ВЗЛОМ ПОЛИНОМИАЛЬНЫХ ХЕШЕЙ ПО РАВНЫМ СТЕПЕНИ ДВОЙКИ МОДУЛЯМ С ПОМОЩЬЮ ПОСЛЕДОВАТЕЛЬНОСТИ МОРСА-ТУЭ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Бутома В.С. Духовник А.Н.

Стройникова Е. Д. - ассистент кафедры информатики

В данной работе будет рассмотрен случай неправильной работы программ, в особенности используемых для решения задач на олимпиадах по программированию, основанных на сравнении (либо любого другого действия) с помощью хешей строки на самом популярном модуле 2^64 (как верхняя граница Int64).

Строка, или рекурсивная последовательность, Морса-Туэ — это набор единиц и нулей, которая может принимать любую длину, равную степени двойки., В 1906 году Акселем Туэ была предложена данная строка, которую также называют "слово Туэ", как апериодическая рекурсивно-вычислимая последовательность. Рекурсивное правило получения строки: s_n = not(s_n-1) + s_n-1, not – логическое отрицание, с начальным значением 1 или 0(в итоге получатся 2 разные последовательности, различаемые инверсированием битов).

Свойства последовательности, начальное значение для которой равно 1 (для начального значение 0 второе и третье свойство меняются четностью):

- Фрактал, поэтому применима в алгоритмах фрактального сжатия.
- Если удалить элементы, которые находятся на четных местах, последовательность не измениться (если длину примем бесконечность)
- Если удалить элементы, которые находятся на нечетных местах, последовательность инверсируется. (если длину примем бесконечность)
- Последовательность не изменяется, если подействовать на нее алгоритмом сжатия Хаффмана.

Некоторая информация о хешах. Если нам дана строка s0..n-1., длины n, то хеш(или полиномиальный хеш, т.к. он основан на сумме степеней некоторого основания с коэффициентами) - это число h = hash(s0..n-1) = [s0 + p*s1 + p2*s2 +… + pn-1*sn-1] (mod B) где p - основания, B - модуль — натуральные числа. Наиболее важное свойство хешей – это то, что у равных строк хеши всегда равны. Таким образом, хеши позволяют быстро проверять, являются ли 2 строки одинаковыми, только сравнив их хеш. Однако существует вероятность коллизий, когда разные строки, будут давать одинаковый хеш.

В ходе работы была найдена интересная особенность этой последовательности. Этой особенностью является то, что она является как бы анти-тестом для программ (в основном применяемых в олимпиадном программировании), которые используют полиномиальные хеши по модулю 〖B= 2〗^64 просто пользуясь переполнением типа. Этот модуль является самым распространенным, т.к. ограничивает стандартный тип long long int(64 бита) в C++ . Строка, сгенерированная по правилу 1 = 'b', 0 = 'a' на строке Морса-Туэ вызывает коллизии уже на размерах строки порядка 〖10〗^3, что намного меньше модуля B. Соответственно все модули, являющиеся степенями 2, также будут неверно работать на таких строках, при этом p может быть абсолютно любым. Далее это будет показано.

Полиномиальный хэш от строки s длины l равен (s0 + p*s1 + p2*s2 +… + pl-1*sl-1,) mod B. В качестве p мы берём простое число большее длины алфавита. Докажем, что hash(s[0... (2k - 1)]) при некотором k совпадёт с hash(s[(2k)... (2k + 1 - 1)]). not(SQ) и SQ встретятся в больших строках не малое количество раз, что явно следует из рекуррентного соотношения, определения данной строки, поэтому вероятность коллизии очень высока. Вместо a и b в строке можно сразу использовать 0 и 1. hash(not(SQ)) — hash(SQ) = T = P0 - P1 - P2 + P3 - P4 + P5 + P6 - P7... ± P2Q - 1, после подстановки 'a' = 0, 'b' = 1 все части, в которых si = 'a' сокращаются.

Применим некоторые преобразования и получим: T = (P1 - 1)(P2 - 1)(P4 - 1)... (P2Q - 1 - 1). Заметим, что P2i + 1 - 1 = (P2i - 1)(P2i + 1) делится на i-ую и ещё на какое-то чётное число P2i + 1. Получаем, что если i-ая скобка делится на 2r, то (i + 1)-ая скобка делится по меньшей мере на 2r + 1.

Таким образом, получаем: (P1 - 1)(P2 - 1)(P4 - 1)...(P2Q - 1 - 1), что делится, по меньшей мере, на 2•22•23•... = 2Q(Q — 1) / 2. А значит, что уже при Q > 11, хеш какой-либо нужной нам подстроки будет зануляться, т.к. его значение делится на B. Получается, что строка будет вызывать коллизии в полиномиальном хеше, следовательно, полиномиальные хеши по модулю 2^64 будут неприменимы ко многим задачам на строках такого типа, т.к. разные строки будут приниматься как равные, если сравнивать их хеши. Заметим, что существуют строки, которые вызывают коллизии для хешей, для абсолютно любого p и B. Для малых B и p достаточно просто найти такую строку простым перебором "в лоб". Для больших значений, полный перебор строк для которых слишком ресурсоемкий, найти такую строку сложно и приходится использовать специальные свойства B и p.

Список использованных источников:

1. Neerc.ifmo [Электронный ресурс]. – Электронные данные. – Режим доступа:

http://neerc.ifmo.ru/wiki/index.php?title=Слово_Туэ-Морса
2. e-maxx [Электронный ресурс]. – Электронные данные. – Режим доступа: http://e-maxx.ru/algo/string_hashes

СТЕГАНОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Губский М. Д.

Стройникова Е. Д. – ассистент кафедры информатики

Во все времена ценность информации была высока. На протяжении столетий люди использовали всевозможные приемы защиты важных данных от несанкционированного доступа, всякий раз придумывая все более сложные и надежные. Как правило, это были криптографические методы. Но порой лишь зашифровать необходимые сведения являлось недостаточным. Для усиления защиты использовались методы сокрытия информации — стеганографические методы.

На сегодняшний день существует множество способов сокрытия данных. Одним из простых в реализации, но не таким простым в обнаружении, является стеганография.

Стеганография (греч. тайнопись) — наука, изучающая передачу и хранение информации при сокрытии самого факта ее существования. Уже в IV в. до н. э. использовали стеганографию, чтобы передавать важные сообщения. Так, например, наносилось необходимое сообщение на обритую голову раба. Когда волосы отрастали, он отправлялся к адресату, который, сбрив голову, считывал сообщение.

За время своего существования стеганография претерпела множество изменений и дополнений. На современном этапе сформировалось три направления стеганографии: классическая, компьютерная, цифровая.

Классическая стеганография — исторически сложившиеся методы сокрытия сведений, которые применяются в повседневной «реальной» жизни, другими словами, некомпьютерные методы. Например, по одной из версий древние шумеры наносили сообщения на глиняные дощечки, после покрывали их слоем глины и вновь наносили надпись, но уже не секретную. Также можно вспомнить запись сообщений на боковой стороне колоды карт, «жаргонные шифры», акростихи и т. д.

Компьютерная стеганография, как видно из названия, включает в себя методы, основанные на особенностях конкретной платформы компьютеров, а также свойствах компьютерных форматов данных. Примерами компьютерных стеганографических методов являются следующие:

- Метод с использованием регистра букв. Его суть заключается в том, что каждый символ секретного сообщения переводится в байт-код. Затем у каждого символа скрывающего текста, которому соответствует единица сообщения, следует поменять регистр. Таким образом, можно зашифровать максимум N/8 символов, где N — количество символов в скрывающем тексте.
- Метод, использующий специфику файловых систем. Как известно, операционные системы для хранения файлов выделяют целое число блоков (для удобства адресации). Соответственно, для хранения маленьких файлов выделяется лишняя память, в которой как раз можно хранить необходимую информацию.
- Использование зарезервированных полей форматов данных также является отличным методом сокрытия сведений. Метод полагается на то, что большинство мультимедийных форматов имеют поля расширения, не использующиеся программой, как правило, они заполнены нулевой информацией.

На сегодняшний день самым важным и наиболее используемым направлением является цифровая стеганография. Это направление характеризуется тем, что необходимая информация внедряется и скрывается в цифровые объекты. К сожалению, цифровая стеганография накладывает некоторые обязательства, такие как сохранение целостности и аутентичности файла, поэтому обычно в качестве контейнеров (хранилищ данных) используют медиафайлы. Существуют следующие алгоритмы встраивания скрываемой информации:

- работающие с цифровым сигналом напрямую (метод LSB);
- внедрение скрытой информации (наложение секретного изображения, аудиофайла, текста поверх оригинала; часто используется для внедрения цифровых водяных знаков);
- использование форматов файлов (к примеру, запись в метаданные).

Одним из самых известных алгоритмов встраивания является метод LSB (Least Significant Bit — англ. наименьший значащий бит). Он основан на замене последних, незначащих бит в контейнере (графическом, аудио, видеофайле) на биты секретного сообщения. Метод опирается на низкий порог чувствительности человеческих органов. Например, изменение в 8-битном изображении двух последних бит приводит к изменению в цвете максимум на 3 бита, такие градации не отображают многие программы (считая их несущественными), не говоря уже о человеческом глазе.

В общем случае система маскирования имеет следующий вид (рис. 1):