Московский физико-технический институт (ГУ МФТИ)
Факультет Радиотехники и Кибернетики

Коллизии хеш-функций MD5, SHA-0,1
(Collisions in hash functions MD5, SHA-0,1)

Федоров В.В.
27 апреля 2007г

Эссе по курсу "Защита информации", кафедра радиотехники
http://www.re.mipt.ru/infsec

1. Введение

Функция хеширования – это функция, преобразующая некоторые входные данные в некоторое число заданной длины, которое можно рассматривать, как цифровой отпечаток этих данных. Такие отпечатки называют хешами. Хеши используются, например, для индексирования хеш-таблиц.

Криптографическая функция хеширования – это функция хеширования, которая обладает следующими свойствами:

1) Необратимость

   Если известен хеш h, то нахождение исходных данных d по этому хешу (так что h=hash(d)) – вычислительно невозможная задача

2) Слабая стойкость к коллизиям

   Если даны входные данные d1, то нахождение других данных d2, таких что d1 != d2 и hash(d1) = hash(d2), должно быть вычислительно невозможно.

3) Сильная стойкость к коллизиям

   Нахождение d1 и d2, таких что hash(d1) = hash(d2), вычислительно невозможно.

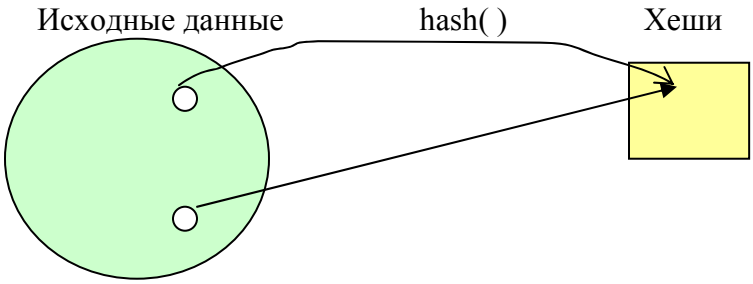
   Это требование влечёт за собой увеличение длины хеша в два раза по сравнению с тем, что необходимо для необратимости, из-за того, что возможна атака, основанная на парадоксе дней рождения.

Однако даже если функция удовлетворяет всем этим критериям, она всё равно может быть ненадёжной. Например, для некоторых функций хеширования существуют так называемые «атаки на увеличение длины»: зная hash(d), криптоаналитик может вычислить hash(d || b) = hash(hash(d), b).

Функции хеширования весьма полезны и находят обширное применение в криптографии:

1) Для проверки целостности сообщения
2) Для верификации паролей
3) В цифровых подписях (ЭЦП)
4) В кодах с обнаружением изменений (MDC)
5) Для аутентификации сообщений (MAC)

2. Коллизии хеш-функций



Практически любая хеш-функция подвержена коллизиям. В хороших функциях хеширования коллизии найти сложнее и они происходят много реже, чем в плохих.

Идеальную хеш-функцию можно составить лишь в тех случаях, когда заранее известны все возможные варианты входных данных.

В функциях, на вход которых можно подавать данные произвольной длины, а на выходе требуется хеш фиксированной длины, коллизии неизбежны, потому что любой хеш может соответствовать нескольким значениям на входе.

Существует два типа атак:

Атака на обнаружение коллизий – процедура нахождения двух произвольных входных данных, имеющих одинаковые хеши.

Атаки на нахождение прообраза – процедура нахождения произвольных данных, хеш которых совпадает с хешем других данных.

Оптимальная атака на нахождение прообраза для n-битной хеш-функции требует порядка 2^n операций. А вспомнив про парадокс дней рождения, можно ожидать, что нахождение коллизий потребует порядка $2^{n/2}$ операций.

Существенная часть современных хеш-функций построена по принципу итераций Меркла - Дамгарда (Merkle, Damgard). Исходное сообщение D разбивается на блоки D[i] фиксированной длины, итеративно используется «функция сжатия» F, принимающая в качестве аргументов очередной блок исходного текста и результат предыдущей итерации. h[i] = F(h[i-1],D[i]). Для первой итерации в качестве предыдущего результата подаётся «инициализирующий вектор» - IV. Выход последней итерации – хеш. Эта схема несовершенна, и поэтому во многих функциях хеширования относительно легко можно найти коллизии.

3.Атака на SHA-1

Расширимые сообщения – разновидность множественной коллизии. При этом исходные сообщения имеют разные длины, но одинаковые промежуточные хеши на входе последней функции сжатия, до этапа обработки длины сообщения.

Если в функции сжатия можно легко найти фиксированные точки, то составление расширимых сообщений потребует примерно в два раза больше операций, чем нахождение коллизии в хеш-функции. Фиксированная точка – это пара (h[i-1],D[i]), такая что F(h[i-1],D[i]) = h[i-1]. В функциях сжатия, использующихся в MD4,5, SHA0,1 и некоторых других, фиксированные точки найти легко.

Генерируется $2^{n/2}$ фиксированная точка (h[i], D[i]); генерируется $2^{n/2}$ пар (F(IV,M(i)), M(i)) где M(i) – блоки требуемого размера, разные для каждого i. Далее, для совпадающих h[i] и F(IV,M(i)) создаются пары (D[i], M(i)) – расширимые сообщения.

После этого, чтобы составить сообщение желаемой длины, нужно к первому блоку расширимого сообщения второй блок требуемое число раз: D[i], M(i), M(i), …

Атака с использованием очень длинных сообщений.

Криптоаналитик сначала хеширует очень длинное сообщение, например, длиной 2^{55} блоков. Далее, нужно найти второе сообщение, которое имеет такой же хеш, как и первое. Ищут D такое, чтобы F(IV, D) совпадало с каким-либо промежуточных хешем для первого сообщения (число таких хешей – порядка 2^{55}, предполагают, что требуется перебрать порядка 2^{105} блоков). Таким образом, вероятность того, что хеш каждого проверяемого блока совпадёт с одним из промежуточных хешей, равна 2^{-105}.

В итоге получается более короткое сообщение, имеющее такой же хеш, как и исходной длинное, перед последней итерацией, до обработки длины. Но в алгоритме Меркла –

Дамгарда сообщения дополняются длиной, и поэтому после заключительной стадии хеши будут разные.

Но с помощью расширимых сообщений эту неприятность можно обойти.
Наибольшая длина сообщения для SHA1 составляет $2^{64} – 1$ бит, это чуть меньше чем 2^{55} блоков. Пусть длинное сообщение составляет $2^{54} + 54 + 1$ блок.
Подсчитываются все промежуточные хеши. Находится расширимое до $2^{54} + 54 + 1$ блока сообщение. Так как нахождение фиксированных точек – вычислительно лёгкая задача, требуется 2^{82} вызовов функции сжатия.
Перебирается 2^{106} блоков до совпадения хеша с одним из $2^{54} + 54$ промежуточных блоков.
Наконец, расширимое сообщение расширяется, компенсируя пропуск блоков исходного длинного сообщения. Получается второй прообраз.
То есть, атака на сообщение из 2^k блоков требет около 2^{n-k+1} вызовов функции сжатия.

Более практические результаты получены исследователями из Китая - Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu (Shandong University in China): нахождение коллизии за 2^{69} операций, используя дифференциальные пути, что много меньше, чем при переборе - 2^{80}.

3. Атака на SHA-0

Локальные коллизии.

Локальная коллизия – это коллизия, происходящая внутри хеш-функции за несколько итераций. Очень важно, что в SHA-0 существует такая коллизия, она может начинаться на любой итерации, и является основой для обнаружения полных коллизий.

Дифференциальные пути (Ванг, Wang).

Это, грубо говоря, последовательность локальных коллизий, возможно, перекрывающихся. Для этого требуется найти набор подходящих начальных итераций. Используют 80-битный «вектор возмущения» с маленьким весом Хэмминга. Показано, что любые 16 подряд идущих переменных в этом векторе определяют остальные, то есть, всего 2^{16} возможных комбинаций.
Для того, чтобы вектор возмущения привёл к возможной коллизии, нужно наложить дополнительные ограничения, например, биты 75 – 79 должны быть нулевыми, чтобы коллизия произошла на последнем шаге. Это приводит к тому, что можно свободно выбирать лишь 6 бит в векторе. Учитывая дополнительное условие, что не должно быть одинаковых идущих подряд 17 бит, получаем следующие три возможности: (001000),

(000100) и (000101). Далее рассматриваются отклонения составных частей IV (a0,b0,c0,d0,e0) , их нужно компенсировать. Разница в b, c, d практически во всех случаях убирается функцией IF. В то время как для a и e нужно предпринимать специальные меры, в том числе, можно подбирать отклонения b, c, d. Ванг заметил, что при этом разница в промежуточных хешах полностью определяется значением a, и ввёл пять групп отклонений a:

1) Различия из-за неполных локальных коллизий, зависящие от выбранного пути.
2) Различия из-за возмущений. Важны для получения локальных коллизий.
3) Различия, требуемые для компенсации отклонений в e.
4) Различия для остальных компенсаций.

Оценочно, сложность «прямой» реализации такой атаки составляет 2^{42} хеш-операций, так как при изменении сообщения требуется удовлетворение 42 условий. Однако, предварительно составив таблицу «хороших» сообщений, можно снизить количество операций до 2^{39} операций, а используя метод многоблочных коллизий - до 2^{33}.

4. Атака на MD5

Ещё в 1996 году Ханс Доббертин (Hans Dobbertin) нашёл «коллизии» в MD5: при использовании определённых IV, отличных от заданных в стандарте, можно для заданного сообщения построить второе, такое что их хеши будут одинаковыми, добавляя к 14му 32х разрядному слову в блочном буфере число 2^9. Хотя это не совсем коллизия, она показывает слабость алгоритма.

Также, было обнаружено, что для MD5(X) = MD5(Y) получаем MD5(X||S) = MD5(Y||S).

Вслед за Доббертином китайские исследователи Ванг и Ю (Xiaoyun Wang, Hongbo Yu) предложили свой мощный метод нахождения коллизий. Они рассматривают сообщения, состоящие из двух блоков (M0, M1), и ищут такие, чтобы после второй итерации хеши совпадали. Нахождение первых блоков – M0, M0’ – требует порядка 2^{39} операций MD5, нахождение вторых – M1, M1’ – 2^{32} операций. (Интересно, что с помощью такой атаки в MD4 можно найти коллизии за секунды, а также найти прообразы для многих хешей)

Дифференциальная атака.

Исследователи скомбинировали XOR и целочисленное вычитание в качестве меры разници для атаки. Вместо вероятностного метода используется метод модификации сообщений, состоящий в предварительном составлении некоторых слов сообщения, чтобы сообщение

удовлетворяло необходимым условиям дифференциальной атаки для 16 первых итераций (первого раунда) MD5.

Сначала выбирается произвольное сообщение M0. Оно модифицируется для соответствия условиям. Тогда, с вероятностью 2^{-37}, M0 и M0'=M0+deltaM0 дадут дифференциал после первой итерации – (deltaH1, deltaM1). К M0 и M0' применяется функция сжатия для проверки условий коллизии.

Далее, выбирается произвольное M1, модифицируется. M1 и M1' после второй итерации дадут второй дифференциал – deltaH, который равен нулю с вероятностью 2^{-30}. И проверяется, действительно ли получили коллизию.

Один раз найдя пару M0, M0', следующие M0 можно выбирать, просто меняя последние два слова из предыдущего M0, таким образом, основная доля работы занимает порядка 2^{39} операций MD5. Для такой пары можно найти множество пар M1, M1', приводящих к коллизиям, что облегчает задачу криптоаналитика.

5. *Как можно использовать коллизии.*

Представим себе ситуацию, когда одна организация заключает договор по переводу денег другой организации для произведения некоторых работ. И если они используют уязвимую функцию хеширования для аутентификации договоров, то стороне-злоумышленнику достаточно создать два разных договора с одинаковыми хешами. Один из них – «честный», а во втором можно, например, потребовать выполнения работ бесплатно…

6. *Литература*

1. Schneier on security. http://www.schneier.com/blog/archives/2005/02/sha1_broken.html
2. Wikipedia
   http://en.wikipedia.org/wiki/MD5
   http://en.wikipedia.org/wiki/Hash_function
   http://en.wikipedia.org/wiki/SHA-1
3. Xiaoyun Wang, Hongbo Yu. How to break MD5 and other hash functions. http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf
4. Xiaoyun Wang, Yiqun Yin, Hongbo Yu, Finding Collisions in the Full SHA-1,Crypto'05. http://www.infosec.sdu.edu.cn/paper/Finding%20Collisions%20in%20the%20Full%20SHA-1.pdf