Таблица 2 – Вероятности ошибок 1-го рода для процедуры Бенжамини-Хохберга.

Кол-во элементов	Кол-во выборок	4	5	6	7
6		0.04467	0.04232	0.04185	0.04079
7		0.04483	0.04415	0.0424	0.04156
8		0.04607	0.04364	0.04230	0.04212

Далее мы планируем исказить нормальность распределения в одном и в обеих направлениях и получить для искаженных выборок результаты аналогичные Таблице 2. Наблюдаем, что эмпирическая вероятность ошибки 1-го рода укладывается в декларируемый уровень значимости и имеется тенденция к понижению этой вероятности с ростом количества сравниваемых выборок.

Представляется полезным итерационным методом найти уровни значимости, соответствующие статистически наблюдаемым ошибкам 1-го рода для используемых «ненормальных» распределений. Имеет смысл провести аналогичные исследования для распределений с другими значениями параметров и другими малыми объёмами выборок. Также мы планируем аналогичным методом исследовать влияние «ненормальности» на мощность критериев.

Список использованных источников

1. Гмурман В.Е. Теория вероятностей и математическая статистика – М., Высшая школа, 2003;
2. С.Гланц. Медико-биологическая статистика. Пер. с англ. – М.,Практика,1998 – 459 с.;
3. Лапач С.Н., Чубенко А.В., Бабич П.Н., Статистические методы в медико-биологических исследованиях с использованием Excel. – 2-е изд. перераб. и доп. – К.:МОРИОН, 2001. – 408 с.
4. Тьюки, Джон (1949). "Сравнение индивидуальных средних в Дисперсионном анализе". Биометрия. 5 (2): 99–114.
5. Y. Benjamini, Y. Hochberg. «Controlling the Discovery Rate : a Rractical and Powerful Approach to Multiple Testing». Journal of the Royal Statistical Society.Series B (Methodological), Vol.57, No. 1 (1995), pp 289-300;

РАЗРАБОТКА АЛГОРИТМА ШИФРОВАНИЯ НА ОСНОВЕ ASCII КОДА

Автор: Никифоров Иван Сергеевич, студент 1 курса филиала «Протвино» Государственного университета «Дубна»

Научный руководитель: Кульман Татьяна Николаевна, к.т.н., доцент Государственного университета «Дубна» (филиал Протвино)

Аннотация

На примере реального проекта рассматривается алгоритм шифрования на основе ASCII кода. Описывается и показывается работа алгоритма

Annotation

An ASCII code-based encryption algorithm is considered as an example of a real project. The algorithm is described and shown

Ключевые слова: алгоритм, шифрование, ASCII

Keywords: algorithm, encryption, ASCII

Алгоритм — конечная совокупность точно заданных правил решения произвольного класса задач или набор инструкций, описывающих порядок действий исполнителя для решения некоторой задачи.

Целью работы является создания алгоритма шифрования на основе ASCII таблицы для шифровки текста. Предложен оригинальный алгоритм, использующий код ASCII

Актуальность работы: Разработка алгоритма шифрования является необходимым условием для защиты инфраструктуры в интернете

Постановка задачи: создания алгоритма шифрования с последующим шифрованием данных.

Предмет исследования: алгоритм шифрования

Цель исследования: наглядно представить пример реального алгоритма шифрования на основе ASCII кода

Задачи: 1. Разработка 2.Изучение 3. Применение

Инструментарий: Microsoft visual studio 2019, C++

Что такое шифрование и как оно работает?

Мы живем в мире, полном утечек данных и взломов, где эксперты по кибербезопасности постоянно подчеркивают, насколько важны надежные пароли.

Шифрование – это современный метод криптографии, который кодирует информацию таким образом, что только авторизованные стороны могут получить к ней доступ. Большинство служб, ориентированных на безопасность и конфиденциальность, используют шифрование в настоящее время. Одним из наиболее распространенных и простых примеров является электронное сообщение. Если вы отправляете зашифрованное электронное сообщение, это означает, что только вы и ваш получатель сможет его увидеть. Интернет-провайдеры, хакеры и любые другие нежелательные личности не смогут прочитать содержимое сообщения.

Шифрование использует комбинацию алгоритмов и ключей для кодирования или декодирования информации. Существует много типов алгоритмов, которые включают в себя различные способы шифрования и расшифровки данных.

Пример:

Сообщение	К	Р	И	П	Т	О	Г	Р	А	Ф	И	Я
Номер 1	12	18	10	17	20	16	4	18	1	22	10	33
Номер 1 +5	17	23	15	22	25	21	9	23	6	27	15	5
Шифр	П	Х	Н	Ф	Ч	У	З	Х	Е	Щ	Н	Д

Ответ: «Пхнфчузхещнд»

Что такое ключ шифрования?

Ключи генерируются либо генераторами чисел, либо компьютерными алгоритмами - они выполняют один и тот же процесс. Ключи могут быть 64-битные, 128-битные или 256-битные. Число относится к двоичным файлам (нулям и единицам), поэтому чем больше число, тем больше времени и усилий нужно затратить на взлом этого ключа. Большинство современных сервисов шифрования используют 128-битные ключи.

Например, для взлома 256-битного ключа потребовалась бы перебрать более 2^256 возможных комбинаций. Даже относительно слабый 64-битный ключ имеет 18 500 000 000 000 возможных комбинаций.

Типы современного шифрования.

Есть два основных типа шифрования: симметричное (или закрытый ключ) и асимметричное (открытый ключ).

Симметричное шифрование применяет один и тот же (единый) секретный ключ как для кодирования простого текста, так и для декодирования зашифрованного текста. Это означает, что обе стороны должны знать ключ - именно поэтому некоторые называют этот метод общим секретным шифрованием. Симметричное шифрование считается лучшим выбором для передачи больших объемов данных, т.к. оно занимает меньше времени для шифрования и расшифровки. Наиболее популярными алгоритмами для этого шифрования являются RC4 (RC 5 и RC6), AES, DES, 3DES и QUAD. (Симметричные - это алгоритмы с ключом)

Асимметричное шифрование более сложное и довольно новое. Оно использует пару ключей: закрытый ключ должен храниться в секрете и быть известен только владельцу, а открытый ключ может быть общедоступным, без ущерба для безопасности. Открытый ключ применяется для шифрования простого текста, а полученный шифрованный текст можно расшифровать только с помощью закрытого ключа. Эта система позволяет двум сторонам безопасно общаться без необходимости обмениваться ключами дешифровки.

(Ассиметричные - с открытым ключом)

Частный (дешифровальный) ключ никогда не покидает устройство отправителя, поэтому нет никакого способа перехватить ключ во время обмена. Наиболее распространенными алгоритмами, используемыми для этого типа шифрования, являются RSA, Diffie-Hellman, ECC, El Gamal и DSA.

Какой тип шифрования более безопасен?

Оба типа шифрования безопасны, предлагают различные сильные стороны и часто используются вместе. Например, когда большие массивы данных нуждаются в быстром шифровании, то применяется симметричное шифрование. Но обе стороны сначала могут

использовать асимметричное шифрование для обмена секретным ключом симметричного шифрования.

На чём основана идея шифрование с использованием ASCII?

На первом этапе создается одномерный массив с нулевыми значениями, индексы которого являются исходными кодами таблицы ASCII. Вторым этапом разработанного алгоритма шифрования является генерирование определенной последовательности кодовой таблицы ASCII (от 0 до 255) и заполнение значений, созданного на первом этапе массива, новой последовательностью кодовой таблицы. Далее по этой таблице шифруются все открытые данные. Для этого выполняется перестановка значений, используя ASCII код исходного символа для выбора индекса и последующей замены кодом, хранящимся в выбранном индексе массива. В зависимости от практической реализации может использоваться ключ, вводимый пользователем, для одновременного хранения сгенерированной кодовой таблицы вместе с зашифрованными данными. Процесс дешифрования обратен функции шифрования, то есть происходит обратное преобразование из произвольной кодировки в ASCII кодировку.

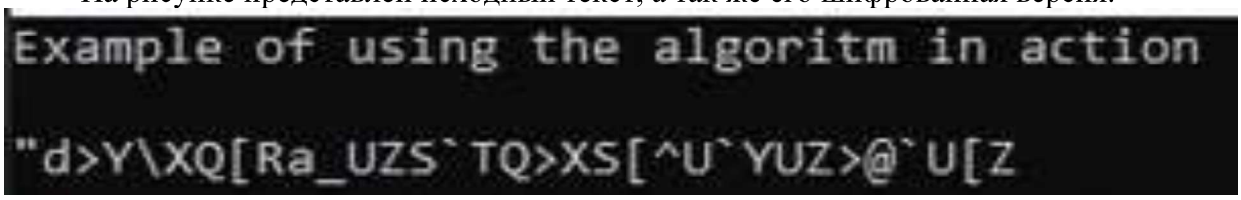
Алгоритм преобразования.

1. Берется массив “CHAR” соответствующий длине текста
2. Происходит заполнение данными в массив
3. Значение “CHAR” переводим в “INT” получаем код “ASCII”
4. Разбиваем код “ASCII” на отрезки: от 0 до 64(+7); от 65 до 122(+15); от 123 до 137(+20); 138-256(0)
5. На каждом из отрезков изменяем значения чисел
6. Преобразуем в тип “CHAR”
7. Получаем зашифрованный текст

Алгоритм дешифровки

1. Берется массив “CHAR” соответствующий длине текста
2. Происходит заполнение данными в массив
3. Значение “CHAR” переводим в “INT” получаем код “ASCII”
4. Разбиваем код “ASCII” на отрезки: от 7 до 7(-7);от 80 до 137(-15); от 143 до 157;158-256(0);
5. На каждом из отрезков изменяем значения чисел
6. Преобразуем в тип “CHAR”
7. Получаем дешифрованный текст

На рисунке представлен исходный текст, а так же его шифрованная версия.



Вывод:

1.Представленный алгоритм является простым алгоритмом шифрования, который способен обеспечить защищенность от большинства лиц, пытающихся получить исходную информацию.

2. Не подходит для обеспечения хорошей защиты от криптоаналитиков.

3. Подойдет для большинства прикладных задач, таких как скрытие правильных ответов к тестовому генератору.