

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Ярославский государственный университет им. П.Г. Демидова»

Кафедра компьютерной безопасности и математических методов обработки
информации

КУРСОВАЯ РАБОТА
«РЕАЛИЗАЦИЯ "ВЗЛОМОВ" КРИПТОГРАФИЧЕСКИХ МЕТОДОВ»

Научный руководитель
_____ Белова Л.Ю.
«__» _____ 2013 г.

Студент группы КБ-51
_____ Бедняков А.И.
«__» _____ 2013 г.

Ярославль 2013

Содержание

1. Введение	2
2. Лингвистический и статистический анализ	3
1. Словарный перебор	3
2. N-граммы	4
3. Классические шифры	6
1. Шифр Цезаря	6
1. Описание	6
2. Криптоанализ	7
2. Аффинный шифр	8
1. Описание	8
2. Криптоанализ	9
3. Шифр Виженера	9
1. Описание	10
2. Криптоанализ	11
4. Энигма	13
1. Описание	13
2. Криптоанализ	15
4. Бинарные шифры	16
1. XOR	16
1. Описание	16
2. Криптоанализ	16
5. Блочные шифры	17
1. Blowfish	17
1. Описание	17
2. Стойкость	18
Список литературы	20

Аннотация

Изучение методов защиты неразрывно связано с изучением возможных атак на алгоритмы и на их реализации. Работы по анализу таких шифров, как DES, ГОСТ 28147-89, IDEA требуют большого ресурса и являются чрезвычайно сложными. В то же время на примерах классических шифров можно проиллюстрировать некоторые важные приемы и методы криптоанализа. После анализа классических шифров возможно изучение современных блочных алгоритмов шифрования, становятся доступными идеи линейного и дифференциального криптоанализа.

В этой работе предпринята попытка написания фреймворка для криптоанализа классических шифров и оценки стойкости современных шифров.

1. Введение

Определение 1. *Криптология* — наука, занимающаяся методами шифрования и дешифрования.

Криптология состоит из двух частей — криптографии и криптоанализа. Криптография занимается разработкой методов шифрования данных, в то время как криптоанализ занимается оценкой сильных и слабых сторон методов шифрования, а также разработкой методов, позволяющих взламывать криптосистемы. Слово «криптология» (англ. *cryptology*) встречается в английском языке с XVII века, и изначально означало «скрытность в речи»; в современном значении было введено американским учёным Уильямом Фридманом и популяризовано писателем Дэвидом Каном.

Определение 2. *Криптоанализ* — наука о методах расшифровки зашифрованной информации без предназначенного для такой расшифровки ключа.

В большинстве случаев под криптоанализом понимается выяснение ключа; криптоанализ включает также методы выявления уязвимости криптографических алгоритмов или протоколов. Первоначально методы криптоанализа основывались на лингвистических закономерностях естественного текста и реализовывались с использованием только карандаша и бумаги. Со временем в криптоанализе нарастает роль чисто математических методов, для реализации которых используются специализированные криптоаналитические компьютеры.

2. Лингвистический и статистический анализ

Определение 3. *Лингвистика* (от лат. lingua — язык) — наука, это наука о всех языках мира как индивидуальных его представителях. В широком смысле слова, лингвистика подразделяется на научную и практическую.

Первоначально методы криптоанализа основывались на лингвистических закономерностях естественного текста и реализовывались с использованием только карандаша и бумаги. Со временем в криптоанализе нарастает роль чисто математических методов, для реализации которых используются специализированные криптоаналитические компьютеры.

Определение 4. *Компьютерная лингвистика* — научное направление в области математического и компьютерного моделирования интеллектуальных процессов у человека и животных при создании систем искусственного интеллекта, которое ставит своей целью использование математических моделей для описания естественных языков.

Современная лингвистика обладает мощными методами анализа языковых структур, в том числе синтеза и анализа. В этой работе внимание концентрируется только на последних.

Определение 5. *Анализ текста* — процесс получения высококачественной информации из текста на естественном языке. Как правило, для этого применяется статистическое обучение на основе шаблонов: входной текст разделяется с помощью шаблонов, затем производится обработка полученных данных.

Возможен анализ документа, написанного на неизвестном языке и/или неизвестной системой письма, но это так-же выходит за рамки данной работы. Разобран вопрос возможности автоматического определения корректности текста по самому тексту и языку его написания. В целях данного исследования реализованы простейшие методы подобного анализа и проведено сравнение их корректности и скорости работы.

Тестирование каждого метода - запуск с романом «Война и мир» Льва Николаевича Толстого в качестве входных данных. Такой выбор текста обусловлен его легендарной длиной, что даст корректное представление о эффективности метода по памяти и по времени, и вкраплением в текст иностранных слов и терминов, что покажет общую корректность метода.

1. Словарный перебор

Наиболее простой метод - сравнение всех слов текста со словарем корректных слов нужного языка. Полученное количество совпавших слов делится на количество слов исследуемого текста. Результирующая величина может рассматриваться как вероятность того, что данный текст принадлежит рассматриваемому языку.

Подобная оценка подходит целям данной работы - критерий по этой "вероятности" позволит отделить зерна от плевел и выделить наиболее пригодный вариант - ложное срабатывание в общем случае маловероятно из-за специфики процесса. При негативном результате мы видим несвязный набор символов.

```
In [1]: import linguistics
```

```
In [2]: a = open('../sample/warandpeace', 'r').readlines()
```

```
In [3]: linguistics.define_language_dict(a)
('ru', 0,864536523576)
```

Практика демонстрирует жизнеспособность такого метода. Во-первых, составление словаря для известного языка в необходимой стилистике не представляет трудности при условии доступности интернета. Во-вторых, современные компьютерные мощности позволяют сравнительно быстрое выполнение подобного анализа:

```
$ time python ./cipher/linguistics.py -d ./sample/warandpeace
python 710.10s user 780.55s system 0
```

Возможно совершенствование данного метода путем написания более эффективных структур для хранения словаря, грамотной сериализации и использования оптимизированных алгоритмов поиска по сортированному массиву. Но, как будет показано, в этом нет необходимости. Во-первых мы не перешагнем известное ограничение сложности в $O(\log(n))$ для алгоритмов поиска. Во-вторых, отсутствует необходимость в строгом соответствии текста языку определенному в словаре.

2. N-граммы

Определение 6. *N-грамма* — последовательность из n элементов. С семантической точки зрения, это может быть последовательность звуков, слогов, слов или букв. На практике чаще встречается N-грамма как ряд слов. Последовательность из двух последовательных элементов часто называют биграммы, последовательность из трех элементов называется триграмма. Не менее четырех и выше элементов обозначаются как N-грамма, N заменяется на количество последовательных элементов.

В области обработки естественного языка, N-граммы используются в основном для предугадывания на основе вероятностных моделей. N-граммная модель рассчитывает вероятность последнего слова N-граммы, если известны все предыдущие. При использовании этого подхода для моделирования языка предполагается, что появление каждого слова зависит только от предыдущих слов.

Определение 7. *Инвертированный индекс* (англ. inverted index) — структура данных, в которой для каждого слова коллекции документов в соответствующем списке перечислены все места в коллекции, в которых оно встретилось. Инвертированный индекс используется для поиска по текстам.

Опишем как решается задача нахождения документов в которых встречаются все слова из поискового запроса. При обработке однословного поискового запроса, ответ уже есть в инвертированном индексе — достаточно взять список соответствующий слову из запроса. При обработке многословного запроса берутся списки, соответствующие каждому из слов запроса и пересекаются.

Пусть у нас есть корпус из трех текстов T_0 ="it is what it is" T_1 ="what is it" и T_2 ="it is a banana" тогда инвертированный индекс будет выглядеть следующим образом:

"a": 2

"banana": 2

"is": 0, 1, 2

"it": 0, 1, 2

"what": 0, 1

Здесь цифры обозначают номера текстов, в которых встретилось соответствующее слово. Тогда отработка поискового "what is it" запроса даст следующий результат $\{0, 1\} \cap \{0, 1, 2\} \cap \{0, 1, 2\} = \{0, 1\}$.

In [1]: import linguistics

In [2]: a = open('../sample/warandpeace', 'r').readlines()

In [3]: linguistics.define_language_wgramms(a)

('ru', 0.892340923846)

Время исполнения такого теста так-же в разы превышает результат с словарями:

\$ time python ./cipher/linguistics.py -w ./sample/warandpeace

python 210.10s user 280.55s system 0

3. Классические шифры

Большое влияние на развитие криптографии оказали появившиеся в середине прошлого века работы американского математика Клода Шеннона. В этих работах были заложены основы теории информации. В своей работе «Математическая теория секретной связи» Клод Шеннон обобщил накопленный до него опыт разработки шифров. Оказалось, что даже в сложных шифрах в качестве типичных компонентов можно выделить шифры замены, шифры перестановки или их сочетания.

Шифрами перестановки называются такие шифры, преобразования из которых приводят к изменению только порядка следования символов исходного сообщения. Обычно открытый текст разбивается на отрезки равной длины и каждый отрезок шифруется независимо. Пусть, например, длина отрезков равна n и π - взаимно-однозначное отображение множества $1, 2, \dots, n$ в себя. Тогда шифр перестановки действует так: отрезок открытого текста x_1, \dots, x_n преобразуется в отрезок шифрованного текста $x_{\pi(1)} \dots x_{\pi(n)}$. К шифрам перестановки относится шифр «Сдвиг».

Шифрами замены называются такие шифры, преобразования из которых приводят к замене каждого символа открытого сообщения на другие символы - шифробозначения, причем порядок следования шифробозначений совпадает с порядком следования соответствующих им символов открытого сообщения. Дадим математическое описание шифра замены. Пусть X и Y - два алфавита открытого и соответственно шифрованного текстов, состоящие из одинакового числа символов. Пусть также $g : X \rightarrow Y$ - взаимнооднозначное отображение X в Y . Это значит, что каждой букве x алфавита X соответствует однозначно определенная буква y алфавита Y , которую мы обозначаем символом $g(x)$, причем разным буквам соответствуют разные. Тогда шифр замены действует так: открытый текст x_1, x_2, \dots, x_n преобразуется в шифрованный текст $g(x_1)g(x_2)\dots g(x_n)$. К шифрам замены относится шифр Цезаря.

1. Шифр Цезаря

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется буквой находящейся на некоторое постоянное число позиций левее или правее него в алфавите. Например, в шифре со сдвигом 3 А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

1. Описание

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифмети-

ки:

$$y = (x + k) \mod n$$
$$x = (y - k + n) \mod n,$$

где x — символ открытого текста, y — символ шифрованного текста, n — мощность алфавита, а k — ключ.

2. Криптоанализ

Шифр Цезаря может быть легко взломан даже в случае, когда взломщик знает только зашифрованный текст. Можно рассмотреть две ситуации:

1. взломщик знает (или предполагает), что использовался простой шифр подстановки, но не знает, что это — схема Цезаря;
2. взломщик знает, что использовался шифр Цезаря, но не знает значение сдвига.

В первом случае шифр может быть взломан, используя те же самые методы что и для простого шифра подстановки, такие как частотный анализ и т. д., Используя эти методы, взломщик, вероятно, быстро заметит регулярность в решении и поймёт, что используемый шифр — это шифр Цезаря.

Во втором случае, взлом шифра является даже более простым. Существует не так много вариантов значений сдвига (26 для английского языка), все они могут быть проверены методом грубой силы. Один из способов сделать это — выписать отрывок зашифрованного текста в столбец всех возможных сдвигов — техника, иногда называемая как «завершение простого компонента». Рассмотрим пример для зашифрованного текста «EXXEGOEXSRGI»; открытый текст немедленно опознается глазом в четвертой строке.

Другой способ применения этого метода — это написать алфавит под каждой буквой зашифрованного текста, начиная с этой буквы. Метод может быть ускорен, если использовать заранее подготовленные полоски с алфавитом. Для этого нужно сложить полоски так, чтобы в одной строке образовался зашифрованный текст, тогда в некоторой другой строке мы увидим открытый текст.

Для обычного текста на естественном языке, скорее всего, будет только один вариант декодирования. Но, если использовать очень короткие сообщения, то возможны случаи, когда возможны несколько вариантов расшифровки с различными сдвигами. Например зашифрованный текст MPQY может быть расшифрован как «aden» так и как «know» (предполагая, что открытый текст написан на английском языке). Точно также «ALIIP» можно расшифровать как «dolls» или как «wheel»; «AFCCP» как «jolly» или как «cheer».

Многokратное шифрование никак не улучшает стойкость, так как применение шифров со сдвигом a и b эквивалентно применению шифра со сдвигом $a+b$. В математических терминах шифрование с различными ключами образует группу.

2. Аффинный шифр

Аффинный шифр — это частный случай более общего моноалфавитного шифра подстановки. К шифрам подстановки относятся также шифр Цезаря, ROT13 и Атбаш. Поскольку аффинный шифр легко дешифровать, он обладает слабыми криптографическими свойствами.

1. Описание

В аффинном шифре каждой букве алфавита размера m ставится в соответствие число из диапазона $[0, \dots, m-1]$. Затем при помощи модульной арифметики для каждого числа, соответствующего букве исходного алфавита, вычисляется новое число, которое заменит старое в шифротексте. Функция шифрования для каждой буквы

$$E(x) = (ax + b) \mod m,$$

где модуль m — размер алфавита, а пара a и b — ключ шифра. Значение a должно быть выбрано таким, что a и m — взаимно простые числа. Функция расшифрования

$$D(x) = a^{-1} \times (x - b) \mod m,$$

где a^{-1} — обратное к a число по модулю m . То есть оно удовлетворяет уравнению

$$1 \equiv a \times a^{-1} \mod m.$$

Обратное к a число существует только в том случае, когда a и m — взаимно простые. Значит, при отсутствии ограничений на выбор числа a расшифрование может оказаться невозможным. Покажем, что функция расшифрования является обратной к функции шифрования:

$$\begin{aligned} D(E(x)) &= a^{-1} \times (E(x) - b) \mod m \\ &= a^{-1} \times ((ax + b) - b) \mod m \\ &= a^{-1} \times (ax + b - b) \mod m \\ &= a^{-1} \times ax \mod m \\ &= x \mod m. \end{aligned} \tag{1}$$

Количество возможных ключей для аффинного шифра можно записать с помощью функции Эйлера как $\varphi(m) \times m$.

2. Криптоанализ

Так как аффинный шифр является по сути моноалфавитным шифром замены, то он обладает всеми уязвимостями этого класса шифров. Шифр Цезаря — это аффинный шифр с $a = 1$, что сводит функцию шифрования к простому линейному сдвигу.

В случае шифрования сообщений на русском языке (т. е. с помощью $m = 33$) существует 627 нетривиальных аффинных шифров, не учитывая 33 тривиальных шифра Цезаря. Это число легко посчитать, зная, что существует всего 20 чисел взаимно простых с 33 и меньших 33 (а это и есть возможные значения a). Каждому значению a могут соответствовать 33 разных дополнительных сдвига (значение b); то есть всего существует 2033 или 660 возможных ключей. Аналогично, для сообщений на английском языке (т.е. $m = 26$) всего существует 1226 или 312 возможных ключей. Такое ограниченное количество ключей приводит к тому, что система крайне не криптостойка с точки зрения принципа Керкгоффса.

Основная уязвимость шифра заключается в том, что криптоаналитик может выяснить (путем частотного анализа, полного перебора, угадывания или каким-либо другим способом) соответствие между двумя любыми буквами исходного текста и шифротекста. Тогда ключ может быть найден путем решения системы уравнений. Кроме того, так мы знаем, что a и m — взаимно простые, это позволяет уменьшить количество проверяемых ключей для полного перебора.

Преобразование, подобное аффинному шифру, используется в линейном конгруэнтном методе (разновидности генератора псевдослучайных чисел). Этот метод не является криптостойким по той же причине, что и аффинный шифр.

3. Шифр Виженера

Шифр Виженера (фр. Chiffre de Vigenère) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Этот метод является простой формой многоалфавитной замены. Шифр Виженера изобретался многократно. Впервые этот метод описал Джован Баттиста Беллазо (итал. Giovan Battista Bellaso) в книге *La cifra del. Sig. Giovan Battista Bellaso* в 1553 году, однако в XIX веке получил имя Блеза Виженера, французского дипломата. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа.

Первое точное документированное описание многоалфавитного шифра было сформулировано Леоном Баттиста Альберти в 1467 году, для переключения между алфавитами использовался металлический шифровальный диск. Система Альберти переключает алфавиты после нескольких зашифрованных слов. Позднее, в 1518 году, Иоганн Трисемус в своей работе «Полиграфия» изобрел *tabula recta* — центральный компонент шифра Виженера.

То, что сейчас известно под шифром Виженера, впервые описал Джованни Батиста Беллазо в своей книге *La cifra del. Sig. Giovan Battista Bellaso*. Он использовал идею *tabula recta* Трисемуса, но добавил ключ для переключения алфавитов шифра через каждую букву.

Блез Виженер представил свое описание простого, но стойкого шифра перед комиссией Генриха III во Франции в 1586 году, и позднее изобретение шифра было присвоено именно ему. Давид Кан в своей книге «Взломщики кодов» отозвался об этом осуждающе, написав, что история «проигнорировала важный факт и назвала шифр именем Виженера, несмотря на то, что он ничего не сделал для его создания».

Шифр Виженера имел репутацию исключительно стойкого к «ручному» взлому. Известный писатель и математик Чарльз Лютвидж Доджсон (Льюис Кэрролл) назвал шифр Виженера невзламываемым в своей статье «Алфавитный шифр» англ. *The Alphabet Cipher*, опубликованной в детском журнале в 1868 году. В 1917 году *Scientific American* также отозвался о шифре Виженера, как о неподдающемся взлому. Это представление было опровергнуто после того, как Касиски полностью взломал шифр в XIX веке, хотя известны случаи взлома этого шифра некоторыми опытными криптоаналитиками еще в XVI веке.

Шифр Виженера достаточно прост для использования в полевых условиях, особенно если применяются шифровальные диски. Например, «конфедераты» использовали медный шифровальный диск для шифра Виженера в ходе Гражданской войны. Послания Конфедерации были далеки от секретных, и их противники регулярно взламывали сообщения. Во время войны командование Конфедерации полагалось на три ключевых словосочетания: «Manchester Bluff», «Complete Victory» и — так как война подходила к концу — «Come Retribution».

Гилберт Вернам попытался улучшить взломанный шифр (он получил название шифр Вернама-Виженера в 1918 году), но, несмотря на его усовершенствования, шифр так и остался уязвимым к криптоанализу. Однако работа Вернама в конечном итоге все же привела к получению шифра, который действительно невозможно взломать.

1. Описание

В шифре Цезаря каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, А стало бы D, В стало бы Е и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая *tabula recta* или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На разных этапах кодировки шифр Виженера использует различные алфавиты из этой таблицы. На каждом этапе шиф-

рования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Например, предположим, что исходный текст имеет вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

LEMONLEMONLE

Первый символ исходного текста А зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца А в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста Х получается на пересечении строки Е и столбца Т. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN Ключ: LEMONLEMONLE Зашифрованный текст: LXFORVEFRNHR

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова ; в данной строке находим первый символ зашифрованного текста . Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если буквы А-Z соответствуют числам 0-25, то шифрование и расшифрование Виженера можно записать в виде формул:

$$C_i \equiv (P_i + K_i) \mod 26$$
$$P_i \equiv (C_i - K_i + 26) \mod 26$$

2. Криптоанализ

Шифр Виженера «размывает» характеристики частот появления символов в тексте, но некоторые особенности появления символов в тексте остаются. Главный недостаток шифра Виженера состоит в том, что его ключ повторяется. Поэтому простой криптоанализ шифра может быть построен в два этапа:

Поиск длины ключа. Можно анализировать распределение частот в зашифрованном тексте с различным прореживанием. То есть брать текст, включающий каждую 2-ю букву зашифрованного текста, потом каждую 3-ю и т. д. Как только распределение частот букв будет сильно отличаться от равномерного (например, по энтропии), то можно говорить о найденной длине ключа.

Метод Касиски В 1863 году Фридрих Касиски был первым, кто опубликовал успешный алгоритм атаки на шифр Виженера, хотя Чарльз Беббидж разработал этот алгоритм уже в 1854 году. В то время когда Беббидж занимался взломом шифра Виженера, John Hall Brock Thwaites представил новый шифр в «Journal of the Society of the Arts»; когда Беббидж показал, что шифр Thwaites'a является лишь частным случаем шифра Виженера, Thwaites предложил ему его взломать. Беббидж расшифровал текст, который оказался поэмой «The Vision of Sin» Альфреда Теннисона, зашифрованной ключевым словом Emily — именем жены поэта.

Тест Касиски опирается на то, что некоторые слова, такие как «the» могут быть зашифрованы одинаковыми символами, что приводит к повторению групп символов в зашифрованном тексте. Например: сообщение, зашифрованное ключом ABCDEF, не всегда одинаково зашифрует слово «crypto».

Зашифрованный текст в данном случае не будет повторять последовательности символов, которые соответствуют повторным последовательностям исходного текста. В данном зашифрованном тексте есть несколько повторяющихся сегментов, которые позволяют криптоаналитику найти длину ключа.

Более длинные сообщения делают тест более точным, так как они включают в себя больше повторяющихся сегментов зашифрованного текста. В данном зашифрованном тексте есть несколько повторяющихся сегментов, которые позволяют криптоаналитику найти длину ключа.

Расстояние между повторяющимися DYDUXRMH равно 18, это позволяет сделать вывод, что длина ключа равна одному из значений: 18, 9, 6, 3 или 2. Расстояние между повторяющимися NQD равно 20. Из этого следует, что длина ключа равна 20 или 10, или 5, или 4 или 2. Сравнивая возможные длины ключей, можно сделать вывод, что длина ключа (почти наверняка) равна 2.

Тест Фридмана (иногда называемый кашпа-тест) был изобретен Вильямом Фридманом в 1920 году. Фридман использовал индекс совпадения, который измеряет частоты повторения символов, чтобы взломать шифр. Зная вероятность κ_p того, что два случайно выбранных символа текста совпадают (примерно 0,067 для англ. языка) и вероятность совпадения двух случайно выбранных символов алфавита κ_r (примерно $1 / 26 = 0,0385$ для англ. языка), можно оценить длину ключа как:

$$\frac{\kappa_p - \kappa_r}{\kappa_o - \kappa_r}$$

Из наблюдения за частотой совпадения следует:

$$\kappa_o = \frac{\sum_{i=1}^c n_i(n_i - 1)}{N(N - 1)}$$

Где α — размер алфавита (26 символов для англ. языка), N — длина текста, и n_1 до n_c — наблюдаемые частоты повторения символов зашифрованного текста. Однако, это только приблизительное значение, точность которого увеличивается при большем размере текста. На практике это было бы необходимо для перебора различных ключей приближаясь к исходному.

Частотный анализ Как только длина ключа становится известной, зашифрованный текст можно записать во множество столбцов, каждый из которых соответствует одному символу ключа. Каждый столбец состоит из исходного текста, который зашифрован шифром Цезаря; ключ к шифру Цезаря является всего-навсего одним символом ключа для шифра Виженера, который используется в этом столбце. Используя методы, подобные методам взлома шифра Цезаря, можно расшифровать зашифрованный текст. Усовершенствование теста Касиски, известное как метод Кирхгофа, заключается в сравнении частоты появления символов в столбцах с частотой появления символов в исходном тексте для нахождения ключевого символа для этого столбца. Когда все символы ключа известны, криптоаналитик может легко расшифровать шифрованный текст, получив исходный текст. Метод Кирхгофа не применим, когда таблица Виженера скремблирована, вместо использования обычной алфавитной последовательности, хотя тест Касиски и тесты совпадения всё ещё могут использоваться для определения длины ключа для этого случая.

4. Энигма

Энигма (от греч. — загадка) — портативная шифровальная машина, использовавшаяся для шифрования и дешифрования секретных сообщений. Более точно, Энигма — целое семейство электромеханических роторных машин, применявшихся с 20-х годов XX века.

Энигма использовалась в коммерческих целях, а также в военных и государственных службах во многих странах мира, но наибольшее распространение получила в нацистской Германии во время Второй мировой войны — именно Энигма вермахта (Wehrmacht Enigma) — германская военная модель — чаще всего является предметом дискуссий.

1. Описание

Как и другие роторные машины, Энигма состояла из комбинации механических и электрических подсистем. Механическая часть включала в себя клавиатуру, набор вращающихся дисков — роторов, — которые были расположены вдоль вала и прилегали к нему, и ступенчатого механизма,двигающего один или несколько роторов при каждом нажатии на клавишу.

Конкретный механизм работы мог быть разным, но общий принцип был таков: при каждом нажатии на клавишу самый правый ротор сдвигается на одну позицию, а при определённых условиях сдвигаются и другие роторы. Движение роторов приводит к различным криптографическим преобразованиям при каждом следующем нажатии на клавишу на клавиатуре.

Механические части двигались, замыкая контакты и образуя меняющийся электрический контур (то есть, фактически, сам процесс шифрования букв реализовывался электрически). При нажатии на клавишу клавиатуры контур замыкался, ток проходил через различные цепи и в результате включал одну из набора лампочек, и отображавшую искомую букву кода. (Например: при шифровке сообщения, начинающегося с ANX..., оператор вначале нажимал на клавишу А — загоралась лампочка Z — то есть Z и становилась первой буквой криптограммы. Далее оператор нажимал N и продолжал шифрование таким же образом далее).

Таким образом, постоянное изменение электрической цепи, через которую шёл ток, вследствие вращения роторов позволяло реализовать многоалфавитный шифр подстановки, что давало высокую, для того времени, устойчивость шифра.

Роторы — сердце Энигмы. Каждый ротор представлял собой диск примерно 10 см в диаметре, сделанный из эбонита или бакелита, с пружинными штыревыми контактами на одной стороне ротора, расположенными по окружности. На другой стороне находилось соответствующее количество плоских электрических контактов. Штыревые и плоские контакты соответствовали буквам в алфавите (обычно это были 26 букв от А до Z). При соприкосновении контакты соседних роторов замыкали электрическую цепь. Внутри ротора каждый штыревой контакт был соединён с одним из плоских. Порядок соединения мог быть различным. Три ротора и шпиндель, к которому они крепятся.

Сам по себе ротор производил очень простой тип шифрования: элементарный шифр замены. Например, контакт, отвечающий за букву Е, мог быть соединён с контактом буквы Т на другой стороне ротора. Но при использовании нескольких роторов в связке (обычно трёх или четырёх) за счёт их постоянного движения получается более надёжный шифр.

Преобразование Энигмы для каждой буквы может быть определено математически как результат перестановок. Рассмотрим трёхроторную армейскую модель. Положим, что Р обозначает коммутационную панель, U обозначает отражатель, а L, M, R обозначают действия левых, средних и правых роторов соответственно. Тогда шифрование Е может быть выражено как:

$$E = PRMLUL^{-1}M^{-1}R^{-1}P^{-1}$$

После каждого нажатия клавиш ротор движется, изменяя трансформацию. Например, если правый ротор R проворачивается на i позиций, происходит трансформация

$\rho^i R \rho^{-i}$, где — циклическая перестановка, проходящая от А к В, от В к С, и так далее. Таким же образом, средний и левый ротор могут быть обозначены как j и k вращений М и L. Функция шифрования в этом случае может быть отображена следующим образом:

$$E = P(\rho^i R \rho^{-i})(\rho^j M \rho^{-j})(\rho^k L \rho^{-k})U(\rho^k L^{-1} \rho^{-k})(\rho^j M^{-1} \rho^{-j})(\rho^i R^{-1} \rho^{-i})P^{-1}$$

2. Криптоанализ

Попытки «взломать» Энигму не предавались гласности до конца 1970-х. После этого интерес к Энигме значительно возрос, и множество шифровальных машин представлено к публичному обозрению в музеях США и Европы.

В Немецком музее в Мюнхене находятся оба немецких военных варианта трёх- и четырёхроторной Энигмы, есть и устаревшие гражданские модели. Работающая модель представлена также в Международном шифровальном музее в Форт-Миде (Fort Meade), в Музее компьютерной истории (Computer History Museum) в США, в Блетчли-Парке (Bletchley Park) в Великобритании, в Австралийском военном мемориале (Australian War Memorial) в Канберре, а также в Германии, США, Великобритании и в некоторых других странах Европы.

В 2007 году запущен проект распределённых вычислений Enigma@Home, целью которого является взлом трех зашифрованных сообщений Энигмы, перехваченных в северной Атлантике в 1942 году.

4. Бинарные шифры

Данный раздел выделен в качестве переходной ступени от шифров, оперирующих алфавитом к шифрам бинарного уровня. Такая смена базиса мало влияет на логику анализа, но кардинально изменяет программные алгоритмы.

1. XOR

XOR - это побитовое сложение по модулю (с инвертированием при переполнении), например, $1 + 1 = 0$ т.к. 1 - максимальное значение. Все варианты:

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 1 = 0$$

1. Описание

То есть, операция $z = x \oplus y$ по сути поразрядная (побитовая — результат не зависит от соседних битов). Если только один из соответствующих битов равен 1, то результат 1. А если оба 0 или оба 1, то результат 0. Если внимательно посмотреть на результат применения XOR к двум двоичным числам, то можно заметить, что мы можем восстановить одно из слагаемых при помощи второго: $x = z \oplus y$ или $y = z \oplus x$.

2. Криптоанализ

Отсюда можно сделать следующие выводы: зная число y и применяя XOR к x , мы получим z . Затем, мы, опять же используя y , получим из z обратно число x . Таким образом мы можем преобразовать последовательность чисел $(x)_i$ в последовательность $(z)_i$. Теперь мы можем назвать число y кодирующим (или шифрующим) ключом. Если человек не знает ключа, то он не сможет восстановить исходную последовательность чисел $(x)_i$.

Поскольку каждая буква будет представлена в шифротексте одним и тем же кодом z , то воспользуясь частотным словарем взломщик сможет вычислить шифрующий ключ y , если у него будет в распоряжении достаточно длинный шифротекст.

В случае длинного ключа применяются уже разодранные методы анализа из шифра Виженера.

5. Блочные шифры

1. Blowfish

Blowfish - это алгоритм, разработанный Брюсом Шнайером специально для реализации на больших микропроцессорах. Алгоритм Blowfish не запатентован.

Алгоритм Blowfish оптимизирован для применения в системах, не практикующих частой смены ключей, например, в линиях связи и программах автоматического шифрования файлов. При реализации на 32-битовых микропроцессорах с большим размером кэша данных, например, процессорах Pentium и PowerPC, алгоритм Blowfish заметно быстрее DES. Алгоритм Blowfish не годится для применения в случаях, где требуется частая смена ключей, например, в коммутаторах пакетов, или в качестве однонаправленной хэш-функции. Большие требования к памяти не позволяют использовать этот алгоритм в смарт-картах.

1. Описание

Blowfish представляет собой 64-битовый блочный алгоритм шифрования с ключом переменной длины. Алгоритм состоит из двух частей: расширения ключа и шифрования данных. Расширение ключа преобразует ключ длиной до 448 битов в несколько массивов подключей общим размером 4168 байт. Шифрование данных заключается в последовательном исполнении простой функции 16 раз. На каждом раунде выполняются зависящая от ключа перестановка и зависящая от ключа и данных подстановка. Используются только операции сложения и XOR над 32-битовыми словами. Единственные дополнительные операции каждого раунда - четыре взятия данных из индексированного массива. В алгоритме Blowfish используется множество подключей. Эти подключи должны быть вычислены до начала зашифрования или расшифрования данных.

Каждый из четырех 32-битовых S-блоков содержит 256 элементов:

$$S1, 0, S1, 1, \dots, S1, 255$$

$$S2, 0, S2, 2, \dots, S2, 255$$

$$S3, 0, S3, 3, \dots, S3, 255$$

$$S4, 0, S4, 4, \dots, S4, 255$$

Алгоритм Blowfish представляет собой сеть Файстеля, состоящей из 16 раундов. На вход подается 64-битовый элемент данных x . Для зашифрования данных: Разбить x на две 32-битовых половины: xL , xR Для i от 1 до 16:

$$xL = xLPi$$

$$xR = F(xL)xR$$

Переставить xL и xR (отнять последнюю перестановку)

$$xR = xRP17$$

$$xL = xLP18$$

Объединить xL и xR

Разделить xL на четыре 8-битовых фрагмента: a, b, c и d

$$F(xL) = ((S1, a + S2, b \mod 232)S3, c) + S4, d \mod 232$$

Расшифрование выполняется точно так же, как и зашифрование, но P1,P2,...,P18 используются в обратном порядке.

В реализациях Blowfish, в которых требуется очень высокая скорость, цикл должен быть развернут, а все ключи храниться в кэше. Подключи рассчитываются с помощью самого алгоритма Blowfish. Вот какова точная последовательность действий.

1. Сначала P-массив, а затем четыре S-блока по порядку инициализируются фиксированной строкой. Эта строка состоит из шестнадцатеричных цифр .
2. Выполняется операция XOR над P1 с первыми 32 битами ключа, XOR над P2 со вторыми 32 битами ключа, и т.д. для всех битов ключа (вплоть до P18). Операция XOR выполняется циклически над битами ключа до тех пор, пока весь P-массив не будет инициализирован.
3. Используя подключи, полученные на этапах 1 и 2, алгоритм Blowfish шифрует строку из одних нулей.
4. P1 и P2 заменяются результатом этапа 3.
5. Результат этапа 3 шифруется с помощью алгоритма Blowfish и модифицированных подключей.
6. P3 и P4 заменяются результатом этапа 5.
7. Далее по ходу процесса все элементы P-массива, а затем все четыре S-блока по порядку заменяются выходом постоянно меняющегося алгоритма Blowfish.

Всего для генерации всех необходимых подключей требуется 521 итерация. Приложения могут сохранять подключи - нет необходимости выполнять процесс их получения многократно.

2. Стойкость

Серж Воденэ (Serge Vaudenay) исследовал алгоритм Blowfish с известными S-блоками и r раундами. Как оказалось, дифференциальный криптоанализ может восстановить P-массив с помощью $28r+1$ подобранных открытых текстов. Для некоторых слабых

ключей, которые генерируют плохие S-блоки (вероятность выбора такого ключа составляет $1/2^{14}$), эта же атака восстанавливает Р-массив с помощью всего $24t+1$ подобранных открытых текстов. При неизвестных S-блоках эта атака может обнаружить использование слабого ключа, но не может восстановить сам ключ (и также S-блоки и Р-массив). Эта атака эффективна только против вариантов с уменьшенным числом раундов и совершенно безнадёжна против 16-раундового алгоритма Blowfish. Разумеется, важно и открытие слабых ключей, хотя они, вероятно, использоваться не будут. Слабым называют ключ, для которого два элемента данного S-блока идентичны. До выполнения расширения ключа невозможно установить факт слабости ключа.

Не известны факты успешного криптоанализа алгоритма Blowfish. В целях безопасности не следует реализовывать Blowfish с уменьшенным числом раундов. Компания Kent Marsh Ltd. встроила алгоритм Blowfish в свой продукт FolderBolt, предназначенный для обеспечения защиты Microsoft Windows и Macintosh. Кроме того, алгоритм входит в Nautilus и PGPfone.

Список литературы

- [1] Шнайер Б., “Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си”, М.: Триумф, 2002. — 816 с.
- [2] Фомичев В.М. «Дискретная математика и криптология», М.: Диалог-МИФИ, 2003 г.
- [3] Яценко В.В. «Введение в криптографию». М., 1988 г.
- [4] Cryptography.ru. Статьи Интернет-сайтов.