

Структура научной презентации

Простейший шаблон

Карпачев Я. О.

Российский университет дружбы народов, Москва, Россия

Информация

- Карпачев Я. О.
- студент
- Российский университет дружбы народов

Цель работы

Освоить на практике применение режима однократного гаммирования

Выполнение лабораторной работы

Пишим скрипт

```
yokarpachev@vbox: ~/Desktop
File Actions Edit View Help
import random
import string
from typing import List, Tuple

EXCHARS = "0123456789ABCDEF"

def generate_key(length: int) → str:
    return ''.join(random.choice(EXCHARS) for _ in range(length))

def xor_strings(a: str, b: str) → str:
    return ''.join(chr(ord(x) ^ ord(y)) for x, y in zip(a, b))

def to_hex(s: str) → str:
    return ''.join(f'{ord(ch):02X}' for ch in s)

def find_possible_keys(cipher: str, fragment: str) → List[Tuple[int, str]]:
    flag_len = len(fragment)
    out: List[Tuple[int, str]] = []
    for pos in range(len(cipher) - flag_len - 1):
        key_candidate = xor_strings(cipher[pos:pos + flag_len], fragment)
        out.append((pos, key_candidate))
    return out

def main() → None:
    plain = "Happy new Year, friends!"
    key = generate_key(len(plain))
    cipher = xor_strings(plain, key)
    print("1. Open text: ", plain)
    print("2. Key: ", key)
    print("3. Crypto: ", to_hex(cipher))
    fragment = input("vvod: ")
    cand = find_possible_keys(cipher, fragment)
    if not cand:
        print("5. Possible keys - No fragment")
        return
    keys_only = [k for _, k in cand]
    print("Possible keys: ", ', '.join(keys_only))
    pos0, key0 = cand[0]
    decrypted = xor_strings(cipher[pos0:pos0 + len(fragment)], key0)
    print("6. decryp fragment: ", decrypted)

if __name__ == "__main__":
    main()
```

```
(yokarpachev@vbox)-[~/Desktop]
$ python3 lab7.py
1. Open text: Happy new Year, friends!
2. Key: CFB65EE0E83C7C431D6F9898
3. Crypto: 0B 27 32 46 4C 65 2B 55 32 62 6A 26 56 31 18 13 57 36 5F 23 57 5C 4A 63
vvod: Happy new year!
Possible keys: CFB65EE0EBC7C9, oS6<
;WJ_3Pj2, z'<Ru\Tyav, -[, QvH}r%, [%KBC!av6D~, -J%BH3F8j2W-, c4B8To3
.S>Qv, S_v_}dwO:B%}
6. decrypt fragment: Happy new year!

(yokarpachev@vbox)-[~/Desktop]
$
```

Рис. 2: Работа программы

Контрольные вопросы

1. Смысл однократного гаммирования – сложение (XOR) текста с однократной случайной гаммой той же длины.
2. Недостатки: нужна истинно случайная гамма; ключ хранить/передавать так же долго, как сообщение; ключ нельзя переиспользовать.
3. Преимущества: абсолютная криптостойкость; простота реализации; симметричность (шифр = дешифр).
4. Длины равны, чтобы каждый символ текста «прикрывался» одним символом гаммы; иначе остаётся статистическая избыточность.
5. Операция – XOR (сложение по модулю 2); даёт -- при повторном применении тем же ключом восстанавливается исходник.
6. Шифротекст: $C_i = P_i \oplus K_i$.
7. Ключ: $K_i = C_i \oplus P_i$.
8. Условия абсолютной стойкости: (а) гамма истинно случайна; (б) длина ключа = длина сообщения; (с) ключ используется лишь однажды.

Я освоить на практике применение режима однократного гаммирования