25/02/2020 - Элементы комбинаторного анализа в Теории Вероятностей

На семинаре мы обсудили две комбинаторные задачи теории вероятностей и их приложения к вопросам защиты информации.

Первая часть семинара была посвящена «атаке дней рождения» и лежащему в её основе «парадоксу» дней рождения.

Атака дней рождения применяется для нахождения коллизий второго рода у хэш-функций.

Определение 1. *Функция* $h : U \to \{0, 1, \dots 2^m - 1\}$, *сопоставляющая каждому элементу* $x \in U$ *множества входных данных вектор фиксированной длины* $h(x)$ *(двоичное представление образа) называется хэш-функцией.*

Если при этом для h выполняются следующие свойства:

- *Необратимость*: для всех $u \in \{0, 1, \dots 2^m - 1\}$ вычислительно невозможно подобрать $x \in U$ такой, что $h(x) = y$;
- *Устойчивость к коллизиям первого рода*: для любого $x_1 \in U$ вычислительно невозможно подобрать $x_2 \in U$ такой, что $h(x_1) = h(x_2)$;
- *Устойчивость к коллизиям второго рода*: вычислительно невозможно подобрать пару $(x_1, x_2) \in U^2$ $(x_1 \neq x_2)$ такую, что $h(x_1) = h(x_2)$;

то h называется криптографической хэш-функцией.

Атака дней рождения предназначена для поиска коллизий второго рода основана на следующей задаче:

Задача 1. *Какое количество* N *элементов множества входных данных* U *нужно перебрать, чтобы с вероятностью как минимум в* 0.5 *найти пару элементов с совпадающим значением хэша?*

Нетрудно видеть, что эта задача является обобщением задачи о днях рождения:

Задача (О днях рождения)**.** *Какое количество* N *людей нужно собрать в одной комнате, чтобы с вероятностью как минимум в* 0.5 *хотя бы у двух из присутствующих совпали дни рождения?*

Мы решили обе эти задачи, и установили, что в задаче о днях рождения $N = 23$, а в задаче о коллизиях второго рода у хэш-функций $N \approx 2^{\frac{m}{2}}$ (что значительно меньше, чем 2^m).

Где почитать:

1. Атака «дней рождения», Википедия.

2. Атака дней рождения, Хабр.

3. Birthday attack, Wikipedia.

На второй половине семинара мы разобрали алгоритм решения задачи о разборчивой невесте:

Задача 2. *Предположим, что в некотором царстве принцесса решила, что ей пора найти себе жениха. Она выбирает из n претендентов.*

Принцесса общается с претендентами в случайном порядке, с каждым не более одного раза. О каждом текущем претенденте известно, лучше он или хуже любого из предыдущих. В результате общения с текущим претендентом принцесса должна либо ему отказать, либо принять его предложение. Если предложение принято, процесс останавливается, если невеста отказывает жениху, то вернуться к нему позже она не сможет.

Цель — выбрать лучшего претендента.

Мы построили оптимальный (в некотором смысле) алгоритм решения этой задачи ($1/e$-law of best choice), придерживаясь которого, принцесса будет выбирать наилучшего претендента с вероятностью приблизительно равной $\frac{1}{e}$.

Где почитать:

1. Задача о разборчивой невесте, Википедия.

2. С. М. Гусейн-Заде, Разборчивая невеста.

3. Secretary problem, Wikipedia.