Проверка качества хеширования

Аметов Имиль, гр. М07-903

20 мая 2020 г.

Задача:

Предложите способ проверки качества хэширования, слабой или сильной стойкости, отсутствия взаимной информации между исходным текстом и результатом хэширования.

Предлагаемое решение:

Для проверки отсутствия взаимной информации между открытым сообщением и результатом хэш-функции можно попробовать применить искусственные нейронные сети. Если получится натренировать сеть на некотором массиве открытых данных и она будет более-менее хорошо предсказывать значение хэш-функции для открытых сообщений не очень отличающихся от тренировочного набора, то это значит что хэш-функция не соответствует предъявленному требованию.

Для проверки стойкости хэш-функции можно использовать то свойство, что для идеальной хэш-функции требуется равномерное распределение значений этой функции. Можно применить такой способ: взять какое-либо длинное сообщение (например, длина сообщения не меньше 256 бит) захэшировать его и сохранить результат. Потом в исходном сообщении изменить первый бит, далее захешировать получившееся сообщение. Сохранить результат. Вернуться к исходному сообщению и изменить второй бит. Захешировать сообщение, сохранить результат. Продолжить со следующим битом и так далее. Если по собранной статистике окажется, что в каждом случае изменялось больше половины битов, и изменённые биты распределялись равномерно, то эта хэш-функция, скорее всего, достаточно стойкая.

Ответ Афанасьева:

Вы представили вобщем разумный текст и очень краткий.

На самом деле, если воспринимать это задание серьезно, это оно трудное, да еще и с подвохом.

В чем подвох?

Хэш-функция — это односторонняя и однозначная функция, которая каждому значению на входе ставит в соответствие одно и только одно значение на выходе и которая не обращается, т.е. не возможно вычислить значение на входе для любого заданного выхода.

Это означает, что хэш-функция реализует строгое (или почти строгое, с точностью до коллизий) взаимно однозначное соответствие между ее входом и выходом. Это означает, что между входом и выходом хэш-функции

есть полная взаимная информация. Это означает, что сам вопрос о взаимной информации противоречит определению и свойства хэш-функции.

К сожалению, должен заметить, что все не обратили на это внимания.

Ваше предложение про нейронную сеть правильно по смыслу но совершенно не реализуемо, так как требует огромного объема тестов на обучение. По существу, это обучение сопоставимо с проверкой сильной или слабой устойчивости к коллизиям, а для этого требуется число тестов порядка корня квадратного из общего числа возможных входов ($2^{256/2}$).

Второе предложение тоже разумно и даже реализуемо только плохо сформулировано.

Представим множество значений входа хэш-функции как одномерный или многомерный числовой массив. В одномерном случае это будет числовая линейка, каждая точка которой является 256-битовым целым числом от 0 до $(2^{256}) - 1$, а в 256-мерном случае это будет двоичный вектор и 256 координат. Введем метрику на этих массивах, например покоординатную сумму квадратов разностей, или сумму абсолютных разностей, или метрику Хэмминга (число не совпадающих координат).

Тест 1: выберем M случайных точек во входном массиве и посчитаем среднюю метрику для выбранных точек. Все выбранные точки пропустим через хэш-функцию и посчитаем такую же метрику для полученного массива выходов. Проверяемая гипотеза — средняя метрика на выходе не меньше средней метрики на входе хэш-функции.

Тест 2: составим массив сдвигов C как массив чисел с ограниченной метрикой (в вашем предложении это вектора с одной единичкой). Для любой точки из массива M прибавим точку из C и для каждой такой суммы найдем значение хэш-функции. Выполним это для всех точек из C и, затем, для всех или некоторых точек из M. Найдем среднюю метрику на выходе. Проверяемая гипотеза — средняя метрика на выходе больше средней метрики на входе хэш-функции.