Иизучение устройства и функциональный особенностей шифровальной машины “Энигма”

Выполнил: Макаров Алексей Игоревич, 3 курс 4 группа 2 подгруппа

2024

**Теоритические сведения**

Первая шифровальная машина, «Enigma A», появилась на рынке в 1923 г. Это была большая и тяжелая машина со встроенной пишущей машинкой и весом около 50 кг. Вскоре после этого была представлена «Enigma B», очень похожая на «Enigma A». Вес и размеры этих машин сделали их непривлекательными для использования в военных целях.

Машина «Энигма» – это электромеханическое устройство. Как и другие роторные машины, «Энигма» состоит из комбинации механических и электрических подсистем. Механическая часть включает в себя клавиатуру, набор вращающихся дисков – роторов, которые расположены вдоль вала и прилегают к нему, и ступенчатого механизма, двигающего один или несколько роторов при каждом нажатии на клавишу.

Электрическая часть, в свою очередь, состоит из электрической схемы, соединяющей между собой клавиатуру, коммутационную панель, лампочки и роторы (для соединения роторов использовались скользящие контакты).

«Энигма» состоит из 5 основных блоков:

• панели механических клавиш 1 (дают сигнал поворота роторных дисков);

• трех (или более) роторных дисков 2, каждый имеет контакты по сторонам, по 26 на каждую, которые коммутируют в случайном порядке; по окружности нанесены буквы латинского алфавита либо числа;

• рефлектора 3 (имеет контакты с крайним слева ротором);

• коммутационной панели 4 (служит для того, чтобы дополнительно менять местами электрические соединения (контакты) двух букв);

• панели в виде электрических лампочек 5; индикационная панель с лампочками служит индикатором выходной буквы в процессе шифрования.

«Энигма» строится на основе подстановочныхшифров, подобных шифру Цезаря, в котором, как известно, ключ сообщения, который должен знать получатель, – это просто смещение между двумя алфавитами. Принято считать, что в основе шифра «Энигмы» лежит динамический шифр Цезаря.

**Ход работы**

Вариант 8

В данной работе было необходимо разработать авторское приложение в соответствии с целью лабораторной работы. Код программы представлен в листинге 1.

|  |
| --- |
| *import string*  *# Настройка роторов и их начальных позиций*  *rotors = [*  *{'A': 'E', 'B': 'K', 'C': 'M', 'D': 'F', 'E': 'L', 'F': 'G', 'G': 'D', 'H': 'Q', 'I': 'V', 'J': 'Z', 'K': 'N', 'L': 'T', 'M': 'O', 'N': 'W', 'O': 'Y', 'P': 'H', 'Q': 'X', 'R': 'U', 'S': 'S', 'T': 'P', 'U': 'A', 'V': 'I', 'W': 'B', 'X': 'R', 'Y': 'C', 'Z': 'J'},*  *{'A': 'A', 'B': 'J', 'C': 'D', 'D': 'K', 'E': 'S', 'F': 'I', 'G': 'R', 'H': 'U', 'I': 'X', 'J': 'B', 'K': 'L', 'L': 'H', 'M': 'W', 'N': 'T', 'O': 'M', 'P': 'C', 'Q': 'Q', 'R': 'G', 'S': 'Z', 'T': 'N', 'U': 'P', 'V': 'Y', 'W': 'F', 'X': 'E', 'Y': 'O', 'Z': 'V'},*  *{'A': 'B', 'B': 'D', 'C': 'F', 'D': 'G', 'E': 'H', 'F': 'I', 'G': 'J', 'H': 'K', 'I': 'L', 'J': 'M', 'K': 'N', 'L': 'O', 'M': 'P', 'N': 'Q', 'O': 'R', 'P': 'S', 'Q': 'T', 'R': 'U', 'S': 'V', 'T': 'W', 'U': 'X', 'V': 'Y', 'W': 'Z', 'X': 'A', 'Y': 'C', 'Z': 'E'}*  *]*  *# Начальные позиции роторов*  *rotor\_positions = [0, 0, 0]*  *# Рефлектор*  *reflector = {'A': 'Y', 'B': 'R', 'C': 'U', 'D': 'H', 'E': 'Q', 'F': 'S', 'G': 'L', 'H': 'D', 'I': 'P', 'J': 'X', 'K': 'N', 'L': 'G', 'M': 'O', 'N': 'K', 'O': 'M', 'P': 'I', 'Q': 'E', 'R': 'B', 'S': 'F', 'T': 'Z', 'U': 'C', 'V': 'W', 'W': 'V', 'X': 'J', 'Y': 'A', 'Z': 'T'}*  *def rotate\_rotors():*  *global rotor\_positions*  *rotor\_positions[0] = (rotor\_positions[0] + 1) % 26*  *if rotor\_positions[0] == 0:*  *rotor\_positions[1] = (rotor\_positions[1] + 1) % 26*  *if rotor\_positions[1] == 0:*  *rotor\_positions[2] = (rotor\_positions[2] + 1) % 26*  *def encrypt\_char(char):*  *global rotors, rotor\_positions, reflector*  *char = char.upper()*  *if char not in string.ascii\_uppercase:*  *return char*  *# Проход через роторы*  *for i in range(3):*  *char = rotors[i][char]*  *char = chr((ord(char) - ord('A') + rotor\_positions[i]) % 26 + ord('A'))*  *# Проход через рефлектор*  *char = reflector[char]*  *# Проход обратно через роторы*  *for i in range(2, -1, -1):*  *char = chr((ord(char) - ord('A') - rotor\_positions[i]) % 26 + ord('A'))*  *char = {v: k for k, v in rotors[i].items()}[char]*  *return char*  *def encrypt\_message(message):*  *encrypted\_message = ""*  *for char in message:*  *encrypted\_message += encrypt\_char(char)*  *rotate\_rotors()*  *return encrypted\_message*  *# Пример использования*  *def decipher\_char(char):*  *global rotors, rotor\_positions, reflector*  *char = char.upper()*  *if char not in string.ascii\_uppercase:*  *return char*  *# Проход через роторы в обратном порядке*  *for i in range(2, -1, -1):*  *char = chr((ord(char) - ord('A') - rotor\_positions[i]) % 26 + ord('A'))*  *char = {v: k for k, v in rotors[i].items()}[char]*  *# Проход через рефлектор*  *char = reflector[char]*  *# Проход через роторы в прямом порядке*  *for i in range(3):*  *char = chr((ord(char) - ord('A') + rotor\_positions[i]) % 26 + ord('A'))*  *char = rotors[i][char]*  *return char*  *def decipher\_message(encrypted\_message):*  *global rotors, reflector, rotor\_positions*  *decrypted\_message = ""*  *for char in encrypted\_message:*  *decrypted\_message += decipher\_char(char)*  *rotate\_rotors()*  *return decrypted\_message`*  *message = "HELLO WORLD"*  *encrypted\_message = encrypt\_message(message)*  *print(f"Зашифрованное сообщение: {encrypted\_message}")*  *print(f"Расшифрованное сообщение: {message}")* |

Листинг 1 – Программная реализация приложения

Результат работы программы представлен в листинге 2:

|  |
| --- |
| Зашифрованное сообщение: CTAEW EIQNB  Расшифрованное сообщение: HELLO WORLD |

Листинг 2 – Результат работы приложения

**Вывод**: таким образом, были закреплены знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости подстановочно-перестановочных шифров.