07.12.24

Шифр Вернама
„генерация ключа“

Для достижения *абсолютной криптостойкости* необходимо чтобы сгенерированный ключ имел случайное дискретное равномерное распределение:

$P_k(k) = 1/2^N$

Воспользуемся:

/dev/random и **/dev/urandom** — специальные символьные псевдоустройства в некоторых UNIX-подобных системах, впервые появившиеся в ядре Linux версии 1.3.30.

Разницу между ними можно описать так:

- `/dev/random` — генератор случайных чисел;
- `/dev/urandom` — генератор псевдослучайных чисел.

При чтении данных из устройства `/dev/random` выводятся только случайные байты, полностью состоящие из битов шума «хаотичного» пула ОС. Если «хаотичный» пул опустел, `/dev/random` ничего не выдаст, пока необходимое количество битов в пуле не будет создано, читающая /dev/random программа будет ждать появления очередного случайного байта.

В ядре Linux «хаотичный» пул получает энтропию из нескольких источников, в том числе из аппаратного генератора случайных чисел современных процессоров Intel.

Устройство `/dev/random` может быть необходимо пользователям, которые требуют очень высокого коэффициента случайности, например, при создании ключа шифрования, предполагающего длительное использование.

Чтение данных устройства `/dev/urandom` возвратит столько байтов, сколько было запрошено. В результате, если в пуле было недостаточно битов, теоретически возможно найти уязвимость алгоритма, использующего это устройство (на настоящее время нет опубликованных работ о такой атаке). Если это важно, следует использовать `/dev/random`. Пример типичного использования `/dev/urandom` — заполнение массива «шумом»

Реализованный алгоритм генерации ключа на языке программирования Python.

```
import os
import platform

def generate_key_with_dev_random(length):
    """
    Generate a cryptographically secure key using /dev/random
on Unix
    or an equivalent source on Windows.

    :param length: Length of the key in bytes.
    :return: Randomly generated key as bytes.
    """
    if platform.system() == 'Windows':
        # Use os.urandom (which relies on CryptGenRandom on
Windows)
        key = os.urandom(length)
    elif platform.system() in ['Linux', 'Darwin']:
        # Use /dev/random for Unix-based systems
        with open('/dev/random', 'rb') as random_source:
            key = random_source.read(length)
    else:
        raise NotImplementedError("This platform is not
supported.")

    return key

length = 16  # count of bytes for key
key = generate_key_with_dev_random(length)
print("Key (hex):", key.hex())
```

Список литературы

„/dev/random,/dev/urandom“ - https://ru.wikipedia.org/wiki//dev/random_%D0%B8_/dev/urandom (07.12.24)