УДК 004.021

Студ. В.Б. Кимсо
Науч. рук. доц. Н. В. Пацей
(кафедра информационных систем и технологий, БГТУ)

# АНАЛИЗ МЕТОДОВ ХЕШИРОВАНИЯ ИНФОРМАЦИИ

Хеширование (англ. hashing), как известно, это преобразование по определённому алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины таким образом, чтобы изменение входных данных приводило к непредсказуемому изменению выходных данных. Называются они хеш-функциями или функциями свёртки, а результаты именуются хешем, хеш-кодом или дайджестом сообщения (англ. messagedigest). Дайджест может быть использован в качестве контрольной суммы исходного сообщения, обеспечивая таким образом контроль целостности информации[1-2].

При использовании хеширования в сверке данных производится проверка информации на идентичность оригиналу без использования оригинала. Для этого используется хеш-значение на тот момент проверяемой информации. Вот основные направления применения хеширование при сверке информации:

- для проверки наличия ошибок, к примеру, контрольная сумма может быть передана вместе с основным текстом. На конце контрольная сумма рассчитывается заново. Если вдруг при сравнении с переданным значением будет обнаружено расхождение, то значит, что при передаче данных возникли искажения;

- для проверки парольной фразы. Чаще всего на целевых объектах хранятся лишь хеш-значения парольной фразы. Так как хранить парольные фразы на целевых объектах нецелесообразно. При несанкционированном доступе к файлу с фразами становятся известны все парольные фразы. В случае хранения хеш- значений можно будет узнать только лишь хеш-значения, которые в свою очередь необратимы в исходные данные.

При использовании хеширования для ускорение поиска данных, к примеру, при записи текстовых полей в базе данных может рассчитываться их хеш-код и данные могут помещаться в раздел, соответствующий этому хеш-коду.

Также хеширование используется для криптографической защиты информации. В криптографии к хеш-функциям предъявляются требования такие как стойкость к коллизиям первого рода: для заданного сообщения M должно быть практически невозможно подобрать другое сообщение M', имеющее такой же хеш. А также учитывается стойкость к коллизиям второго рода: должно быть практически не-

возможно подобрать пару сообщений (M, M'), имеющих одинаковый хеш.

Самые распространенные методы хеширования:

- DES/TripleDES. (DataEncryptionStandard, стандартшифрованияданных). В основе данного стандарта лежит добавление 8-битовой строки, состоящий из случайных букв или цифр. Поскольку современные вычислительные средства позволяют быстро раскрыть ключ, то был разработан алгоритм Triple DES, представляющий собой процедуру шифрования такую же как в алгоритме DES, но повторяющуюся 3 раза;
- AES. (Advanced Encryption Standard, улучшенныйстандартшифрования). Этот способ хеширования использует 16 байтовое разбиение на блоки и поддерживает размеры ключей (дописываемых символов) - 16, 24 и 32 бита.;
- MD4/5, SHA MD (MessageDigest).Стандарт является одним из самых известных на сегодняшний день. Алгоритм MD5 в процессе своей работы разбивает пароль на части по 64 байта, затем дописывает последовательности 64 -битового значения случайных символов. После чего создаются 4 переменных с определенным значением. Дальше выполняются 4 цикла по 4 оператора, в которых с помощью специальной функции эти переменные перемешиваются и после каждого цикла записываются в хеш;
- RSA (RivestShamirAlgorithm). Криптографическая стойкость этого алгоритма ниже чем у MD5[3].

Алгоритм выработки хеша должен обладать следующими свойствами:

- постоянством размера. Для входного массива данных произвольного размера результатом должен быть блок данных фиксированного размера;
- вычислительной необратимостью. Для заданного хеша не должно быть способа подбора массива данных под него более эффективным способом, чем перебор по возможным значениям массива данных: свободой от коллизий.

Хеширование может обеспечить хорошие результаты и в том случае, когда набор ключей заранее неизвестен. В этом случае не приходится говорить о генерации совершенной хеш-функции. Даже если известен диапазон значений ключей, то в лучшем случае применение совершенного хеширования потребует наличия в памяти массива с размером, равным мощности множества ключей. Кроме того, неизвестны выполняемые в разумное время алгоритмы генерации совершенной хеш-функции для множества ключей большой мощности.

Распространенным методом является использование эмпирически подобранной хеш-функции [4], которая по значению ключа производит вычисление значения индекса в границах массива и равномерно распределяет ключи по элементам массива. Если выражение ORD(k) обозначает порядковый номер ключа k в упорядоченном множестве допустимых ключей, а N - число элементов в массиве записей, то одной из наиболее естественных хеш-функций является H(k) = ORD(k) MOD N, т.е. взятие остатка от деления порядкового номера ключа на число элементов массива. Такая функция выполняется очень быстро, если N является степенью числа 2. Для числовых ключей функция обеспечивает достаточную равномерность распределения ключей в массиве.

Однако, если ключом является последовательность символов (что чаще всего и бывает), то при применении такой функции возникает большая вероятность выработки одного и того же значения для ключей, отличающихся небольшим числом символов. Ситуацию несколько облегчает использование в качестве N - простого числа. Вычисление функции становится более сложным, но вероятность выработки одного значения для разных ключей уменьшается.

Используются и более сложные способы вычисления хеш-функции [3], основанные, например, на вырезке поднабора бит из битового представления ключа или вычислении квадратичного выражения от ORD(k). Но в любом случае с ненулевой вероятностью хеш-функция может выдать одно значение для разных значений ключа. Такая ситуация называется коллизией ключей. Для решения задач хеширования используются два класса методов: статистические и динамические. Необходимо отметить, что статические методы неприемлемы для задач, в которых размер базы данных меняется часто и существенно. Решение задач хеширования возможно следующими методами: использовать изначальную хеш-функцию, теряя производительность из-за роста коллизий; выбрать хеш-функцию «с запасом», что повлечет потери дискового пространства; периодически менять функцию, пересчитывать все адреса. Такой метод отнимает много ресурсов.

Эти функции полезны для сортировки, которая не потребует никакой дополнительной работы. Возможно избежать множества сравнений, так как, для того, чтобы отсортировать объекты по возрастанию, достаточно просто линейно просканировать хеш-таблицу. Всегда можно создать такую функцию, при условии, что хеш-таблица больше, чем пространство ключей.

Таким образом можно сделать следующие выводы:

- применяя методы хеширования для поиска информации возможно значительно сократить процессорное время;

- для защиты информации разработано достаточно много алгоритмов, среди которых на сегодняшний день одним из самых популярных является алгоритм MD5. Однако, он является и самым требовательным к ресурсам;

- полностью необратимые хеш-функции неприемлемы для решения задач обеспечения крипто стойкости, так как по значению функции нельзя получить однозначное значение аргумента.

ЛИТЕРАТУРА

1. Боричев С.Г., Серов Р.Е. Основы современной криптографии, 2009.

2. Варфоломеев А.А., Жуков А.Е., Пудовкина М.А. Поточные криптосистемы. Основные свойства и методы анализа стойкости. М.: ПАИМС, 2000.

3. КнутД. Искусство программирования, том 3. Сортировка и поиск М.: «Вильямс», 2007.

4. Методы сортировки и поиска [Электронный ресурс]. - URL: http:// www.citforum.ru/programming/theory/sorting/sorting2.shtml – Дата доступа: 20.03.2016

УДК 004.934

Студ. О. В. Манкевич
Науч. рук. доц. Д. В. Шиман
(кафедра информационных систем и технологий, БГТУ)

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ НОРМОКОНТРОЛЯ ТЕКСТОВЫХ ДОКУМЕНТОВ

Зачастую к сопроводительным документам предъявляются особые требования оформления в соответствии с действующими нормативными документами, такими как стандарты предприятия (СТП) и государственные стандарты (ГОСТ) оформления документации, в которых описаны действующие нормы, требования и правила оформления технической и других видов документации.В таком случае разработанная документация будет обязательно подвержена процессу нормоконтроля – проверки документации на соответствие определенным требованиям, нормам и правилам оформления.

Основной целью проведения нормоконтроля является обеспечение однозначности применения технической документации, а также установленных в ней норм, требований и правил.Основными задачами нормоконтроля являются соблюдение требований, правил и норм