными и слабыми сторонами. Выбор метода с большой надежностью противостоит методу с высокой скоростью обработки. Например, патч-подход имеет очень высокую устойчивость по отношению к большинству видов атак, но он может скрыть лишь очень небольшое количество информации. Поэтому более разумно скрывать информацию в дополнительных преобразованиях, а не в исходных файлах. ДВП более надежно, потому что позволяет скрыть сообщение в области частот. Данная область менее подвержена зрению человека.

ЛИТЕРАТУРА

1. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие / П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.

2. Шутько, Н. П. Моделирование стеганографической системы в задачах по охране авторских прав / Н.П. Шутько, Н.И. Листопад, П.П. Урбанович // Восьмая Междунар. научно-техн. конф. «Информационные технологии в промышленности» (ITI`2015): тезисы докладов. – Минск: ОИПИ НАН Беларуси, 2015. – С. 30–31.

3. Сайеди, С. А. Сравнение методов стеганографии в изображениях / С. А. Сайеди, Р. Х. Садыхов // Информатика. – 2013. – №1. – С. 66–75.

УДК [004.056+003.26](075.8)

Студ. Т.С. Белявский, А.П. Пулатов
Науч. рук. проф. П. П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

ИСПОЛЬЗОВАНИЕ ХЕШ-ФУНКЦИИ SHA-256 В КРИПТОВАЛЮТНОМ АЛГОРИТМЕ BITCOIN

Криптографическая хеш-функция – это функция, реализующая алгоритм, выполняющий проверку целостности данных, защиту файлов, а также аутентификацию на основе использования математических преобразований, которые преобразовывают массив данных в состоящую из букв и символов строку фиксированной длины.

Хеширование – это процесс ввода информации любой длины и размера в исходной строке и выдача результата фиксированной длины и размера, заданного алгоритмом функции хеширования [1].

Работа криптовалют включают понятие блокчейна, которое неразрывно связано с использованием хеш-функций. Технология блокчейн сравнима с книгой отчётности, куда каждый пользователь вносит свои данные и может быстро сверяться с чужими записями.

В протоколе Биткойна для всех операций хеширования используется алгоритм SHA-256 [2].

Алгоритм SHA-256 представляет собой однонаправленную функцию для создания цифровых отпечатков фиксированной длины (256 бит) из входных данных размером до 2,31 эксабайт (2^{64} бит).

Первым этапом работы алгоритма является преобразование информации в двоичный код и добавление к нему длины входного сообщения размером 512 битов. Вторым этапом является инициализация восьми хеш-значений из дробных частей квадратных корней и 64-х констант из дробных частей кубических корней из простых чисел. Исходный двоичный код делится на сообщения по 32 бита и добавляются дополнительные 48 слов, инициализированных нулем, до размера массива в 64 слова. Обнуленные индексы в конце массива заменяются при использовании соответствующего алгоритма. Переменные *a, b, c, d, e, f, g, h* инициализируются текущим значениям хеш-функции *h0…h7*, и далее запускается цикл сжатия, который изменит значения *a…h*. Все вычисления выполняются 64 раза, меняя переменные *a…h*. Следующим шагом является изменение хеш-значений и добавления к ним соответствующих переменных *a…h*. Финальным этапом является соединение всех хеш-значений в единый хеш [3].

В ходе исследования произведено сравнение скорости работы алгоритма хеширования SHA-256 при различной частоте центрального процессора. Алгоритм хеширования был реализован на языке C++, а скорость его работы измерялась с помощью встроенной библиотеки *chrono*, с помощью таймера *steady_clock*, который представляет так называемые устойчивые часы, ход которых не подвержен внешним изменениям.

Для проведения исследования, был использован 8-ядерный 16-поточный процессор AMD Ryzen 7 2700. Диапазон частоты работы процессора в момент исследования находился в пределах от 3.2 ГГц до 4.2 ГГц с интервалом измерения в 200 МГц. Для более высокой точности исследования данный алгоритм хеширования выполнялся циклически от 500 тыс. до 3 млн. раз. В результате исследования наблюдается

линейная зависимость времени выполнения алгоритма SHA-256 от частоты работы центрального процессора, в результате чего можно сделать вывод о том, что скорость работы алгоритма хеширования SHA-256 практически прямо пропорционально зависит от частоты работы процессора.

Алгоритм SHA-256 имеет преимущества перед другими алгоритмами. Это наиболее востребованный алгоритм хеширования данных в криптовалютах. Он характеризуется достаточной устойчивостью [4]. «Добыча» Биткойна с алгоритмом SHA-256 – это подбор правильного значения хешированной суммы без остановки, перебор чисел для того, чтобы создать очередной блок. В связи с этим, можно сделать вывод: чем мощнее оборудование и чем выше частота работы процессора, тем выше конкурентоспособность майнинга и тем больше шансов стать владельцем того самого правильного блока.

ЛИТЕРАТУРА

1. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие / П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.

2. Как Bitcoin использует криптографию [Электронный ресурс]. Режим доступа: https://bitnovosti.com/2021/09/13/kak-bitcoin-ispolzuet-kriptografiyu. Дата доступа: 15.03.2022.

3. Movable Type Scripts. [Электронный ресурс]. Режим доступа: https://www.movable-type.co.uk/scripts/sha256.html. Дата доступа: 11.03.2022.

4. Алгоритм шифрования SHA-256: особенности, преимущества и недостатки, майнинг. [Электронный ресурс]. Режим доступа: https://ecrypto.ru/blokchejn/algoritm-shifrovaniya-sha-256-osobennosti-preimushhestva-i-nedostatki-majning.html. Дата доступа: 14.03.2022.

УДК 004.05

Студ. А.Г. Блинов
Науч. рук. А.Д. Томко
(кафедра информационных систем и технологий, БГТУ)

НОВОЕВ RUBY ON RAILS 7

RubyonRails(RoR) – один из наиболее популярных и используемых фреймворков для веб разработки. В качестве примера аналогов для RoR, можно привести PythonDjango, а также более старый ASP.NET.

На данный момент, фреймворки для веб приложения занимают важное место в общем рейтинге программных средств для выполнения любых задач. Так как потребность в них есть и будет оставаться, разра-