

Шифр Виженера

1 Шифр Виженера

Шифр Виженера это $(\Sigma^*, \Sigma^*, \Sigma^*, E, D)$

Открытый текст $m = m_1 m_2 \dots m_n$

Криптограмма $c = c_1 c_2 \dots c_n$

Ключ (какое-то слово) $k = k_1, \dots, k_p$, где $p \ll n$ (т.е. длина ключа p намного меньше длины открытого текста n)

Ключ циклически распространяем на весь открытый текст

$$c_i = (m_i + k_{[(i-1)\%p]+1}) \% |\Sigma|$$

$$m_i = (c_i - k_{[(i-1)\%p]+1}) \% |\Sigma|$$

2 Криптоанализ шифра Виженера

2.1 Атака №3 (классический вариант, когда буквы расположены в алфавитном порядке, А-0ая, Б-1ая и тд)

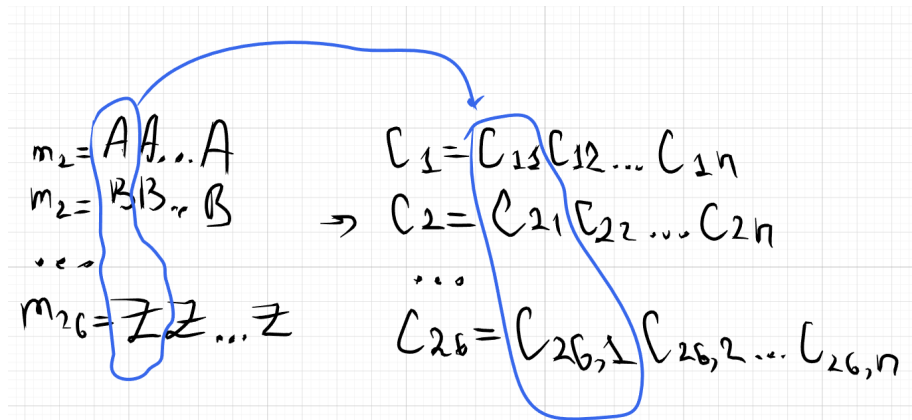
Шифруем длинную строку из букв А. Полученная криптограмма будет представлять собой строку вида ключключключ, т.к. у буквы А - номер 0 в алфавите.

2.2 Атака №3 (неклассический вариант, не можем пронумеровать буквы, т.к. не знаем их порядок)

По очереди шифруем длинные строки вида:

- $m_1 = \text{AAA}..A$
- $m_2 = \text{BBB}..B$
- и тд
- $m_{26} = \text{ZZZ}..Z$

получив строчки c_1, c_2, c_3 и тд c_{26} . Записываем эти строчки друг под другом. Таким образом мы знаем как действует i -ый символ ключа на i -ый столбик открытого текста.



2.3 Атака № 2 (классический вариант)

Есть пара:

$$m_1 = m_{1,1}m_{1,2}...m_{1,t} \quad c_1 = c_{1,1}c_{1,2}...c_{1,t}$$

$$m_2 = m_{2,1}m_{2,2}...m_{2,t} \quad c_2 = c_{2,1}c_{2,2}...c_{2,t}$$

и тд

Чтобы найти ключ можно посчитать разность, т.к

$$k_i = (c_i - m_i) \mod (|\Sigma|)$$

Т.е в теории достаточно даже одной пары

2.4 Атака № 2 (неклассический вариант)

Используя пары узнаем, куда i -ый символ ключа переводит i -ый символ открытого текста. Узнаем значительную часть ключа, а дальше перебор.

2.5 Вопрос как понять где начинается ключ, и его длину, ведь в общем случае пары могут быть зашифрованы разными фрагментами ключа

2.6 Атака №2'

Метод протяжки вероятного слова, подставляем слово в наиболее вероятную часть шифр-текста, а дальше вычисляем разность таким образом находим часть ключа.

Если это слово ключевое, то начинаем сдвигать его вправо и пробовать расшифровать шифр текст.

2.7 Атака № 1

Необходимо найти длину ключа.

$$\begin{array}{r}
 m_s m_{s+1} \dots m_{s+n} \\
 - c_s c_{s+1} \dots c_{s+n} \\
 \hline
 k_s k_{s+1} \dots k_{s+n}
 \end{array}$$

Рис. 1: Метод протяжки

3 Поиск длины ключа

3.1 Метод Казиски

Наиболее вероятная причина появления в шифре текста длинных повторяющихся частей это шифрование одинаковых частей открытого текста одинаковыми частями ключа.

Идея метода основана на том, что ключи являются периодическими, а в естественном языке существуют часто встречающиеся буквосочетания: биграммы и триграммы. Это наводит на мысль, что повторяющиеся наборы символов в шифротексте — повторения популярных биграмм и триграмм исходного текста.

Метод Казиски заключается в поиске групп символов, которые повторяются в зашифрованном тексте. **Группы должны состоять из не менее чем трех символов.** Тогда расстояния между последовательными возникновениями групп, вероятно, будут кратны длине ключевого слова. **Предполагаемая длина ключевого слова кратна наибольшему общему делителю всех расстояний.**

Причина, по которой метод работает — это то, что если две группы символов повторяются в исходном тексте и расстояние между ними является кратным длине ключевого слова, то буквы ключевого слова выровняются с обеими группами.

Пример:

3.2 Подсчёт числа совпадений

3.2.1 Определение

Пусть Σ - некоторый конечный алфавит. Тогда **индексом совпадения** для слова $w = w_1 w_2 \dots w_n$ называют

$$IC(w) = \sum_{i=1}^{|\Sigma|} \frac{F_i(F_i - 1)}{n(n - 1)}$$

There are two ways of constructing a software design:
 One way is to make it so simple that there are obviously
 no deficiencies, and the other way is to make it so complicated
 that there are no obvious deficiencies.
 The first method is far more difficult.

После удаления пробелов и пунктуации и преобразования в верхний регистр получается следующее:

THERE ARETW OWAYS OFCON STRUC TINGA SOFTW AREDE SIGNO NEWAY
 ISTOM AKEIT SOSIM PLETH ATTHE REARE OBVIO USLYN ODEFI CIENC
 IESAN DTHEO THERW AYIST OMAKE ITSOC OMPLI CATED THATT HEREA
 RENOO BVIUO SDEFI CIENC IESTH EFIRS TMETH ODISF ARMOR EDIFF
 ICULT

Затем полученный текст шифруется с помощью 6 буквенного ключевого слова **SYSTEM** следующим образом:

LFWKI MJCLP SISWK HJOGL KMGVU RAGKM KMXMA MJCVM WUYLG GIISW
 ALXAE YCXMF KMKBQ BDCLA EFLFW KIMJC GUZUG SKECZ GBWYM OACFV
 MQKYF WXTWM LAIDO YQBWF GKSDI ULQGV SYHJA VEFWB LAEFL FWKIM
 JCFHS NNGGN WPWDA VMQFA AXWFZ CXBVE LKWML AVGKY EDEMJ XHUXD
 AVYXL

Рис. 2: Пример метода Казиски

Где:

- F_i - это частота встречаемости буквы w_i в w

Если по-простому то индекс совпадений это доля пар совпадающих букв

Пример:

- $IC(\text{математика}) = \frac{3+1+1}{C_{10}^2} = \frac{1}{9}$

3.3 Применяем теорему из 4 билета об индексе совпадений в криптограмме, зашифрованной шифром Виженера

Для шифра Виженера мы выбираем такое $U \subseteq \mathcal{K}$ что:

- $i \sim j \iff p|(j - i)$ т.е отношение волна связывает все пары букв на расстоянии кратном длине ключа

Пусть для простоты $p|n$, т.е $n = p \cdot k$

Отношение \sim разбивает множество $\{1, 2, \dots, n\}$ на p классов, в каждом из которых по k элементов.

т.е разбиение $\sim = \{ \{1, p+1, 2p+1, \dots\}, \{2, p+2, 2p+2, \dots\}, \dots \{p, 2p, 3p, \dots\} \}$ буквы в одном классе шифруются одной и той же буквой ключа

$| \sim | n = p \cdot k \cdot (k - 1) =$ где:

- p - кол-во классов

- k - кол-во элементов в классе
- k-1 - кол-во пар в одном классе

Тогда $|\sim| = n(n-1) - pk(k-1) =$

Преобразуем выражения, сделав замену $[k = \frac{n}{p}]$:

$$|\sim| = n(n-1) - \frac{n(n-p)}{p} = \frac{n^3(p-1)}{p}$$

⊗ можем преобразовать ф-лу из теоремы ⊗

$$|\sim| = n \cdot \frac{1}{n(n-1)} \cdot \sum_{i \in \Sigma} p_i^2 + |\sim| \cdot \frac{1}{n(n-1)} \cdot \frac{1}{|\Sigma|} = \frac{n-p}{p(n-1)} \cdot \sum_{i \in \Sigma} p_i^2 + \frac{n(p-1)}{p(n-1)} \cdot \frac{1}{|\Sigma|}$$

3.4 Замечание о $\sum_{i=0}^{l-1} p_i$

Таблица распределения букв открытого текста:

m_i	0	1	2	...	$ \Sigma -1$
P	P_0	P_1	P_2		$P_{ \Sigma -1}$

Рис. 3: Таблица распределения букв

Тогда $\sum_{i=0}^{|\Sigma|-1} p_i = 1$ т.к. таблица распределения

$$\frac{1}{|\Sigma|} = \sum_{i=0}^{|\Sigma|-1} \left(\frac{1}{|\Sigma|} \right)^2$$

- это сумма квадратов для равномерного распределения

$$\frac{1}{|\Sigma|} < \sum_{s \in \Sigma} P(M=s)^2$$

4 1ый метод Фридмана

- работает только для малых p не более 5

Пусть C - это шифртекст.

1. Вычисляем IC(c)
2. Составляем таблицу из

$$\frac{n-p}{p(n-1)} \sum_{i \in \Sigma} p_i^2 + \frac{n(n-1)}{p(n-1)} \cdot \frac{1}{|\Sigma|}$$

для разных p

3. Сравниваем IC(c) и суммы из таблицы

5 2ой метод Фридмана

Строим для каждого d ($d \in \{2, 3, \dots, p\}$) разбиение криптограммы на подмножества:

- $C^1 = \{c_1, c_{d+1}, c_{2d+1}, \dots\}$

- $C^2 = \{c_2, c_{d+2}, c_{2d+2}, \dots\}$

- и тд

- $C^d = \{c_d, c_{2d}, c_{3d}, \dots\}$

если $p|d$, то $I(C^1) = \sum p_i^2$,

- т.к буквы в разбиении шифруются одним и тем же символом ключа

если $p \nmid d$, то $I(C^1) = \sum p_i^2$ находится между $\frac{1}{|\Sigma|}$ и $\sum p_i^2$

Можно вычислить $I(C^i) \forall i \in \{1, 2, 3, \dots, d\}$, а потом вычислить среднее, чтобы получить более точный результат

Постепенно увеличивая d и считая $IC(C)$, мы должны заметить скачок $IC(C)$ до табличного значения $\sum p_i^2$ для выбранного языка. Это признак того, что мы добрались до истинной длины ключа.

Для русского языка ($e = \ddot{e}$):

	$\frac{1}{ \Sigma }$	$\sum p_i^2$	
Рус. яз	0,03125	0,055	
Англ. яз	0,038	0,068	

Рис. 4: Таблица для русского языка

- Эта штука всегда работает

После того как мы нашли длину ключа, то в каждой из последовательностей:

- $C^1 = \{c_1, c_{d+1}, c_{2d+1}, \dots\}$

- $C^2 = \{c_2, c_{d+2}, c_{2d+2}, \dots\}$

- и тд

- $C^d = \{c_d, c_{2d}, c_{3d}, \dots\}$

необходимо провести частотный анализ, т.к по существу каждая из них зашифрована шифром Цезаря.

После частотного анализа мы узнаем чему равна i -ая буква ключа.

6 Посмотри как ломать Виженера, зная длину ключа