**Основы прикладной криптографии**

**Модуль 3. Криптографические атаки**

**Задание № 1**

*Докажите, что при неравномерном распределении вероятностей на множестве ключей криптосистемы минимум средней трудоёмкости метода полного перебора достигается при опробовании ключей в порядке убывания их вероятностей.*

**Решение:**

Пусть у нас есть криптосистема с множеством ключей K и вероятностями P(k), где k принимает значения из множества ключей K.

Средняя трудоемкость метода полного перебора ключей в криптосистеме определяется как сумма произведений вероятности P(k) на количество попыток, необходимых для взлома с использованием ключа k, для всех ключей k:

Средняя трудоемкость = Σ(P(k) \* k)

Где k - количество попыток, необходимых для взлома с использованием ключа K.

Допустим, что будем испытывать ключи в порядке убывания их вероятностей. Это означает, что мы начнем с ключа k1, который имеет наибольшую вероятность P(k1), затем перейдем к ключу k2 с следующей по величине вероятностью P(k2), и так далее, пока не переберем все ключи. Ключ k1 имеет наибольший шанс быть правильным ключом, следовательно вероятность успеха при первой попытке высока, а значит и Трудоемкость(k1) будет относительно низкой.

Затем мы переходим к ключу k2 с меньшей вероятностью P(k2). Вероятность успеха для этого ключа ниже, поэтому Трудоемкость(k2) будет выше, чем для k1, но это все равно меньше, чем для остальных ключей. И так далее, по мере продвижения к менее вероятным ключам, Трудоемкость для каждого из них будет увеличиваться.

Поскольку мы опробовали ключи в порядке убывания их вероятностей, мы начали с ключа с наибольшей вероятностью успеха и постепенно переходили к менее вероятным ключам. Это минимизирует среднюю трудоемкость, так как мы начали с более легких ключей и увеличивали ее по мере продвижения к менее вероятным ключам. Следовательно, при неравномерном распределении вероятностей на множестве ключей криптосистемы минимум средней трудоемкости метода полного перебора достигается при опробовании ключей в порядке убывания их вероятностей.

**Задание № 2**

*Временная сложность дешифрования криптосистемы на момент разработки в 2023 году оценена: а) в 100 лет, б) в 1000 лет.*

*Что нужно сделать: Определить, сколько лет в соответствии с законом Мура время дешифрования криптосистемы не превысит года.*

**Решение:**

Исходно:

t\_начальное = 100 лет = 100 \* 365 дней = 36,500 дней

Согласно закону Мура, количество транзисторов на микросхеме удваивается каждые 18-24 месяца. Мы можем взять средний период удвоения, например, 21 месяц, что составляет приблизительно 1.75 года.

Оценим, сколько раз максимальная производительность вырастет с 2023 года до того момента, когда время дешифрования сократится до 1 года:

Количество удвоений = log2(36,500 / 365) / log2(2)

Количество удвоений = 6.6

Это означает, что производительность увеличится примерно в 6.6 раз. Теперь, чтобы найти количество лет, которое потребуется для сокращения времени дешифрования до 1 года, мы делим 100 лет на это увеличение в 6.6 раз:

t\_год = t\_начальное / 6.6 => 100 лет / 6.6 = 15.15 лет

Таким образом, при условии соблюдения закона Мура и оценки времени дешифрования в 100 лет на момент разработки в 2023 году, потребуется примерно 15.15 лет для сокращения времени дешифрования до 1 года.

**Задание № 3**

*Оцените трудоёмкость реализации оперативного метода Хеллмана для симметричного блочного шифра с ключевым множеством порядка 264, если размер блоков данных равен 64 бита.*

Оперативный метод Хеллмана - это атака, которая пытается найти общий секретный ключ, путем перебора всех возможных ключей и сравнения результатов шифрования или дешифрования с зашифрованным текстом или открытым текстом.

Для оценки трудоемкости оперативного метода Хеллмана нам нужно учитывать следующие факторы:

Размер ключевого множества (K): У нас есть 2^64 возможных ключа.

Размер блока данных (N): У нас блок данных размером 64 бита.

Время на проверку каждого ключа: это время, которое занимает проверка одного ключа с использованием блочного шифра и сравнение полученного зашифрованного текста с известным.

Трудоемкость рассчитывается по формуле: Трудоемкость = (K \* N) / (2 \* t), где:

K - размер ключевого множества (2^64 в данном случае).

N - размер блока данных (64 бита).

t - время на проверку каждого ключа (в зависимости от вычислительной мощности).

Допустим, t - это время, которое занимает проверка одного ключа, и давайте предположим, что это занимает одну миллисекунду (0,001 секунды). Тогда трудоемкость можно оценить как:

Трудоемкость = (2^64 \* 64) / (2 \* 0.001) = 1,87\*10^16 лет

Это огромное время, и означает, что перебор 2^64 ключей с использованием оперативного метода Хеллмана на обычном компьютере не является выполнимой при разумных условиях.