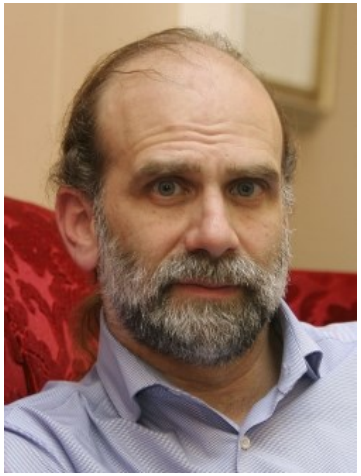
Криптоанализ

Лекция №10

Определение Криптоанализа



Существует только один путь стать хорошим разработчиком криптографических алгоритмов - быть хорошим криптоаналитиком и взламывать алгоритмы. Множество. Снова и снова. Только после того, как обучающийся продемонстрирует способности к криптоанализу чужих алгоритмов, он сможет серьезно браться за разработку собственных алгоритмов.

Брюс Шнайер (Bruce Schneier)

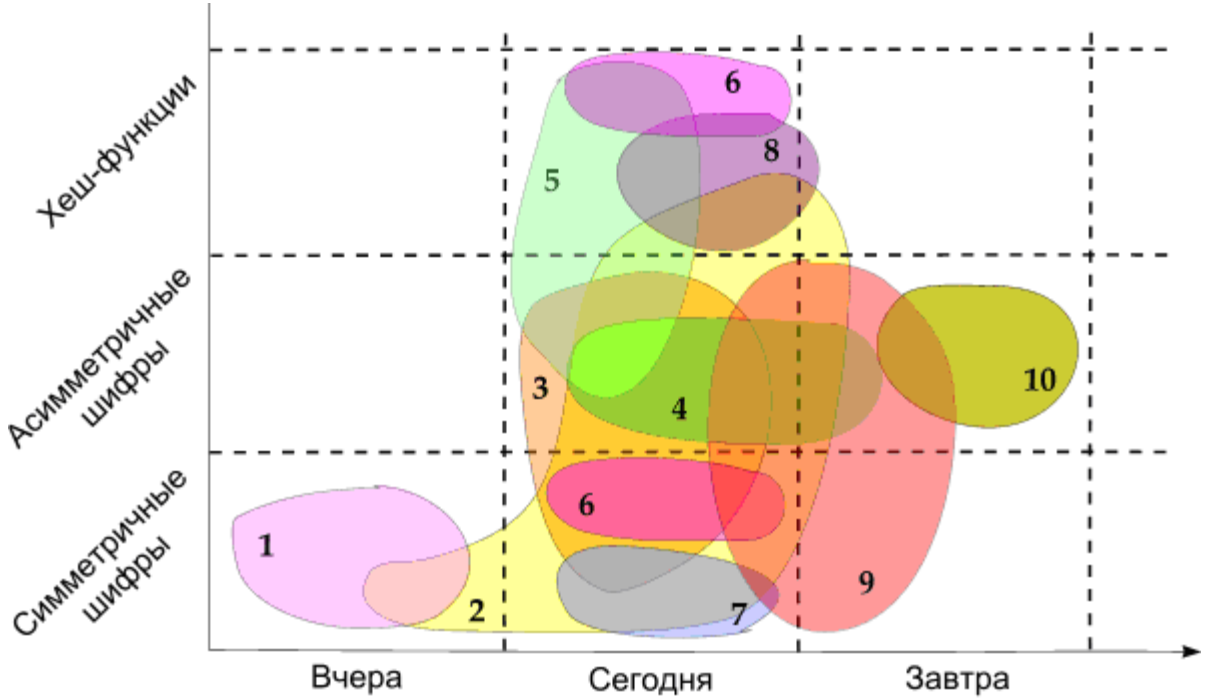
Криптоанализом (от греческого krypts - "скрытый" и analein - "ослаблять" или "избавлять") называют науку восстановления (дешифрования) открытого текста без доступа к ключу

Попытка криптоанализа называется **атакой**

Классификация исходов

- **Полный взлом** – криптоаналитик извлекает секретный ключ
- **Глобальная дедукция** – криптоаналитик разрабатывает функциональный эквивалент исследуемого алгоритма, позволяющий зашифровывать и расшифровывать информацию без знания ключа
- **Частичная дедукция** – криптоаналитику удается расшифровать или зашифровать некоторые сообщения
- **Информационная дедукция** – криптоаналитик получает некоторую информацию об открытом тексте или ключе

Методы криптоанализа



1 - частотный анализ
2 - полный перебор ключей
3 -анализ ключевого генератора
4 - факторизация/дискретное логарифмирование
5 - метод "встречи посередине"
6 - разностный анализ
7 - линейный анализ
8 - метод коллизий
9 - анализ по побочным каналам
10 - квантовый анализ

Виды криптоанализа

- Частотный анализ
- Дифференциальный криптоанализ
- Линейный криптоанализ
- Модификации дифференциального и линейного анализов
- Интерполяционный криптоанализ
- Методы, основанные на слабости ключевых разверток

Метод полного перебора

- **Распараллеливание**
  - Конвейер
  - Разбиение ключевого множества («Китайская лотерея», «DESозавр», Криптоаналитические водоросли)
- **Задача определения «осмысленности» выходных данных**
  - Априорная вероятность
  - Апостериорная вероятность

Кол-во знаков	Кол-во вариантов	Стойкость	Время перебора
1	36	5 бит	менее секунды
2	1296	10 бит	менее секунды
3	46 656	15 бит	менее секунды
4	1 679 616	21 бит	17 секунд
5	60 466 176	26 бит	10 минут
6	2 176 782 336	31 бит	6 часов
7	78 364 164 096	36 бит	9 дней
8	$2{,}821\ 109\ 9 \times 10^{12}$	41 бит	11 месяцев
9	$1{,}015\ 599\ 5 \times 10^{14}$	46 бит	32 года
10	$3{,}656\ 158\ 4 \times 10^{15}$	52 бита	1 162 года
11	$1{,}316\ 217\ 0 \times 10^{17}$	58 бит	41 823 года
12	$4{,}738\ 381\ 3 \times 10^{18}$	62 бита	1 505 615 лет

Китайская лотерея

- Представьте, что микросхема, вскрывающая алгоритм грубой силой со скоростью миллион проверок в секунду, встроена в каждый проданный радиоприемник и телевизор. Каждая микросхема запрограммирована для автоматической проверки различного набора ключей после получения пары открытый текст/шифротекст по эфиру
- Каждый раз когда китайское правительство хочет раскрыть ключ, оно передает исходные данные по радио. Все радиоприемники и телевизоры в стране начинают считать. В конечном счете, правильный ключ появляется на чьем-нибудь дисплее
- Китайское правительство платит приз тому человеку - это гарантирует, что результат будет сообщен быстро и правильно, и также способствует рыночному успеху радиоприемников и телевизоров с микросхемами вскрытия
- Если у каждого человека в Китае, будь то мужчина, женщина или ребенок, есть радиоприемник или телевизор, то правильное значение 56-битового ключа появится через 61 секунду. Если радиоприемник или телевизор есть только у каждого десятого китайца(что близко к действительности), то правильный ключ появится через 10 минут. Правильный 64-битовый ключ будет раскрыт через 4.3 часа (43 часа, если радиоприемник или телевизор есть только у каждого десятого китайца)

Частотный анализ

- Вероятности появления отдельных букв, а также их порядок в словах и фразах естественного языка подчиняются задокументированным статистическим закономерностям
- например, пара стоящих рядом букв "ся" в русском языке более вероятна, чем "цы", а "оь" не встречается никогда

Буква	Частота	Буква	Частота	Буква	Частота	Буква	Частота
О	0.09	в	0.038	з	0.016	ж	0.007
е, ё	0.072	л	0.035	ы	0.016	ш	0.006
А	0.062	к	0.028	б	0.014	ю	0.006
И	0.062	м	0.026	ь,ъ	0.014	ц	0.004
Н	0.053	д	0.025	г	0.013	щ	0.003
Т	0.053	п	0.023	ч	0.012	э	0.003
С	0.045	у	0.021	й	0.01	ф	0.002
Р	0.04	я	0.018	х	0.009		

Дифференциальный анализ

- Разработан в 1990 году израильскими криптографами **Эли Бихамом** (Eli Biham) и **Али Шамиром** (Ali Shamir)
- Выбираем пары входных текстов (**P**) с фиксированной разностью, смотрим, как отличаются шифры (**C**) от них:

$$\Delta \mathbf{P} = \mathbf{P_1} \oplus \mathbf{P_2} \quad \Delta \mathbf{C} = \mathbf{C_1} \oplus \mathbf{C_2}$$

- Анализируем множество таких пар и находим наиболее вероятный ключ

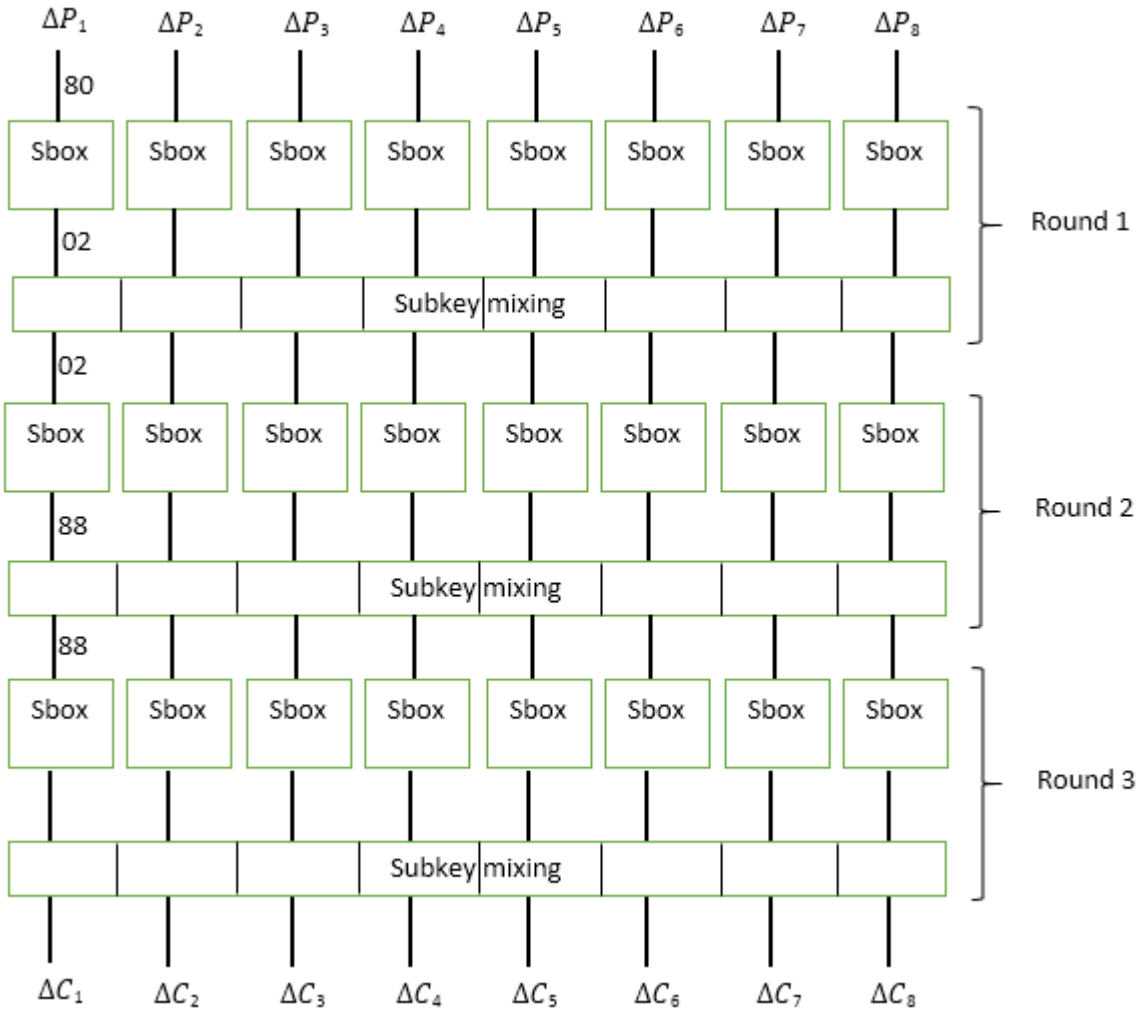
Эли Бихам

Али Шамир

Дифференциальный анализ

- Для двух заранее подобранных шифротекстов $\mathbf{P_1}$ и $\mathbf{P_2}$ злоумышленник вычисляется «дифференциал» $\mathbf{\Delta P = P_1 \oplus P_2}$
- С помощью **ΔP** пытается определить каким должен быть «дифференциал» шифротекстов $\mathbf{\Delta C = C_1 \oplus C_2}$
- Зачастую невозможно предугадать со 100% точностью какое именно будет иметь значение **ΔC**
- Единственное, что может злоумышленник, это определить с какой частотой шифр возвращает различные значения **ΔC**, для заданного заранее **ΔP**
- Это знание позволяет атакующему вскрыть часть ключа или ключ целиком

Дифференциальный анализ



Дифференциальный анализ

- Предположим, что злоумышленник решил проверить дифференциал **0x80**
- Для этого он генерирует произвольный байт $\mathbf{X_1}$, и вычисляет $\mathbf{X_2=X_1 \oplus 80}$
- Далее атакующий прогоняет $\mathbf{X_1}$ и $\mathbf{X_2}$ через функцию **Sbox** и получает значения $\mathbf{Y_1}$ и $\mathbf{Y_2}$
- Для каждой такой пары $\mathbf{X_1}$ и $\mathbf{X_2}$, дифференциал которых равен 80, атакующий в состоянии получить дифференциал **ΔY**
- Анализируя полученные значения, атакующий выбирает такое значение **ΔY**, которое имеет большую вероятность возникновения
- Допустим, что из всех 256 пар $\mathbf{X_1}$ и $\mathbf{X_2}$, в 192 случаях $\mathbf{Y_1 \oplus Y_2=02}$. Таким образом, вероятность того, что при заданном **ΔX=80**, значение **ΔY=02**, составляет 192/256=**3/4**
- Это означает, что при заданном **ΔX=80**, с вероятностью $\mathbf{P_1=3/4}$ на вход второго раунда попадут два значения $\mathbf{U_1}$ и $\mathbf{U_2}$, такие что **ΔU=02**

Дифференциальный анализ

- Для раскрытия свойств второго раунда, злоумышленник генерирует новые 256 пар входных байт $\mathbf{X_1}$ и $\mathbf{X_2}$, таких, что $\mathbf{X_1 \oplus X_2 = 02}$
- Произведя вычисление функции **Sbox** для каждой пары $\mathbf{X_1}$ и $\mathbf{X_2}$, атакующий замечает, что в 64 случаях из 256 $\mathbf{\Delta Y = 88}$, т.е. вероятность того, что $\mathbf{\Delta Y = 88}$, для заданного $\mathbf{\Delta X = 02}$, составляет $\mathbf{P_2} = 64/256 = \mathbf{1/4}$
- Таким образом, произведя нехитрый подсчет вероятностей, атакующий понимает, что для указанного шифра для каждой пары байт $\mathbf{X_1}$ и $\mathbf{X_2}$, таких что $\mathbf{\Delta X = 80}$, с вероятностью $\mathbf{P} = \mathbf{P_1} * \mathbf{P_2} = 3/4 * 1/4 = \mathbf{3/16}$, дифференциал внутреннего состояния шифра перед последним раундом составляет $\mathbf{\Delta Y = 88}$
- Обладая этим знанием атакующий генерирует несколько пар текстов таких, что $\mathbf{\Delta P = 80808080808}$ и приступает к побайтовому подбору подключа третьего раунда

Дифференциальный анализ

Недостатки Дифференциального анализа:

- С увеличением числа раундов сложность криптоанализа увеличивается, однако остаётся меньше сложности полного перебора при количестве циклов меньше 16
- Метод требует большого объема памяти для хранения возможных ключей

Зависимость от количества раундов [скрыть]	
Число раундов	Трудоёмкость атаки
4	2^{4}
6	2^{8}
8	2^{16}
9	2^{26}
10	2^{35}
11	2^{36}
12	2^{43}
13	2^{44}
14	2^{51}
15	2^{52}
16	2^{58}

Линейный анализ

- Разработан в 1993 году **Митцуру Матцуи**
- Основан на поиске линейных зависимостей между исходным текстом, шифротекстом и ключом
- Подробное описание: https://habr.com/ru/post/233905/

Митцуру Матцуи

Интерполяционный анализ

- Предложен в 1997 году **Т.Джекобсеном** и **Л.Кнудсеном**
- Предполагает, что раундовая функция – многочлен, тогда весь шифр может быть записан как многочлен, коэффициенты которого зависят от ключа
- Далее многочлен интерполируется по большому количеству исходных текстов:

$$f(x) = \sum_{i=1}^{n} y_i \prod_{1 \le j \le n, j \ne i} \frac{x - x_j}{x_i - x_j}$$

- Подробное описание: https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B0%D1%82%D0%B0%D0%BA%D0%B0

Атака по ключам

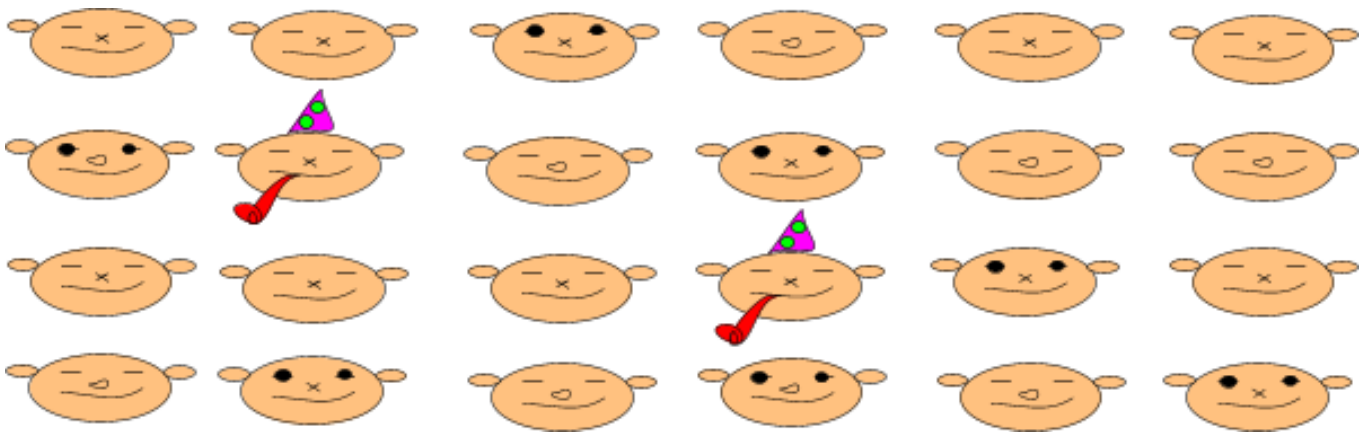
- **Фундаментальное допущение Кирхгоффа:** секретность сообщения всецело зависит от ключа, предполагается, что весь механизм шифрования, кроме значения ключа, известен противнику
- **Слабый ключ** – это ключ, не обеспечивающий достаточного уровня защиты или использующий в шифровании закономерности, которые могут быть взломаны
- **Генераторы псевдослучайных чисел** – слабое место многих криптосистем

Методы подбора паролей

- неоптимизированный перебор
- перебор, оптимизированный по словарям вероятных паролей
- перебор, оптимизированный на основе встречаемости символов и биграмм
- перебор, ориентированный на информацию о подсистеме аутентификации ОС
- перебор с использованием знаний о пользователе

Парадокс дней рождения

- Если считать, что дни рождения распределены равномерно, то в группе из 23 человек с вероятностью 0,5 у двух человек дни рождения совпадут.



Парадокс дней рождения

- Рассчитаем вероятность того, что в группе из 365 человек дни рождения всех людей будут различными
- Возьмём наугад одного человека из группы и запомним его день рождения. Затем возьмём наугад второго человека, при этом вероятность того, что у него день рождения не совпадёт с днем рождения первого человека, равна $1 - \frac{1}{365}$
- Затем возьмём третьего человека; при этом вероятность того, что его день рождения не совпадёт с днями рождения первых двух, равна $1 - \frac{2}{365}$

Парадокс дней рождения

- Перемножая все эти вероятности, получаем вероятность того, что все дни рождения в группе будут различными:

$$\bar{p}(n) = \left(1 - \frac{1}{365}\right) \cdot \left(1 - \frac{2}{365}\right) \cdot \ldots \cdot \left(1 - \frac{n-1}{365}\right) = \frac{365 \cdot 364 \cdot \ldots \cdot (365 - n + 1)}{365^n} = \frac{365!}{365^n (365 - n)!}$$

n	$p(n)$
10	12 %
20	41 %
22	47,57 %
23	50,73 %
30	70 %
50	97 %
100	99,99996 %
200	99,999999999998 %
365	100 %

Если $a\sqrt{b}$ предметов выбираются из некоторой совокупности размером b, то вероятность того что 2 из них совпадут равна:

$$1 - e^{-\frac{a^2}{2}}$$

Если b = 365, $a\sqrt{b}$ = 23, то a = 1,204

Метод «Встреча посередине»

- **Задача**: пусть нам нужно найти ключ K по известному открытому тексту **X** и криптограмме **Y**
- **Условие**: множество ключей криптоалгоритма замкнуто относительно композиции, то есть для любых ключей $\mathbf{K^1}$ и $\mathbf{K^2}$ найдется ключ **K** такой, что $\mathbf{E_{K2}(E_{K1},X) = E_K(X)}$
- **Решение**: Поиск ключа **K** можно свести к поиску пары эквивалентных ключей $\mathbf{K^1}$ и $\mathbf{K^2}$
- Для текста **X** построим базу данных содержащую случайное множество ключей $\mathbf{K^1}$ и соответствующих криптограмм

$$\mathbf{W = \{W_1, W_2, \ldots W_N\}},$$

где **N** – мощность множества ключей $\mathbf{K^1}$

Метод «Встреча посередине»

- Затем подбираем случайным образом ключи $\mathbf{K^2}$ для расшифровки текстов **Y** и результат расшифрования как $\mathbf{V} = \mathbf{E_{K2}}(\mathbf{Y})$
- Для каждого ключа $\mathbf{K^2}$ сравниваем **V** с **W**, которые хранятся в БД
- Если **V** = **W**, то ключ $\mathbf{K^1K^2}$ эквивалентен ключу **K**

Атаки на Хэш-функции

Коллизия: **H**(**M**)=**H**(**M’**), где

H – хеш-функция

M – исходное сообщение

M’ – сообщение, подобранное злоумышленником

Атака на обнаружение коллизий для n-битной хеш-функции:

Требует ~ $\mathbf{2^{n/2}}$ операций (парадокс дней рождения)

Анализ ассиметричных систем

- Криптоанализ систем шифрования, основанных на сложности задачи **дискретного логарифмирования**
- Криптоанализ систем шифрования, основанных на сложности задачи **факторизации**
- Задача дискретного логарифмирования вычислительно сложнее задачи разложения. Если будет найден полиномиальный алгоритм ее решения, станет возможным и разложение на множители (обратное не доказано)

Логарифм в конечном поле

- Вычислить дискретный логарифм числа b по основанию a в конечном поле Z_p означает найти

$$x = \log_a b \in Z, \text{ при котором } a^x = b$$

- **Пример:**

$$24^x = 19 \ (mod\ 37)$$

Найти X?

Алгоритм Адлемана: 1 Этап

- Сформировать факторную базу Q состоящую из всех простых чисел q:

$$Q = \left\{ q \leq B = e^{\sqrt{\log p \log \log p}} \right\}$$

- **Пример:**

$$24^x = 19 \ (mod\ 37)$$

Факторная база: 2, 3, 5 (для p = 37)

Алгоритм Адлемана: 2 Этап

- С помощью перебора найти натуральные числа r_i такие, что:

$$a^{r_i} \equiv \prod_{q \le B} q^{a_{iq}} \, mod \, p$$

- То есть a^{r_i} раскладывается по факторной базе

$$r_i \equiv \sum_{q \le B} a_{iq} \log_a q \, mod \, (p-1)$$

- **Пример:**

$$\begin{cases} 24^1 = 2^3 * 3 \, (mod \, 37) \\ 24^3 = 2^2 * 3 \, * 5 \, (mod \, 37) \\ 24^7 = 5 \, (mod \, 37) \end{cases} \rightarrow \begin{cases} 1 = 3 * \log_{24} 2 + \log_{24} 3 \, (mod \, 36) \\ 3 = 2 * \log_{24} 2 + \log_{24} 3 + \log_{24} 5 \, (mod \, 36) \rightarrow \\ 7 = \log_{24} 5 \, (mod \, 36) \end{cases}$$

$$\begin{cases} \log_{24} 2 = 5 (mod \, 36) \end{cases}$$

Алгоритм Адлемана: 3 Этап

- С помощью перебора найти одно значение r такое, что:

$$a^r \equiv \prod_{q \le B} q^{b_q} p_1 \dots p_k \bmod p$$

- То есть:

$$\log_a b = -r + \sum_{q \le B} b_q \log_a q \bmod (p-1)$$

- **Пример:**

$$b = 19 \to 24^1 * 19 = 12 = 2^2 * 3 \ (mod\ 37)$$
$$\log_{24} 19 = -1 + 2 * \log_{24} 2 + \log_{24} 3 = 31 \ (mod\ 36)$$

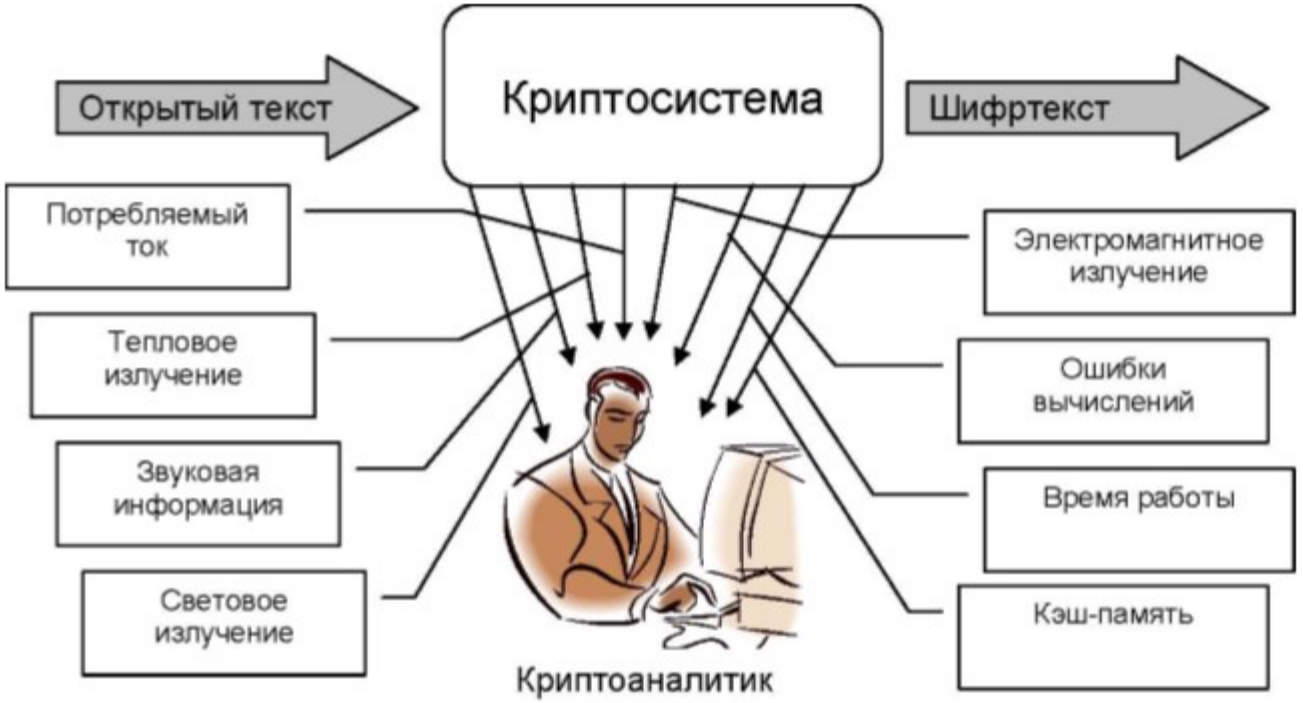
$$x = 31$$

Атаки по побочным каналам

- Используют информацию, которая может быть получена с устройства шифрования и не является при этом ни открытым текстом, ни шифртекстом

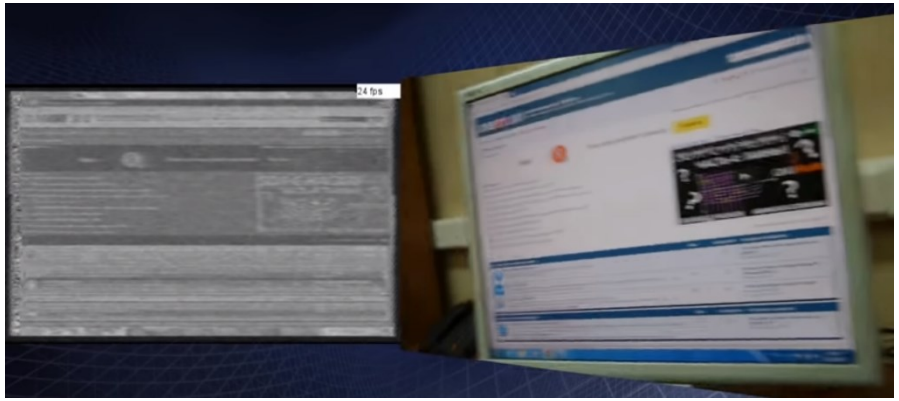
- Основаны на корреляции между значениями физических параметров, измеряемых в разные моменты во время вычислений, и внутренним состоянием вычислительного устройства, имеющим отношение
к ключу.

Атаки по побочным каналам



Пример атаки по побочным каналам

Software-defined radio - это когда все характеристики будущей радиоволны (например, её частота) создаются в компьютере, и затем такая «нарисованная» в компьютере волна отправляется на внешнее устройство, чтобы быть отправленной как настоящая радиоволна



https://youtu.be/PV_v1HgjN3Q