Лабораторная работа No 8

Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Выполнила: Белкина Анастасия Михайловна, НБИбд-01-18

Преподаватель: Кулябов Дмитрий Сергеевич

Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

Задание

1. Изучить указания к работе
2. Прочитать оба закодированных текста, не ища ключ
3. Разработать приложение
4. Определить вид шифротекстов при известном плюче
5. Определить аналитически способ, при котором злоумышленник может прочитать оба текста

Теоретическое введение

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо.

И это очень важно, потому что локальный доступ к файлам для всех программ и всех пользователей позволил бы вирусам без проблем уничтожить систему. Но новым пользователям могут показаться очень сложными новые права на файлы в linux, которые очень сильно отличаются от того, что мы привыкли видеть в Windows. В этой статье мы попытаемся разобраться в том как работают права файлов в linux, а также как их изменять и устанавливать.

Изначально каждый файл имел три параметра доступа. Вот они:

Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;

Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;

Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Но все эти права были бы бессмысленными, если бы применялись сразу для всех пользователей. Поэтому каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение. Группа - любая

группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу. Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла. Именно с помощью этих наборов полномочий устанавливаются права файлов в linux. Каждый пользователь может получить полный доступ только к файлам, владельцем которых он является или к тем, доступ к которым ему разрешен. Только пользователь Root может работать со всеми файлами независимо от их набора их полномочий.

Но со временем такой системы стало не хватать и было добавлено еще несколько флагов, которые позволяют делать файлы не изменяемыми или же выполнять от имени суперпользователя

Выполнение лабораторной работы

1. Код программы на Python

```
A = 15
B = 17
M = 4096
Y0 = 4003

def Gamma(y):
    gamma_list = []
    for _ in range(8):
        y = (A * y + B) % M
        gamma_list.append(y)
    return gamma_list

def Crypt():
    gamma = Gamma(Y0)
    res = open("Result.txt", "w",encoding="utf-8")
    with open('Sourse.txt', 'r',encoding="utf-8") as f:
        r_int = ""
        r=""
        while True:
            temp = f.read(8)
            if temp:
                for i, item in enumerate(temp):
                    r_int = r_int + " " + str(ord(item) ^ gamma[i])
                    r = r + " " + chr(ord(item) ^ gamma[i])
                    res.write(chr(ord(item) ^ gamma[i]))
            else: break
            print(r_int)
            print(r)
    res.close()

    res = open("Result2.txt", "w",encoding="utf-8")
    with open('Sourse2.txt', 'r',encoding="utf-8") as f:
        r_int = ""
        r=""
        while True:
            temp = f.read(8)
            if temp:
                for i, item in enumerate(temp):
                    r_int = r_int + " " + str(ord(item) ^ gamma[i])
                    r = r + " " + chr(ord(item) ^ gamma[i])
                    res.write(chr(ord(item) ^ gamma[i]))
            else: break
            print(r_int)
            print(r)
    res.close()
```

```python
def DeCrypt():
    gamma = Gamma(Y0)
    res = open("NewResult.txt", "w",encoding="utf-8")
    with open('Result.txt', 'r',encoding="utf-8") as f:
        with open('Result2.txt', 'r',encoding="utf-8") as f2:
            r_int = ""
            r = ""
            while True:
                temp = f.read(8)
                temp2 = f2.read(8)
                temp2 = list(temp2)
                if temp:
                    for i, item in enumerate(temp):
                        r_int = r_int + " " + str(ord(item) ^ temp2[i])
                        r = r + chr(ord(item) ^ temp2[i])
                        res.write(chr(ord(item) ^ temp2[i]))
                else: break
            print(r_int)
            print(r)
    res.close()
```

```python
Crypt()
```

```
 3775 3955 499 3389 780 2296 2738 2627
 □ ཱི ǳ ഽ ̌ࣸ ળ □
 3775 3955 499 3389 780 2296 2738 2627 3757 2925 474 3389 770 3218
 □ ཱི ǳ ഽ ̌ࣸ ળ □ ອ ୭ ǚ ഽ ̂ಒ
 3775 3955 462 3389 776 2183 2747 3618
 □ ཱི ǎ ഽ ̈□ □ ย
 3775 3955 462 3389 776 2183 2747 3618 3804 2913 464 3391 1823
 □ ཱི ǎ ഽ ̈□ □ ย ໜ ୡ ǐ ി ܟ
```

```python
DeCrypt()
```

```
1057 32 1053 1086 1074 1099 1084 32 1075 1086 1076 1086 1084 33
С Новым годом!
3775 3955 462 3389 776 2183 2747 3618 3804 2913 464 3391 1823
С Рождеством!
```

Рис.1 Код

2. Файл Sourse

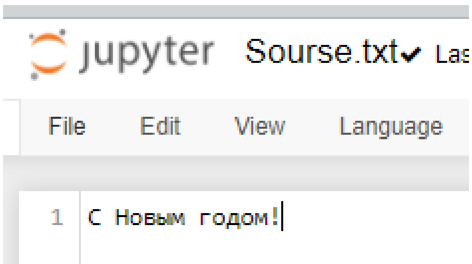

Рис.2 Sourse.txt

3. Файл Sourse2

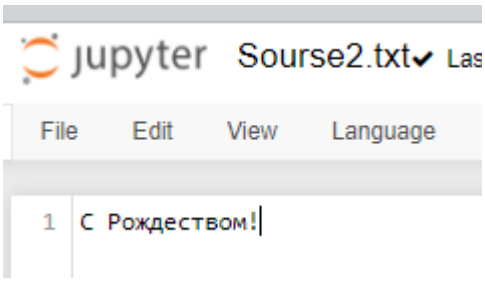

Рис.3 Sourse2.txt

4. Файл Result

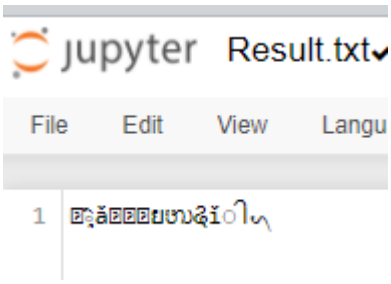

Рис.4 Result.txt

5. Файл Result2

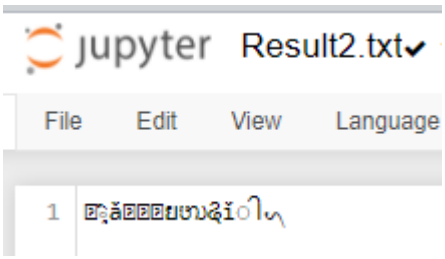

Рис.5 Result2.txt

6. Файл NewResult

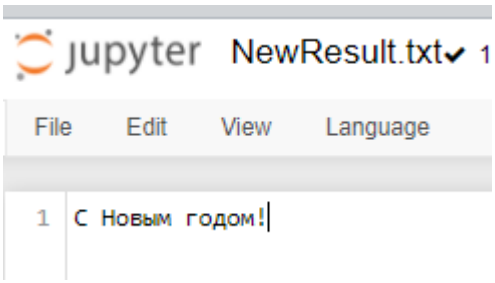

Рис.6 NewResult.txt

7. Файл NewResult2

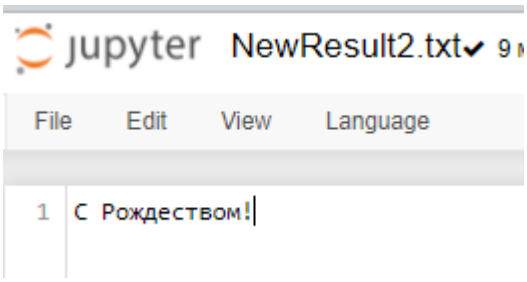

Рис.7 NewResult2.txt

8. Код функций:

A = 15 B = 17 M = 4096 Y0 = 4003

def Gamma(y): gamma_list = [] for _ in range(8): y = (A * y + B) % M gamma_list.append(y) return gamma_list

def Crypt(): gamma = Gamma(Y0) res = open("Result.txt", "w",encoding="utf-8") with open('Sourse.txt', 'r',encoding="utf-8") as f: r_int = "" r="" while True: temp = f.read(8) if temp: for i, item in enumerate(temp): r_int = r_int + " " + str(ord(item) ^ gamma[i]) r = r + " " + chr(ord(item) ^ gamma[i]) res.write(chr(ord(item) ^ gamma[i])) else: break print(r_int) print(r) res.close()

```
res = open("Result2.txt", "w",encoding="utf-8")
with open('Sourse2.txt', 'r',encoding="utf-8") as f:
    r_int = ""
    r=""
    while True:
        temp = f.read(8)
        if temp:
            for i, item in enumerate(temp):
                r_int = r_int + " " + str(ord(item) ^ gamma[i])
                r = r + " " + chr(ord(item) ^ gamma[i])
                res.write(chr(ord(item) ^ gamma[i]))
        else: break
        print(r_int)
        print(r)
res.close()
```

def DeCrypt(): gamma = Gamma(Y0) res = open("NewResult.txt", "w",encoding="utf-8") with open('Result.txt', 'r',encoding="utf-8") as f: with open('Result2.txt', 'r',encoding="utf-8") as f2: r_int = "" r = "" while True: temp = f.read(8) temp2 = f2.read(8) temp2 = list(temp2) if temp: for i, item in enumerate(temp): r_int = r_int + " " + str(ord(item) ^ temp2[i]) r = r + chr(ord(item) ^ temp2[i]) res.write(chr(ord(item) ^ temp2[i])) else: break print(r_int) print(r) res.close()

Выводы

Освоила на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом