Департамент образования администрации г. Перми

МБОУ «Лицей №1» г. Перми

Информатика

Шифратор и дешифратор типизированных файлов

Выполнил:

Арамилев Захар Владимирович, 202кл

Руководитель УИР:

Аспирант каф. ММСП

Грибов Дмитрий Сергеевич

Пермь, 2018

Zusammenfassung

In diesem Artikel untersuchen wir Verschlüsselungsalgorithmen. Das wichtigste Element der Digitaltechnik, insbesondere bei Computern und Steuerungssystemen, sind Verschlüsselungs- und Decodiergeräte, die offene Informationen in geschlossene und umgekehrt umwandeln können.

Ziel der Arbeit ist es, die einfachsten Verschlüsselungsmethoden zu untersuchen und einen eigenen Verschlüsselungs- und Entschlüsselungsalgorithmus zu entwickeln.

Das Ergebnis der Arbeit war die Erstellung eines eigenen Encoders.

Die Relevanz des Themas der Arbeit ist, dass alle modernen Computergeräte oder Eingabegeräte Encoder haben. Sie sind wichtige Bestandteile jeder digitalen Technologie, das Prinzip ihrer Arbeit ist im Titel festgelegt.

In dieser Lehr- und Forschungsarbeit werden die Begriffe "Verschlüsselung", "Verschlüssler" betrachtet. Ein besonderes Augenmerk wird auf Verschlüsselungsalgorithmen gelegt.

Оглавление

Введение

Во все времена люди пытались скрыть ту или иную информацию от других. По мере развития цивилизации информации становилось всё больше, а необходимость её скрывать всё важнее и труднее. Всегда существовала и совершенно секретная информация, которая известна определенному кругу лиц и не должна быть обнародована.

Важнейшим элементом цифровой техники, особенно в компьютерах и системах управления, являются шифраторы и дешифраторы, способные преобразовывать открытую информацию в закрытую и обратно. Шифрование выполняется согласно алгоритму шифрования с использованием ключа - небольшой порции информации.

Целью работы является изучение простейших методов шифрования и разработка собственного алгоритма шифрования и дешифрования.

Для достижения поставленной цели необходимо решить следующие **задачи:**

- описать существующие алгоритмы шифрования, их стойкость на взлом;
- сравнить способы шифрования по надежности, по способу реализации и скорости шифрования;
- написать программу, шифрующую данные несколькими способами с расширением *.txt;
- разработать и описать собственный алгоритм шифрования;
- провести анализ результатов, сформулировать предложения по повышению эффективности алгоритма.

Итогом работы является создание собственного шифратора.

Актуальность темы состоит в том, что все современные компьютерные устройства или устройства ввода имеют шифраторы, начиная от клавиатуры, где шифратор кодирует информацию о нажатой клавише, заканчивая вычислительными системами.

Шифраторы и дешифраторы являются важными компонентами любой цифровой техники, принцип их работы заложен в названии. Даже работа обыкновенного калькулятора невозможна без шифратора, который мгновенно выполняет преобразования.

В данной учебно-исследовательской работе рассмотрены понятия «шифрование», «шифратор». Особое внимание уделено алгоритмам шифрования.

Глава 1. Концептуальная постановка задачи

Проблема защиты информации от прочтения имеет многовековую историю. Шифрованием пользовались ещё спартанцы в V в. до н.э., известен шифр римского императора Юлия Цезаря. Многие философы средневековья ломали головы над разработкой шифров для защиты своих трудов или сообщений. уже тогда появилась криптография или тайнопись - совокупность методов, направленных на то, чтобы сделать передаваемую или сохраняемую информацию бесполезной для злоумышленника.

Шифрование – такое преобразование данных, в результате которого их можно прочесть только при помощи ключа. Шифрованием занимается наука, которая называется криптографией**.**

В криптографии любой незашифрованный текст называется открытым текстом, а зашифрованные данные называются зашифрованным текстом.

Современные алгоритмы шифрования представляют собой сложную математическую задачу, для решения которой без знания дешифрующего ключа требуется выполнить гигантский объем вычислений и получить ответ, возможно, только через несколько лет.

Состав пароля	Число знаков пароля						
	4	5	6	7	8	9	10
Только цифры	0 с	0,01 с	0,08 с	0,83 с	8 с	4 мин	14 мин
Латинские буквы без учета регистра	0,04 с	0,9 с	25 с	12 мин	4,9 ч	5,2 дня	0,4 лет
Латинские буквы без учета регистра и цифры	0,14 с	5,5 с	3 мин	1,8 ч	2,7 дня	0,27 лет	9,7 лет
Латинские буквы с учетом регистра и цифры	1,2 с	1,3 мин	1,3 ч	3,4 дня	0,58 лет	35,7 лет	2220 лет
Все возможные символы	6 мин	1,06 дня	0,74 лет	190 лет	48,7 тыс. лет	12 млн лет	3,2 млрд лет

рис. 1

Вне зависимости от метода шифрования любой шифр является слабым (т. е. вскрываемым за реальное время), если длина пароля недостаточно велика. Например, приводимые на рис. 1 (стр.5) данные показывают время, требуемое на подбор пароля на ЭВМ класса Pentium/200 МГц в зависимости от длины пароля и допустимых при его формировании знаков при вскрытии информации (3).

Шифратор представляет собой компонент электронного устройства в виде микросхемы. Ее задачей является преобразование полученного кода из одной системы счисления в другую.

Если говорить о компьютерной технике и большинстве электронных приборов, то самыми популярными и широко используемыми являются шифраторы, преобразовывающие код из позиционного десятичного в параллельный двоичный.

Типизированный файл – это файл, в котором содержатся однотипные данные.

Шифратор — в криптографии устройство для автоматического шифрования. Электромеханические шифраторы появляются в начале 1920-х годов в США и Европе и широко используются вплоть до 1980-х годов, когда распространение получают шифры, предназначенные для использования с вычислительной техникой.

Кодек — устройство или программа, способная выполнять преобразование данных или сигнала. Кодеки часто используются при цифровой обработке видео и звука.

Шифратор (электроника) — логическое устройство, выполняющее преобразование позиционного кода в n-разрядный двоичный код. Таким образом, шифратор — это комбинационное устройство, реализующее обратную дешифратору функцию.

Устройство называют комбинационным, если его выходные сигналы в некоторый момент времени однозначно определяются входными сигналами, имеющими место в этот момент времени.

Глава 2. Математическая постановка задачи

Шифрование информации, хранимой и обрабатываемой в электронном виде, — это кодировка данных, исключающая или серьезно затрудняющая возможность их прочтения (получения в открытом виде) без соответствующего программного или аппаратного обеспечения и, как правило, требующая для открытия данных предъявления строго определенного ключа (пароля, карты, отпечатка и т.д.).

Рассмотрим некоторые, наиболее распространенные сегодня способы шифрования информации.

Криптоалгоритм - это последовательность математических или алгоритмических преобразований, производимых над блоками исходных данных для получения зашифрованного блока данных, недоступного для прочтения сторонними лицами.

1. Base64 – это такой формат шифрования, который используется браузерами. Например, для реализации проверки подлинности вводимых данных в форму пароля и логина. Однако, использовать это шифрование для серьезной защиты информации нельзя, это крайне простой и ненадёжный способ.

Base64 – это двусторонний шифр. Ну, на самом деле, у кодировки Base64 есть еще одно очень хорошее применение. Она отлично подходит для кодирования сложных двоичных файлов и данных в простое текстовое представление.

Base64 — стандарт кодирования двоичных данных при помощи только 64 символов ASCII. Алфавит кодирования содержит текстово-цифровые латинские символы A-Z, a-z и 0-9 (62 знака) и 2 дополнительных символа, зависящих от системы реализации. Каждые 3 исходных байта кодируются 4 символами (увеличение на 1\3).

MIME — стандарт, описывающий передачу различных типов данных по электронной почте, а также, в общем случае, спецификация для кодирования информации и форматирования сообщений таким образом,

чтобы их можно было пересылать по Интернету.

В формате электронной почты MIME base64 — это схема, по которой произвольная последовательность байт преобразуется в последовательность печатных ASCII символов. Используются только символы латинского алфавита в верхнем и нижнем регистре — символы (A—Z, a—z), цифры (0—9), и символы «+» и «/», с символом «=» в качестве специального кода суффикса.

Для того, чтобы преобразовать данные в base64, первый байт помещается в самые старшие восемь бит 24-битного буфера, следующий — в средние восемь и третий— в младшие значащие восемь бит.

Если кодируется менее, чем три байта, то соответствующие биты буфера устанавливаются в ноль. Далее каждые шесть бит буфера, начиная с самых старших, используются как индексы строки́ «ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01234 56789+/» и её символы, на которые указывают индексы, помещаются в выходную строку.

Если кодируются только один или два байта, в результате получаются только первые два или три символа строки, а выходная строка дополняется двумя или одним символами «=». Это предотвращает добавление дополнительных битов к восстановленным данным. Процесс повторяется над оставшимися входными данными.

Исходный текст	**M**								**a**								**n**							
Коды ASCII	77 (0x4d)								97 (0x61)								110 (0x6e)							
Двоичный вид	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Полученный индекс в Base64	19								22								5							
Конечный результат в Base64	**T**								**W**								**F**							

рис. 2

2. RC4. Потоковый шифр RC4 был создан Рональдом Ривестом, сотрудником компании «RSA Security», в 1987 году. Сокращение «RC4» официально обозначает «Rivest cipher 4» или «шифр Ривеста», но его часто

считают сокращением от «Ron’s code» («код Рона»).

В течение семи лет шифр являлся коммерческой тайной, и точное описание алгоритма предоставлялось только после подписания соглашения о неразглашении, но в сентябре 1994 года его описание было анонимно отправлено в список рассылки «Cypherpunks». Вскоре описание RC4 было опубликовано в группе новостей usenet«sci.crypt». Оттуда исходный код попал на множество сайтов в сети Интернет. Опубликованный алгоритм на выходе выдавал шифротексты, совпадающие с шифротекстами, выдаваемыми подлинным RC4. Обладатели легальных копий исходного кода RC4 подтвердили идентичность алгоритмов при различиях в обозначениях и структуре программы.

Поскольку данный алгоритм известен, он более не является коммерческой тайной. Однако, название «RC4» является торговой маркой компании «RSA Security». Чтобы избежать возможных претензий со стороны владельца торговой марки, шифр иногда называют «ARCFOUR» или «ARC4».

Алгоритм шифрования RC4 применяется в некоторых широко распространённых стандартах и протоколах шифрования (например, WEP, WPA, SSL и TLS).

RC4 стал популярен благодаря:

- простоте его аппаратной и программной реализации;
- высокой скорости работы алгоритма в обоих случаях.

В США длина ключа, рекомендуемая для использования внутри страны, равна 128 битам. Соглашение, заключённое между «SPA» и правительством США, разрешило экспортировать шифры RC4 с длиной ключа до 40 бит. 56-и битные ключи разрешено использовать заграничным отделениям американских компаний.

Алгоритм шифрования.

1. Функция генерирует последовательность битов ().
2. Затем последовательность битов посредством операции «суммирование по модулю два» (xor) объединяется с открытым текстом. В результате

получается шифрограмма.

Реализация алгоритма шифрования Base64:

1. Подсчёт всех символов из файла.
2. Запись всех символов из файла в массив.
3. Запись номеров символов, используя таблицу ASCII.
4. Перевод номеров в двоичную систему (как в таблице ASCII).
5. 256 символов то двоичное число будет иметь 8 разрядов т.е. каждый номер в 10-тиричной с.с. будет состоять из 8 нулей и единиц в 2-чной с.с..
6. Запись всех нулей и единиц в другой массив, который заполнен нулями и длиной, кратной шести.
7. Создаем один массив, состоящий из 6 элементов. Этот массив будет обрабатывать предыдущий массив, проходя каждые 6 элементов, т.е. нулей и единиц, и переводя эту последовательность из 6 - нулей и единиц в 10-тиричное число. Максимальное такое число будет - 63, т.к максимальное число, состоящее из 6 нулей и единиц это 111111 в 2-ичной с.с, что есть 63. Минимальное-000000, что в 10-ричной с.с.0. Получившиеся числа мы сравниваем индексами массива «encr», состоящими из 64 элементов: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01 23456789+/". Записываем в файл символы, которые получились при совпадении индексов из массива «encr» и числами, полученными при переводе из 2-ичной в 10-ричную систему счисления.

Реализация алгоритма шифрования RS4

1. Подсчёт всех символов из файла, запись всех в массив исходный массив carr и запись кодов символов(по таблице ASCII) в массив key.
2. Распределение массива key по длине равной количеству символов исходного текста. И запись в массив «pe» с длиной, равной количеству символов исходного теста.

   Например;

   ключ: 56, 78, 54, а длина исходного текста равна 7 тогда в «pe» запишется:

   . 56,78,54,56,78,54,76.
3. Реализация блока S, где S - массив размером от 0 до 256(не включительно) S[i] = i.
4. Перестановка блока S. Индекс «i» = 0 это индекс текущего элемента блока S, т.е. элемента 0. «j» = 0- индекс для перестановки блока S, который будет неравномерно изменяться: j = (j + S[i] + pe[j % (количество символов исходного текста)]) % 256; Затем происходит перестановка двух элементов с помощью функции Swap(S[i], S[j]), и алгоритм будет работать пока индекс «i» не станет равным 255
5. Следующий этап – это создание «Псевдослучайной ключевой последовательности» в массив «K» , равный длине исходного текста:

   . i = 0; j = 0;

   . Цикл генерации:

   . i = (i + 1) mod 256;

   . j = (j + S[i]) mod 256;

   . swap(s[i],s[j]);

   . k[i2] = s[(s[i] + s[j]) mod 256];

   . Шифрование кодов исходного теста с помощью операции «XOR» или побитовой операции, исключающее «или» (одно и то же) и запись получившихся цифр в массив «sh». В языке C++ операция «XOR» Обозначается «^». i = 0;

   . while(i < (длины исходного текста)) {sh[i] = k[i] ^(int)carr[i];}

Глава 3. Результаты моделирования шифратора

Реализация собственного алгоритма на основе трёх шифраторов.

Мой алгоритм шифрования построен на основе первых двух алгоритмов и взятой из алгоритма «DES» - «Начальной перестановки IP». Т.е сначала открытый текст будет зашифрован с помощью RC4, потом через Base64 и в конце через «начальную перестановку IP»

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

. Начальная перестановка IP. (рис 3).

. Шифрованный текст (который прошёл RC4 и Base64) будет разбиваться на 64 - битные блоки или 8 –символьные (по таблице ASCII).

. Шифрование будет происходить поблочно с помощью «начальной перестановки IP». По рисунку шифруем первый блок шифротекста, т.е. на место первого бита будет поставлен бит № 58, на место второго бита № 50, на место третьего бита будет поставлен бит № 42 и.т.д. И так с каждым блоком шифротекста.

Структура алгоритма, построенного на основе не одного, а нескольких шифраторов, существенно затрудняет криптоанализ. Расшифрование алгоритма без знания ключевых параметров усложняется.

В результате работы мы получили надёжный криптостойкий шифратор, состоящий из трёх алгоритмов.

Заключение

Информация - это одна из самых ценных вещей в современной жизни. Появление глобальных компьютерных сетей сделало простым получение доступа к информации как для отдельных людей, так и для больших организаций. Но легкость и скорость доступа к данным с помощью компьютерных сетей, таких как Интернет, также сделали значительными следующие угрозы безопасности данных при отсутствии мер их защиты:

- Неавторизованный доступ к информации
- Неавторизованное изменение информации
- Неавторизованный доступ к сетям и другим сервисам
- Другие сетевые атаки, такие как повтор перехваченных ранее транзакций и атаки типа "отказ в обслуживании"

Именно поэтому в современном мире важно уметь защищать информацию. Эффективность защиты информации криптографическими методами зависит не только от криптостойкости шифра, но и от множества других факторов. Проблема использования криптографических методов в информационных системах (ИС) стала в настоящий момент особо актуальна ещё и потому, что появление новых мощных компьютеров, технологий сетевых и нейронных вычислений сделало возможным дискредитацию криптографических систем, еще недавно считавшихся практически нераскрываемыми.

Все это постоянно подталкивает исследователей на создание новых криптосистем и тщательный анализ уже существующих.

В данной работе автор исследовал некоторые методы шифрования и разработал собственный алгоритм шифрования и дешифрования информации. В дальнейшем работу по данной теме автор планирует продолжить.

Список литературы

1. *Бабаш А.В., Шанкин Г.П.* История криптографии. Часть I. — М.: Гелиос АРВ, 2002. — 240 с. — 3000 экз. — ISBN 5-85438-043-9.
2. *Баричев С. Г.,* Гончаров В. В., Серов Р. Е. Основы современной криптографии — 3-е изд. — М.: Диалог-МИФИ, 2011. — 176 с. — ISBN 978-5-9912-0182-7
3. Балдин К. В., Уткин В. Б.. Информационные системы в экономике: Учебник. — 5-е изд. — М.: Издательско-торго- вая корпорация «Дашков и К0», — 395 с.. 2008
4. Вильям Столлингс. Криптография и защита сетей: принципы и практика. М.: Вильямс, 2001. ISBN 5-8459-0185-5.
5. Герасименко В. А. Защита информации в автоматизированных системах обработки данных. Кн. 1, 2. М.: Энергоатомиздат