УДК 003.26

ИССЛЕДОВАНИЕ СТОЙКОСТИ АЛГОРИТМА ХЕШИРОВАНИЯ MD5 НА ВЗЛОМ МЕТОДОМ ПРЯМОГО ПЕРЕБОРА

Зарипов Р.Р.

ФГАОУ ВПО Казанский (Приволжский) федеральный университет,
420008, г. Казань, ул. Кремлевская, д.18

e-mail: zaripov.rinat.r@gmail.com

поступила в редакцию 11 декабря 2014 года

Аннотация

В статье приведены результаты работы программы, восстанавливающей прообраз хэш-функции MD5 методом прямого перебора, запущенной на бытовом компьютере.

Ключевые слова: *MD5, CUDA, метод прямого перебора.*

Введение. Хеширование – преобразование по определённому алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Применяется для построения ассоциативных массивов, контрольного суммирования с целью обнаружения случайных или намеренных ошибок при хранении или передаче, для хранения паролей в системах защиты.

Одним из широко распространённых алгоритмов хеширования является MD5 [1], несмотря на то, что сведения об уязвимости алгоритма начали появляться ещё в 1993 году [2]. Однако все исследования проводились на компьютерах, значительно превышающих по производительности бытовые персональные компьютеры и использовали уязвимости алгоритма.

Целью исследования является исследование стойкости алгоритма хеширования MD5 на взлом методом прямого перебора с использованием мощностей бытового персонального компьютера.

Предметом исследования является программная реализация алгоритма полного перебора, выполненная на языке C++ с применением технологии фирмы NVIDIA CUDA (Compute Unifed Device Architecture).

Для достижения цели исследования необходимо решить следующие задачи:

1. Изучить необходимый теоретический материал;
2. Разработать приложение, реализующее алгоритм полного перебора;
3. Провести исследование скорости вычисления исходного входного массива данных, найденного алгоритмом полного перебора.

В данной работе рассмотрена практическая реализация атаки методом прямого перебора хэш-функции MD5 на графических вычислителях с поддержкой технологии nVidia CUDA, позволяющей ускорить процесс восстановления прообраза. Проведено сравнение скорости реализованного алгоритма восстановления прообразов с реализацией на обычном процессоре.

Постановка задачи. Пусть однонаправленная хэш-функция H(M) применяется к прообразу M и возвращает значение h фиксированной длины m. Функция H(M) называется криптографической, если она является криптостойкой, а именно, удовлетворяет трем основным требованиям:

− устойчивость к нахождению прообраза. Для заданного h должно быть трудно найти такое M, что H(M) = h;

− устойчивость к нахождению второго прообраза. Для заданного M1 должно быть сложно найти такое M2, чтобы H(M1) = H(M2);

− устойчивость к коллизиям. Должно быть сложно найти такую пару M1 и M2, чтобы H(M1) = H(M2).

В данной работе сосредоточимся на задаче восстановления прообраза. Несмотря на то, что хэш-фунция MD5 не является криптостойкой и считается устаревшей, она продолжает использоваться для обеспечения конфиденциальности паролей пользователей в некоторых операционных системах и программных продуктах.

Для эксперимента было написано консольное приложение, осуществляющее восстановление прообраза [3] на графических вычислителях с поддержкой технологии nVidia CUDA. Также для сравнения скорости работы была написана аналогичная программа, использующая для восстановления прообраза ресурсы центрального процессора. В качестве образца для сравнения скорости перебора был использован программный продукт oclHashcat.

Результаты экспериментальных вычислений.

Лабораторный стенд имел следующую конфигурацию:

CPU: Intel Core2 Duo E7400 2.80 GHz

GPU: NVIDIA GeForce 9600 GSO 512

RAM: 1024 Mb

Проводилось восстановление прообразов различной длины символов с алфавитом, состоящим из прописных и строчных латинских символов и цифр(a-z, A-Z, 0-9). Полученные результаты представлены в таблице 1.

Таблица 1. – Скорость подбора исходных значений хеш-функции (среднее значение для 10 запусков).

вычислитель / длина массива	CPU	GPU	oclHashcat plus
4	1 секунда	1 секунда	1 секунда
5	9 секунд	1 секунда	1 секунда
6	5 минут	12 секунд	10 секунд
7	9 часов	4 минут	3 минут 27 секунд
8	-	3 часа 20 минут	3 часа

Выводы. Полученная программа успешно находит прообраз хеш-функции MD5 на персональном компьютере. По результатам, представленным в таблице можно судить, что использование алгоритма MD5 для хранения паролей является опасным, так как прообраз восстанавливается за относительно короткое время.

Список литературы

1) Интернет-ресурс: The MD5 Message-Digest Algorithm. http://tools.ietf.org/html/rfc1321/.

2) den Boer B., Bosselaers A., Collisions for the compression function of MD5, Advances in Cryptology, Proc. Eurocrypt’93, LNCS 765, T. Helleseth, Ed., Springer Verlag, 1994. P.293.

3) Интернет-ресурс: C: реализация функции для получения md5 хэша. http://nig.org.ua/2011/07/poluchenie-md5-hesha-sredstvami-c/.