Белый Шум. Задание 4.

Бехруз Аъзам, Дмитрий Лишуди, ПМИ-193 30.03.2023

1 Постановка задачи

Существуют N пользователей сети, q из которых являются Мориарти. Пользователи используют M файлов с белым шумом для общения. Каждый пользователь хранит лишь подмножество файлов. Для протокола общения требуется, чтобы для каждой пары пользователей, нашлось хотябы примерно l общих файлов. Требуется найти оптимальное общее кол-во файлов и оптимальное распределение файлов между пользователями при котором вероятность компрометации сообщений между двумя пользователями была бы менее α.

2 Решение

2.1 Наблюдение

Можем заметить, что не имеет смысла давать некоторым пользователям больше файлов чем остальным, т.к. это поставит их в превелигированное положение из которого будет легче скомпромитировать чужие сообщения. Из этого можем сделать вывод, что у каждого пользователя должно быть ровно k файлов с белым шумом.

2.2 Оганичения на k

Давайте попробубем задать ограничения на количество файлов выдаваемых пользователю.

Представим алгоритм при котором мы из M файлов выдаем случайные k. Из этого вытекают некоторые вероятности:

$$P(a_i \in A) = \frac{k}{M}$$

Получив вероятность того, что конкретный файл будет у конкретного пользователя, найдем ожидание мощности пересечения двух случайных пользователей.

$$\mathbb{E}\,|A_i \cap A_j| = M \cdot P(a_i \in A)^2 = \frac{k^2}{M} \geq l$$

Из этого найдем оценку снизу на k.

$$k \geq \sqrt{Ml}$$

Теперь попробуем найти оценку сверху из следующего условия:

$$P(A_i \cap A_j \subseteq \bigcup_{q \in Q} A_q) < \alpha$$

$$P(a_i \subseteq \bigcup_{q \in Q} A_q) = 1 - (1 - P(a_i \in A))^{|Q|}$$

Грубо оценим $|A_i \cap A_j|$ как константу равную l.

$$P(A_i \cap A_j \subseteq \bigcup_{q \in Q} A_q) \approx (1 - (1 - \frac{k}{M})^{|Q|})^l < \alpha$$

$$k < M(1 - \sqrt[|Q|]{1 - \sqrt[l]{\alpha}})$$

Посмотреть как выглядят эти ограничения можно на графике 1. Далее, для минимизации M и k, будем выбирать их из слудующего уравнения:

$$\sqrt{Ml} = k = M(1 - \sqrt[|Q|]{1 - \sqrt[l]{\alpha}})$$

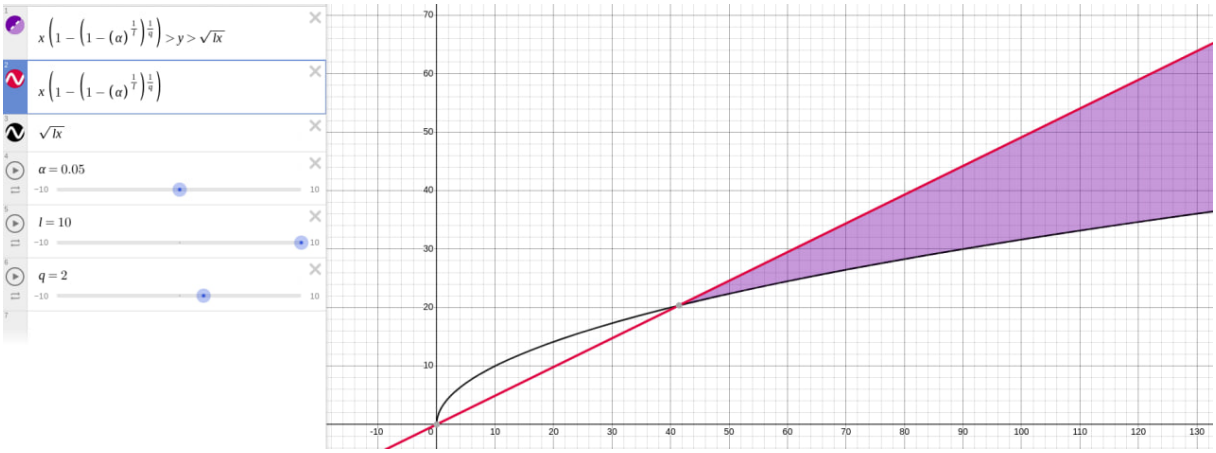

Рис. 1: В данном примере у нас 2 Мориарти от которых мы хотим защититься с вероятностью более 95%. При этом мы хотим, чтобы у каждой пары пользователей было около 10 общих файлов. По оси X - общее кол-во файлов с белым шумом M. По оси Y - кол-во файлов на человека k. Фиолетовым отмечена оптимальная зона параметров M и k.

2.3 Еще наблюдений

С ростом числа мориарти мы вынуждены сильно увеличивать кол-во файлов в природе. Для наглядности:

- Для 2 Мориарти достаточно 40 файлов в природе и каждый имеет при себе 20 из них.
- Для 10 Мориарти достаточно 625 файлов в природе и каждый имеет при себе 79.
- Для 20 Мориарти достаточно 2342 файлов в природе и каждый имеет при себе 153.

На графике 2 видно, что этот тренд квадратичный. Для интереса можно посмотреть зависимость вероятности компрометации от кол-ва Мориарти на графике 3.

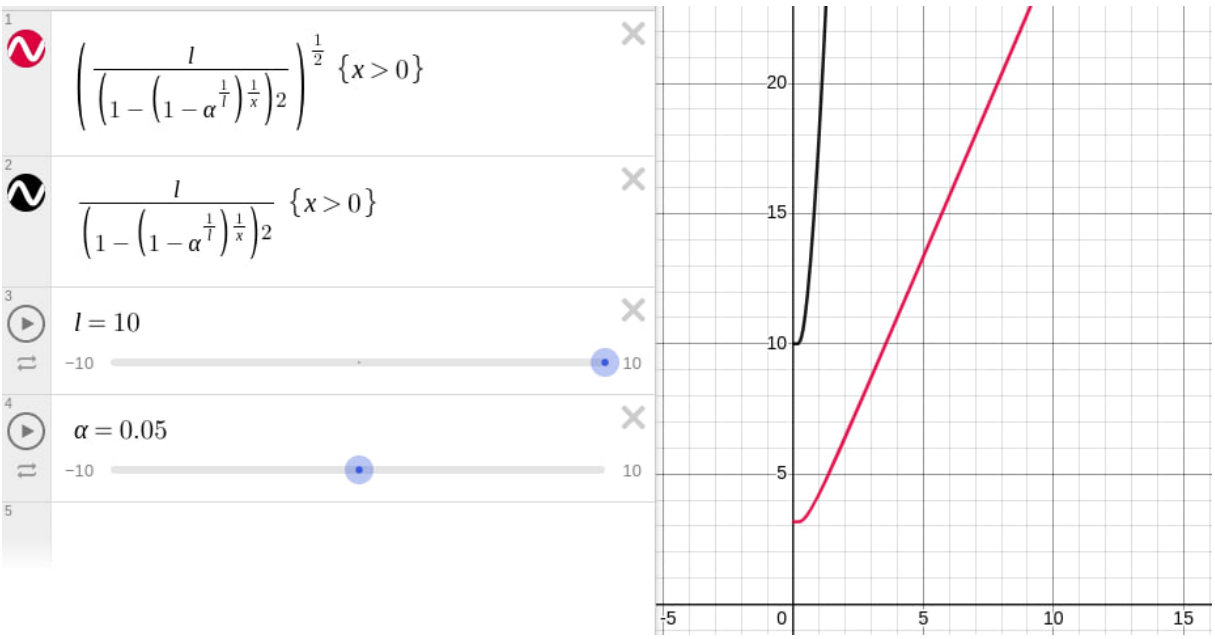

Рис. 2: По оси X - кол-во Мориарти. По оси Y - кол-во необходимых файлов в природе для черного графика и корень этой величины для красного графика. Кол-во файлов для каждого пользователя можно высчитать как $\sqrt{Ml}$, т.е. зависит линейно от кол-ва Мориарти.

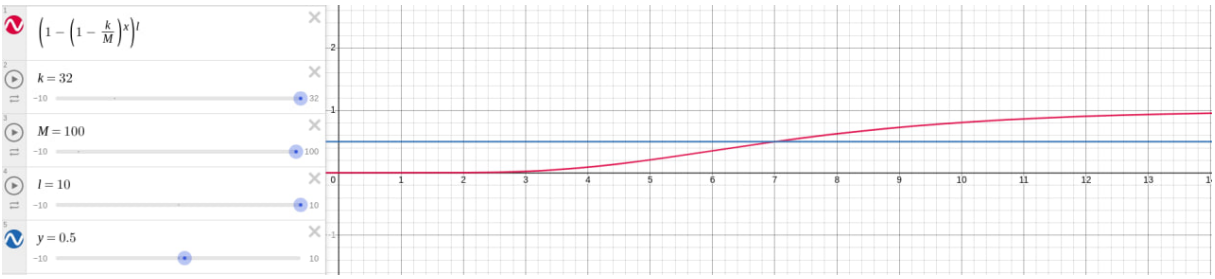

Рис. 3: По оси X - кол-во Мориарти. По оси Y - вероятность компрометации соединения. Видим, что даже при наличии 100 файлов, синдикат из 7 Мориарти может прослушать канал с вероятностью 50%.

2.4 Вывод

Найдем количество необходимых файлов по следующей формуле:

$$M = \frac{l}{(1 - \sqrt[q]{1 - \sqrt[l]{\alpha}})^2}$$

А количество файлов для каждого пользователя вычисляется так:

$$k = \sqrt{Ml} = \frac{l}{1 - \sqrt[q]{1 - \sqrt[l]{\alpha}}}$$

Где q - кол-во Мориарти, l - ожидаемое кол-во общих файлов для пары пользователей, α - допустимая вероятность компрометации.

3 Ближе к практике

На практике, хоть и возможно распространить файлы в случайном порядке, но это будет вынужден делать некий центр, который может быть скомпромитирован. Нам нужно добиться нужного распределения файлов децентрализованным путем. Давайте оценивать кол-во Мориарти как βN и попробуем привести алгоритм для поддержания M и k на нужном уровне. Пусть каждый юзер привнесет свои x файлов с белым шумом. Тогда $M = N \cdot x$. Из этого делаем вывод, что каждый новый пользователь должен привнести:

$$x = \frac{l}{N \cdot (1 - \sqrt[\beta N]{1 - \sqrt[l]{\alpha}})^2}$$

На графике 4 можно посмотреть как растет это кол-во с увеличением числа пользователей.

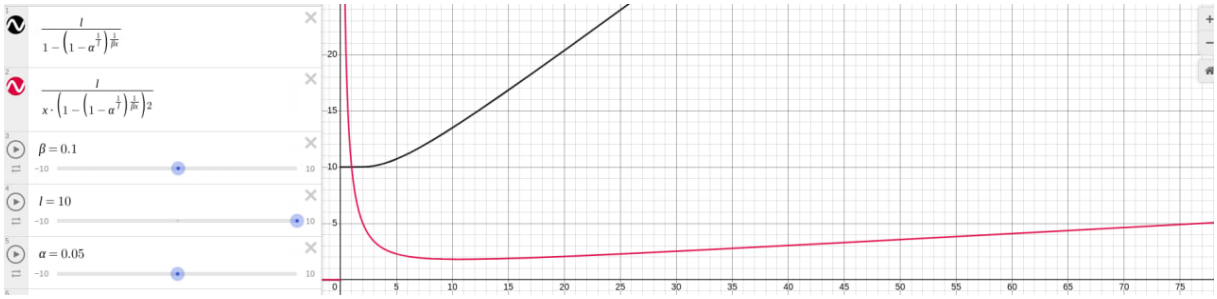

Рис. 4: По оси X - кол-во пользователей. По оси Y - кол-во файлов. Красная функция показывает сколько должен привнести файлов каждый участник (заметим что речь о единицах). Черная функция показывает количество файлов на пользователя. Делаем предположение, что 10% пользователей - Мориарти.

Теперь нужно придумать алгоритм перемешивания файлов для удовлетворения второго ограничения: у каждого пользователя должно быть ровно k файлов. Давайте развернем задачу и посмотрим на нее со стороны файла и пользователя. Чем больше у файла пользователей, тем он полезнее для общения, но менее безопасн. Чем больше пользователь распространяет файл тем больше он себя потенциально компромитирует.

Давайте посчитаем сколько в среднем пользователей $\hat{N}$ обладают файлом a_i.

$$\mathbb{E}\,\hat{N} = N \cdot P(a_i \in A) = N \cdot \frac{k}{M} = N(1 - \sqrt[q]{1 - \sqrt[l]{\alpha}})$$

Посмотрим на эту функцию на графике 5. Видим, что порядка 10 владельцев для группы людей.

В связи с этим напрашивается алгоритм. Пусть каждый пользователь отправит каждый свой изначальный файл $\mathbb{E}\,\hat{N} - 1$ случайным пользователям. Тогда у каждого в среднем будет по k файлов.

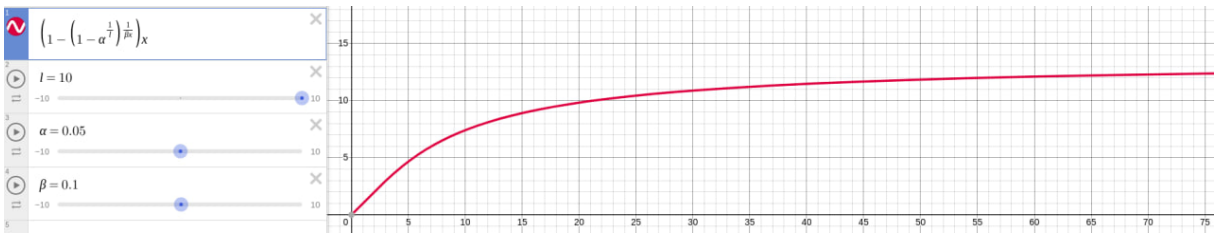

Рис. 5: По оси X - кол-во пользователей. По оси Y - ожидаемое кол-во пользователей владеющих файлом.

3.1 Итог

Всего у нас M файлов которые случайно разбросаны между участниками и в среднем у человека k файлов. Этих условий достаточно, чтобы соблюдались инварианты на ожидаемое кол-во общих файлов у случайной пары и на низкую вероятность компрометации.

3.2 Алгоритм

1. Каждый пользователь генерирует $\frac{l}{N\cdot(1-\sqrt[\beta N]{1-\sqrt[l]{\alpha}})^2}$ файлов.
2. Каждый файл раздается $N(1-\sqrt[\beta N]{1-\sqrt[l]{\alpha}})-1$ случайным пользователям.

Где N - кол-во пользователей. β - доля Мориарти. α - допустимая вероятность компрометации, l - ожидаемое кол-во общих файлов для пары пользователей.

4 Ближе к полевым условиям

В предыдущей секции мы сделали предположение, что пользователи могут попарно встречаться, что в общем случае не так. Давайте представим поле боя в котором связисты могут передавать флешки только соседним батальонам. Вполне разумно предположить, что такой граф является планарным.

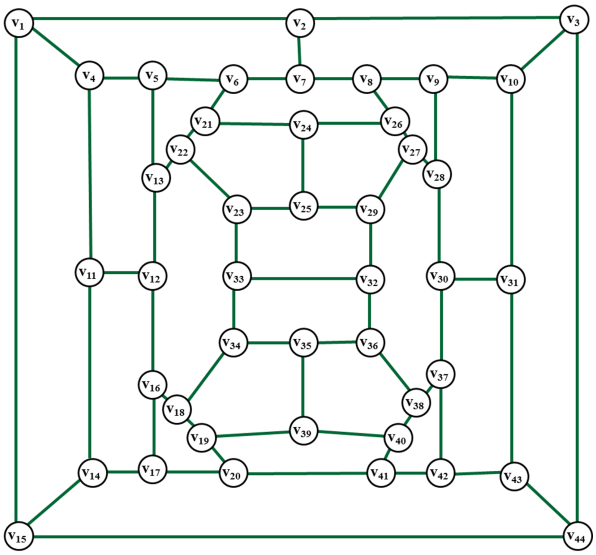

Рис. 6: Планарный граф

4.1 Постановка задачи

У каждого связиста есть $\frac{l}{N\cdot(1-\sqrt[\beta N]{1-\sqrt[l]{\alpha}})^2}$ файлов, которые можно передавать по цепочке $N(1-\sqrt[\beta N]{1-\sqrt[l]{\alpha}})-1$ раз. Когда у двух связистов появляется достаточно общих файлов, то в графе появляется ребро между ними. Требуется предложить алгоритм при котором граф станет станет максимально связным. Связностью графа назовем величину обратную среднему расстоянию между всеми парами вершин.

4.2 Жадный алгоритм

Предлагается решать чуть другую задачу: давайте для каждой вершины минимизировать путь до самой далекой от нее вершины. Для этого пронумеруем вершины в случайном порядке, где каждая вершина действует по порядку номера. Действе выглядит следующим образом:

1. Найти вершину u которая наиболее удаленная от данной v.
2. Передать очередной файл по кратчайщему пути от v до u.

Заметим, что файл может и не дойти до получателя, но при этом расстояние между вершинами в какой-то степени станет меньше. Если мы представим, что достаточно одного общего файла для проведения ребра, то передача файла через x вершин заменяет их всех на одну. Таким образом мы сжимаем наш граф в x раз, что дает нам экспоненициальное увеличение связности графа.

5 Рассуждения

В ходе исследования мы делали множество допущений и дали гарантии лишь в среднем. Основной результат в том, что мы предложили функциональную зависимость гиперпараметров M, k, x и $\hat{N}$ от свойств которыми мы хотим, чтобы наша модель обладала. Таким образом для увеличения безопасности соединения, стоит уменьшать α, а для повышения гарантий на то, что два пользователя могут пообщаться, стоит увеличивать l. Очевидно мы не можем сделать эти параметры запредельными, т.к. мы предполагаем, что устройства способны хранить лишь ограниченное количество файлов с белым шумом.