А. Д. Трофимов

# ХРАНЕНИЕ ДАННЫХ В ХЕШ-ТАБЛИЦАХ

*В современном мире высоких технологий все больше внимания уделяется данным, способам их чтения и хранения, а также способам защитить данные. Обеспечение безопасности хранения данных на данный момент является одной из важнейщих задач, стоящих перед всеми государственными структурами и частными предприятиями. И решением данной задачи на одном из уровней, является использование хеш-таблиц для хранения захешированных данных в них.*

## Введение

Хеш-таболицы - это структура данных, которая предназначена специально для хранения пар ключ-значение. В такой структуре данных возможны три действия: добавление пары ключ-значение, поиск и удаление пары по ключу. Процесс хеширования преобразует входную строку в выходную битовую строку определенной длины. Криптографические хеши должны отвечать нескольким параметрам: они должны быть стойки к коллизиям, т.е. для строки s должно быть вычислительно невычислимо подобрать строку t, хеши которых были бы равны: H(s) = H(t); они должны быть необратимы: для каждого H(s) = x должно быть невозможно найти s зная x; они должны быть стойки к коллизиям второго рода: должно быть вычислитеьно невозможно подобрать парные значения s и S, имеющие одинаковы хеш.

## I. Недостатки существующих решений

К сожалению, в настоящее время не всегда достаточно хранить данные просто захешировав их. Существует несколько способов взлома и получения данных из хеш-таблиц. Начиная от обратного хеширования не стойких хеш-функций и заканчивая подбором хешей с использованием радужных таблиц. Также одной из слабостей простого хеширования является сравнение хешей строк и поиска одинаковых хешей(а значит и одинаковых зашифрованных строк). Второй недостаток существующих решений — это хранение всей необходимой информации для дешифровки хешированных данных в самой Базе Данных.

## II. Решение поставленной задачи

Для защититы от подбора хешей было принято решение использовать современные, стойкие к подбору алгоритмы, такие как MD6, bcrypt/scrypt. Для защититься от поиска одинаковых хешей был использован “добавочный пароль”. Добавочный пароль - это случайная строка, прибавляемая к пароль до его хеширования. Для еще большего усложнения следует добавочный пароль был добавлен после пароля, а не до. В этом случае подбирающий лишился возможности подбирать отдельно части суммы H(добавочный пароль) + H(s). Для решения проблемы когда вся необходимая информация для дешифровки хешированных данных хранилась в самой Базе Данных, был использован другой крайне действенный способ защитить данные — использовать локальный параметр. Суть этого метода заключается в дописывании одинакового второго добавочного пароля ко всем хешам, хешированным по тем же принципам что и первый, за исключением того, что второй добавочный пароль хранится за пределами БД и попадает в нее только из настроек.

## III. Выводы

Таким образом, используя выбранную мной в ходе работы комбинацию методов хеширования данных, была получена криптостойкая База Данных, стостоящая из хеш-таблиц. Разработанная База Данных отвечает всем параметрам безопасности. При реализации были выявлены и отлажены возможные проблемы скорости чтения и хеширования, а конечные алгоритмы хеширования были протестированы на стойкость. После введения в систему комбинации методов хеширования, было выявлено, что никаких дополнительных физических или программных ресурсов система не потребует. Полученный продукт для хранения данных может быть использован на предприятиях, системы которых должны отвечать параметрам безопасности хранения и использования данных.

1. Cisco blog[Электронный ресурс] – Режим доступа: http://blogs.cisco.com/ – Дата доступа: 21.01.2014
2. Хабрахабр [Электронный ресурс] – Режим доступа: http://habrahabr.ru – Дата доступа: 21.01.2014
3. Small Buiseness Data [Электронный ресурс] – Режим доступа: http://www.smallbizlabs.com/ – Дата доступа: 10.02.2014

*Трофимов Алексей Дмитриевич*, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, a.d.trofimov@gmail.com

*Научный руководитель: Гуринович Алевтина Борисовна*, кандидат физико-математических наук, доцент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники, gurinovich@bsuir.by.