Душкин Роман Викторович
Директор по науке и технологиям искусственной интеллектуальной системы для персональной медицины «Джейн», автор курса по основам искусственного интеллекта (ai101.ru).

Шифр полиалфавитной замены

Статья продолжает цикл по криптографии и криптоанализу (см. «Потенциал», № 9 за 2017 г. и № 1 за 2018 г.) и рассматривает шифр полиалфавитной, или многоалфавитной, замены. Полиалфавитная замена – это другое развитие шифров одноалфавитной замены, которое требует линейного увеличения количества символов, используемых в шифрограммах, по сравнению с N-граммными шифрами, требующими степенного увеличения.

Мы переходим к третьей статье нашего цикла по криптографии и криптоанализу, который запущен в журнале «Потенциал» и будет состоять из значительного количества статей – вплоть до квантовых методов шифрования и абсолютно надёжных схем. Если вы читаете эту статью, не ознакомившись с предыдущими двумя, то категорически советую вам это сделать, так как в самом цикле имеется внутренняя логика и статьи необходимо читать последовательно.

Наша сегодняшняя статья посвящена новой системе шифрования, которая называется «полиалфавитная замена» или, если использовать русские корни в научных словах, – «многоалфавитная замена». Эта система шифрования так же, как и шифрование N-граммами, изученное в предыдущей статье цикла, вышло из шифра одноалфавитной, или простой замены, который мы изучали в самой первой статье. Это можно изобразить при помощи диаграммы, представленной на рис. 1.

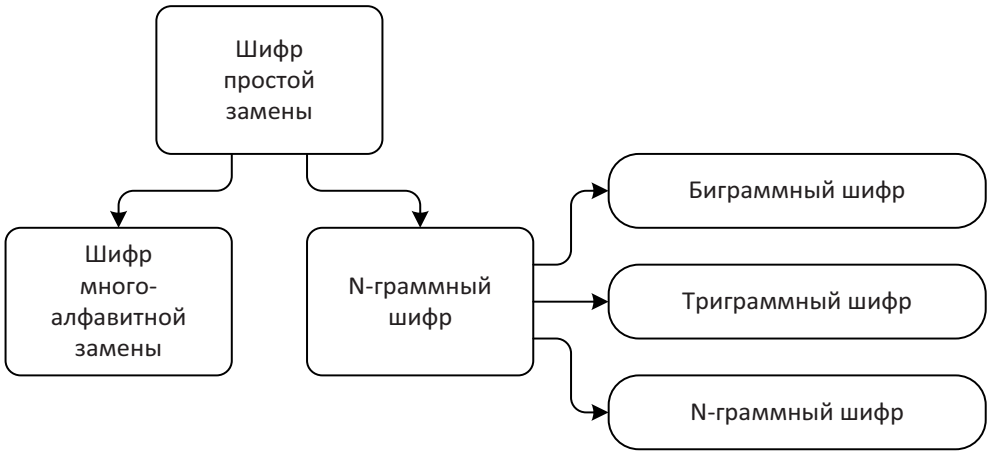

Рис. 1

Несмотря на то, что это тоже не очень-то надёжный метод шифрования, он открывает дорогу к абсолютно надёжной системе – одноразовому блокноту, свойства которого доказаны теоретически (и мы рассмотрим эту систему шифрования в одной из следующих статей). Также имеет смысл изучить многоалфавитные шифры, чтобы иметь представление о том, как их использовать и как взламывать.

Метод шифрования

Итак, шифр полиалфавитной, или многоалфавитной замены. Уже само название подсказывает то, как при помощи него шифровать сообщения – для этого используется несколько алфавитов для замены символов. Но как? Что если сначала каждый символ исходного сообщения зашифровать при помощи одного алфавита, выполнив подстановку, а потом уже зашифрованное сообщение прогнать через второй алфавит, сделав вторую подстановку? Даст ли это новые возможности по защите информации?

Совсем нет. Если вспомнить первую статью цикла, то процесс шифрования описывается при помощи функции, которой на вход подаётся строка символов (исходное сообщение) и которая на выходе возвращает тоже строку символов (зашифрованное сообщение), при этом обратная функция расшифровки однозначна. Другими словами, шифрование и расшифровка – это прямая и обратная функции во взаимно-однозначном процессе:

$$m = D\big(E(m)\big),$$

где m – исходное сообщение (или его символ), E – функция шифрования (от англ. *Encrypt*), D – функция расшифровки (от англ. *Decrypt*).

В математике последовательное применение функций – это операция «композиция», которая обозначается символом $\circ$. Другими словами, приведённая выше формула может быть записана следующим образом: $m = (E \circ D)(m)$, но это не очень красиво. Ведь математика – это ещё и кра-

сота. Так что обычно просто записывают « $E \circ D$ », и в этой записи главное не путать, что сначала применяется первая функция, а потом вторая. Само собой разумеется, что эта операция некоммутативна: $E \circ D \neq D \circ E$. Но теперь понятно, что

$$I = E \circ D,$$

где I – это функция тождества, которая всегда возвращает свой аргумент, никогда не изменяя его. Эта ситуация иллюстрируется диаграммой на рис. 2.

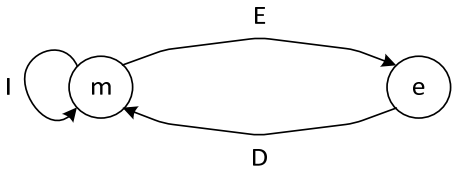

Рис. 2

А теперь давайте посмотрим, что будет, если применить процедуру шифрования два раза подряд. Если воспользоваться предложенной нотацией с операцией композиции функций, то запись будет выглядеть следующим образом:

$$m = (E_1 \circ E_2 \circ D_2 \circ D_1)(m),$$

или, что то же,

$$I = E_1 \circ E_2 \circ D_2 \circ D_1.$$

В виде диаграммы приведённая формула иллюстрируется на рис. 3.

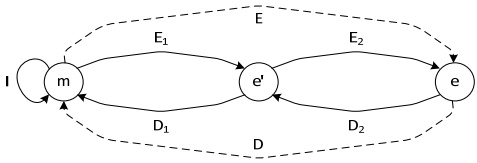

Рис. 3

Интерес вызывает то, что операция композиции является ассоциативной, так что в приведённой ранее формуле можно произвольным образом расставить скобки. На диаграмме рис. 3 видно, как были расставлены скобки и соответствующим образом переименованы «стрелки»:

$$I = (E_1 \circ E_2) \circ (D_2 \circ D_1) = E \circ D,$$

где $E = E_1 \circ E_2$, $D = D_2 \circ D_1$.

Несомненно, описанная схема подходит для любых методов шифрования (ведь в символах E, E_1, E_2, D, D_1, D_2 нигде не указаны методы шифрования и расшифровки, а потому это вполне универсальная схема (и мы с ней ещё столкнёмся в дальнейших статьях цикла). Однако если рассматривать шифр одноалфавитной замены, то надо учесть, что он работает не со всем сообщением, а посимвольно, т. е. все функции шифрования и расшифровки получают отдельные символы и преобразуют их в другие символы, причём это преобразование взаимно-однозначное. Из этого как раз и следует, что последовательное применение двух методов шифрования одноалфавитной замены тождественно применению одного такого метода с соответствующим изменением ключа. Это так же можно проиллюстрировать схемой на рис. 4.

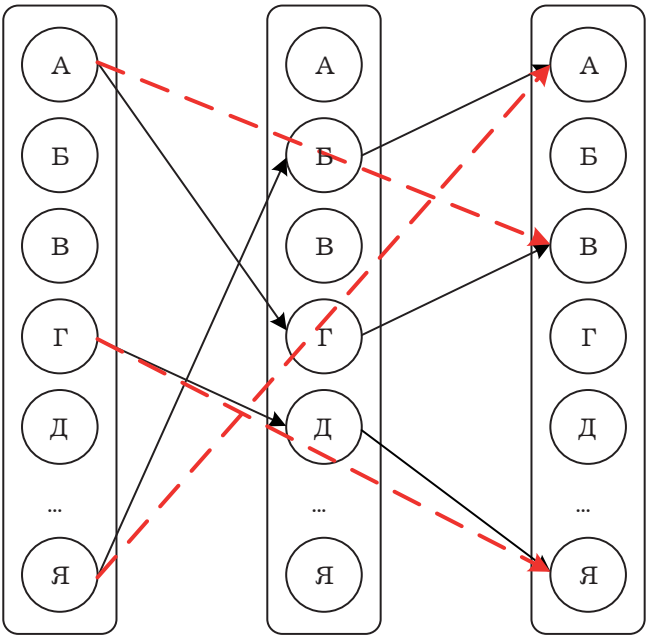

Рис. 4

Что ж, мы изучили математические основы (и не только рассматриваемого метода шифрования, но и кое-чего пошире – мы немного, совсем чуть-чуть прикоснулись к *теории категорий*), так что теперь перейдём к самому методу шифрования полиалфавитной заменой. Поскольку мы разобрались, что применять разные алфавиты несколько раз к одной и той же букве смысла нет, остаётся другой способ – применять разные алфавиты к разным буквам. Как такое возможно? Давайте рассмотрим на простом примере двухалфавитного шифра.

Пусть есть два алфавита. Для удобства мы будем считать, что шифрование всегда производится символами того же алфавита, которым написано и открытое сообщение. Как вы должны были понять из первой статьи цикла, вид значков для шифровки не имеет никакого значения, так что проще всего использовать произвольную перестановку символов того же самого алфавита для шифрования. Так вот, пусть есть две перестановки исходного алфавита. Например, такие:

Исх.	**А**	**Б**	**В**	**Г**	**Д**	**Е, Ё**	**Ж**	**З**	**И**	**Й**	**К**	**Л**	**М**	**Н**	**О**	**П**
1	П	О	Н	М	Л	К	Й	И	З	Ж	Е	Д	Г	В	Б	А
2	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	Т	С	Р
Исх.	**Р**	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ъ**	**Ы**	**Ь**	**Э**	**Ю**	**Я**
1	Я	Ю	Э	Ь	Ы	Ъ	Щ	Ш	Ч	Ц	Х	Ф	У	Т	С	Р
2	П	О	Н	М	Л	К	Й	И	З	Ж	Е	Д	Г	В	Б	А

Шифрование при помощи такой таблицы проводится так. Все нечётные буквы исходного сообщения шифруются при помощи строки № 1, а все чётные – при помощи строки № 2. Например, шифрование фразы «КРИПТОГРАФИЯ ПРЕКРАСНАЯ НАУКА» приведёт к появлению такой шифрограммы:

ЕПЗРЭСМППЛЗА АПКХЯЮТПА ВЯЬХП

Расшифровка производится таким образом: нечётные буквы шифрограммы расшифровываются по строке № 1, только в обратную сторону, а чётные – по строке № 2. Попробуйте!..

Теперь расширим эту методику на произвольное число N. Если у нас есть N алфавитов, то N-алфавитный шифр получается при помощи циклического применения алфавитов к отрезкам открытого текста длины N. Если, к примеру, N = 5, то весь открытый текст разбивается на пятёрки символов, и к первому символу каждой пятёрки применяется первый алфавит, ко второму – второй, и так далее.

Было бы здорово, если бы каждый алфавит можно было получить быстро на основании какого-либо простого правила. Действительно, зачем выдумывать большое количество алфавитов, придумывать перестановки и т. д., если такие перестановки можно получить при помощи применения несложной процедуры? И таких процедур придумано несколько, парочку из них мы сейчас рассмотрим.

Использование сдвигового шифра в качестве каждого алфавита

Такой шифр определяется той буквой, которая ставится вместо буквы «А». Например, сдвигаем на +3 («шифр Цезаря»), и тогда вместо буквы «А» используется буква «Г» (вместо «Б» – «Д» и т. д.). То есть алфавит для сдвига на +3 определяется буквой «Г». Соответственно все буквы алфавита определяют сдвиговый шифр, при этом шаг сдвига равен номеру буквы в алфавите без единицы. Это позволяет в качестве быстрого мнемонического правила использовать ключ – это или какое-либо слово, или же просто случайный набор букв. Например, если ключ – слово «СОЛНЦЕ», то он определяет 6-алфавитный шифр, первый алфавит которого – это сдвиг на +17, второй – сдвиг на +14 и т. д. Эта схема шифрования называется *шифром Виженера*.

Использование XOR-шифра

Поскольку мы можем каждую букву перевести в битовый код (например, 5-битный), к двум кодам двух букв можно применять операцию XOR («Исключающее ИЛИ»). Например, пусть буква «Б» кодируется 5-битным числом 00001, а буква «Г» – 5-битным числом 00011. Тогда Б $\oplus$ Г = 00001 $\oplus$ 00011 = 00010 = В. Поэтому, как и в предыдущем варианте, набор алфавитов может быть определён ключевым словом. В этом случае алфавит определяется при помощи применения операции XOR к

букве открытого текста и соответствующей букве ключа.

Что ж, на этом про шифрование всё. Так что перейдём к рассмотрению методов атаки на шифрограммы, зашифрованные методом многоалфавитной замены.

Метод атаки

Вообще говоря, одна из главных задач криптоаналитика заключается в определении того, какой шифр был применён при сокрытии сообщения. Как только ответ на этот вопрос получен, дальше дело техники – для многих шифровальных систем есть как минимум один метод атаки, гарантированно вскрывающий секретное сообщение. Не для всех, конечно, систем. Также существуют системы с теоретически доказанной невзламываемостью. Но о таких мы поговорим чуть позже.

Проблема в том, что методов однозначно определить тип шифровальной системы по виду шифрограммы, в общем-то, нет. Частотный анализ, к примеру, может помочь выявить шифр одноалфавитной замены – гистограмма частот символов в шифрограмме должна примерно совпадать с такой же гистограммой в принципе для текстов на используемом языке. Вместе с тем абсолютно такая же гистограмма будет у перестановочных шифров, взломать которые без ключа практически невозможно. Также и с многоалфавитными шифрами. Для них гистограмма частотности будет более плоской, чем для одноалфавитного шифра, и чем больше алфавитов используется, тем более равномерными по высоте будут столбцы гистограммы. Но такая же ситуация и с пропорциональным шифром, взломать который намного сложнее, чем многоалфавитный шифр. В общем, всё не так просто.

Но давайте сейчас считать, будто мы знаем, что полученная для криптоанализа шифрограмма зашифрована при помощи многоалфавитного шифра. Как подступиться к её взлому? Для этого надо ответить на два вопроса.

1. Какой длины ключ, при помощи которого зашифрован текст?

2. Каков сам ключ или, что то же самое, какие алфавиты использовались для замены?

Давайте попробуем ответить на первый вопрос. Здесь есть два пути. Во-первых, можно попробовать подобрать длину ключа, особенно если известно, что она небольшая (скажем, в пределах от 2 до 10). Это длинный и муторный путь, но он вполне возможен. Алгоритм подбора достаточно прост. Для выбранной длины ключа N осуществляется подсчёт частот для каждого из N алфавитов. Например, если N = 3, то считаются частоты для символов, стоящих на первом, втором и третьем местах каждой тройки. И если длина ключа действительно оказалась равной N (т. е. 3 в рассматриваемом примере), то все N гистограмм частот символов для каждого из N алфавитов должны быть похожи на гистограмму того языка, на котором написана шифрограмма. Долго, неудобно и не всегда помогает.

Во-вторых, есть более хитрый способ. Его предложил Фридрих Вильгельм Касиски в своём труде о шифровании и дешифровке, который был опубликован в 1863 году. Своей работой Касиски показал, что многоалфавитные шифры, которые

не поддавались взлому более 400 лет с момента их изобретения, не являются безопасными. А схема атаки на многоалфавитные шифры довольно проста и основана на том наблюдении, что в естественном языке часто повторяются буквосочетания и даже целые слова и фразы, и такие повторения вполне могут существовать и в шифрограмме. В свою очередь, это означает, что существует ненулевая вероятность того, что такие повторения будут зашифрованы одинаково – ключ несколько раз наложится на них с одним и тем же сдвигом, и такая вероятность тем выше, чем меньше длина ключа.

Соответственно, если найти в тексте шифрограммы повторы, причём желательно состоящие из как можно большего числа символов, то при помощи таких повторов можно определить длину ключа. Если повторяющиеся сочетания символов в шифрограмме будут иметь слишком маленькую длину (например, 2 или даже 3), то повышается вероятность того, что разные сочетания символов открытого текста наложились на разные символы ключа, в результате чего получились одинаковые символы шифрограммы. С увеличением длины повторяющихся сочетаний символов такая вероятность стремительно обнуляется.

Итак, беря на вооружение описанную идею, получаем такой метод. В шифрограмме находится как можно больше повторений наибольшей длины. Затем считаются разницы между позициями повторений. Например, первое появление буквосочетания длины 5 находится на 19-й позиции, а второе на 33-й. Разница между позициями равна 14. У числа 14 имеются следующие делители: 1, 2, 7 и 14. И это значит, что длиной ключа могут быть эти числа. Собственно, число 1 обычно не проверяют, а вот все остальные систематически проверяются. Но множество возможных длин ключа можно сузить, и это делается при помощи дальнейшего поиска повторений. Например, в той же шифрограмме другое буквосочетание нашлось на 27-й и 48-й позициях. Разница между 48 и 27 равна 21, и у этого числа следующие делители: 1, 3, 7 и 21. Пересечение двух множеств длин ключей даёт два варианта: 1 и 7, то есть фактически длина ключа равна 7.

Описанный процесс может закончиться неудачно – в результате пересечения множеств может остаться только число 1, что будет противоречить информации о том, что расшифровывается многоалфавитный шифр замены с длиной ключа более 1. Обычно это означает, что как раз и возникла ситуация, когда произошло различное наложение ключа так, что в результате получились одинаковые последовательности символов. Но может оказаться и так, что все расчёты произведены правильно, а в результате всё равно получается длина ключа, равная 1. Тогда имеет смысл пересмотреть гипотезу о том, что перед нами шифрограмма, зашифрованная системой многоалфавитной замены.

После того как определена длина ключа, остаётся дело техники. Для каждого алфавита осуществляется взлом, как для шифра простой замены – при помощи частотного анализа. Для этого проще всего весь текст шифрограммы выписать в столбик шириной в длину ключа, и тогда в каждой колонке будут символы, зашифрованные одним и тем же алфавитом, и к каждому столбцу можно будет применить частотный анализ. Сделать это будет не так просто, как в чистом одноалфавитном шифре, так как для подбора слов надо будет

переходить из алфавита в алфавит, но всё же возможно. А при использовании программных средств эта задача облегчается многократно.

Заключение

Если принять оценку необходимого объёма шифрограммы, данную в прошлой статье цикла, то в отличие от N-граммных шифров многоалфавитные шифры требуют всего лишь в N раз больше объёма знаков шифрограммы по сравнению с одноалфавитным шифром. Другими словами, если для взлома одноалфавитного шифра требуется примерно 100 символов в шифрограмме, то для взлома многоалфавитного шифра с длинной ключа N требуется 100N символов. Количество времени и усилий для дешифровки увеличивается сравнимо. Это, в свою очередь, означает, что многоалфавитные шифры являются менее секретными, чем N-граммные шифры.

Упражнения

В качестве домашнего задания и самостоятельных упражнений рекомендую выполнить следующее.

1. Прочитайте произведения:

Кан Д. Взломщики кодов.

Сингх С. Книга шифров. Тайная история шифров и их расшифровки.

2. Расшифруйте шифрограмму:

ЙНВГЯ СЪПЯР ЧЗУПТ ДЧЫЬС НОНРК ЯМЗДТ СДАЮА ТГДСО
ЯПНКЫ ЗЖЦДС ЪПЕФА ТЙБОН РЙНКЫ ЙТЬСН ЖЯМЮС ЗДОНР
БЮШДМ НФЗЛЗ ЗСНБР ДЗМСД ПДРМЪ ДЗБЯЁ МЪДРК НБЯРБ
ЮЖЯМЪ РФЗЛЗ ДИЙЯЙ СЪБЗГ ЗЧЫОЯ ПНКЫЬ СНМЯЖ БЯМЗД
ФЗЛЗЦ ДРЙНВ НЬКДЛ ДМСЯБ НЖЫЛЗ ДВНОД ПБТЭА ТЙБТЗ
МЯИГЗ ГПТВН ИФЗЛЗ ЦДРЙЗ ИЬКДЛ ДМСМЯ ЖБЯМЗ ДЙНСН
ПНВНМ ЯЦЗМЯ ДСРЮР ЬСНИЁ ДАТЙБ ЪЗМЯЖ БЯМЗД ЙНСНП
НВНРЯ ЛНДГК ЗММНД ЗЖБРД ФСЯЙЗ ФМЯЖБ ЯМЗИЬ СНЗАТ
ГДССП ДСЫДБ НКЧДА МНДРК НБНЬС НВНХЗ ЙКЯРС ЯСДИЖ
ЯОЗЧЗ ДВНТС ДАЮЗФ ГНКЁМ НАЪСЫ ТЁДСП З

Как это принято в криптографии, шифрограмма для удобства разделена на пятёрки символов. При дешифровке их необходимо соединить в одну длинную строку, убрав все пробелы.

3. На своём любимом языке программирования напишите программу, реализующую метод Касиски для атаки на многоалфавитные шифры.