Исторически, в условном IV веке до нашей эры, существовало два метода дизайна шифров: шифры подстановки и перестановки. Шифры подстановки — алгоритм, где в те времена заменяли одну букву сообщения на другую по какому-то принципу. Простой шифр подстановки — по таблице: берем таблицу, где написано, что А меняем на Я, Б на Ю и т. д. Дальше по этой таблице шифруем, по ней же дешифруем.

Как вы считаете, с точки зрения размера ключа насколько это сложный алгоритм? Сколько вариантов ключей существует? Порядок факториала длины алфавита. Мы берем таблицу. Как мы ее строим? Допустим, есть таблица на 26 символов. Букву А можем заменить на любой из них, букву Б — на любой из оставшихся 25, С — на любой из оставшихся 24… Получаем 26\*25\*24\*… — то есть факториал от 26. Факториал размерности алфавита.

Если взять log226!, это будет очень много. Думаю, вы точно получите в районе 100 бит длины ключа, а то и поболее. Оказалось, что с точки зрения формального представления стойкости указанный алгоритм шифрования — довольно неплохой. 100 бит — приемлемо. При этом все, наверное, в детстве или юности, когда сталкивались с кодировками, видели, что такие алгоритмы дешифруются тривиально. Проблем с расшифровкой нет.

Долго существовали всякие алгоритмы подстановки в разных конструкциях. Одним из них, еще более примитивным, является шифр Цезаря, где таблица формируется не случайной перестановкой символов, а сдвигом на три символа: А меняется на D, B на Е и т. д. Понятно, что шифр Цезаря вместе со всеми его вариантами перебрать очень легко: в отличие от табличной подстановки, в ключе Цезаря всего 25 вариантов при 26 буквах в алфавите — не считая тривиального шифрования самого в себя. И его как раз можно перебрать полным перебором. Здесь есть некоторая сложность.

Почему шифр табличной подстановки такой простой? Откуда возникает проблема, при которой мы можем легко, даже не зная ничего про криптографию, расшифровать табличную подстановку? Дело в частотном анализе. Есть самые распространенные буквы — какая-нибудь И или Е. Их распространенность велика, гласные встречаются намного чаще, чем согласные, и существуют негативные пары, никогда не встречающиеся в естественных языках, — что-то вроде ЬЪ. Я даже давал студентам задание сделать автоматический дешифратор шифра подстановки, и, в принципе, многие справлялись.

В чем проблема? Надо статистику распределения букв исказить, чтобы распространенные буквы не так светились в зашифрованном тексте. Очевидный способ: давайте будем шифровать самые часто встречающиеся буквы не в один символ, а в пять разных, например. Если буква встречается в среднем в пять раз чаще, то давайте по очереди — сначала в первый символ будем зашифровывать, потом во второй, в третий и т. д. Далее у нас получится маппинг букв не 1 к 1, а, условно, 26 к 50. Статистика, таким образом, нарушится. Перед нами первый пример полиалфавитного шифра, который как-то работал. Однако с ним есть довольно много проблем, а главное, очень неудобно работать с таблицей.

Дальше придумали: давайте не будем шифровать такими таблицами, а попробуем брать шифр Цезаря и для каждой следующей буквы изменять сдвиг. Результат — шифр Виженера.