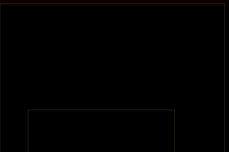


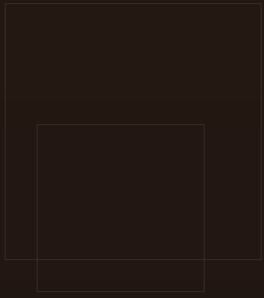
# Лабораторная работа №3: Шифрование методом гаммирования

Хамза Хуссен

Шифрование методом гаммирования

Лабораторная работа №3





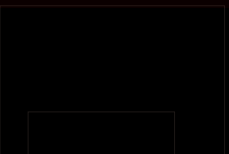
# Ц е л ь и с с л е д о в а н и я

Изучение и практическое освоение алгоритма шифрования методом гаммирования.

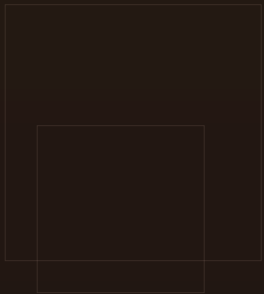
Теоретические основы потокового шифрования.

Реализация алгоритма на Python.

Анализ криптостойкости метода.



# П р и н ц и п   г а м м и р о в а н и я



Открытый текст объединяется с гаммой (псевдослучайной последовательностью).

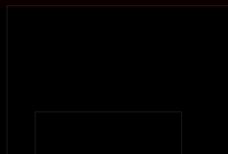
Основная операция: побитовое сложение по модулю 2 (XOR).

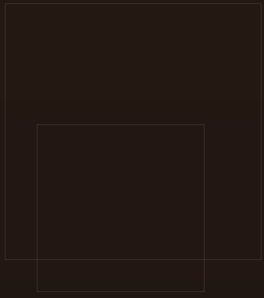
Процесс:

Генерация гаммы.

Наложение гаммы на текст.

Расшифрование — обратный процесс.





# Особенности метода

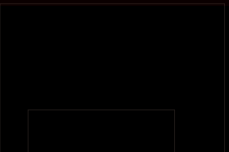
Криптостойкость зависит от гаммы:

Длинный период.

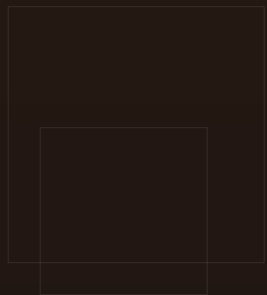
Статистическая непредсказуемость.

Шифр Вернама: абсолютная стойкость при истинно случайной гамме.

На практике используются детерминированные ГПСЧ.



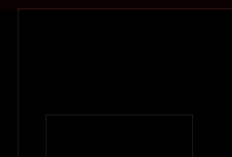
# Уязвимости и модификации

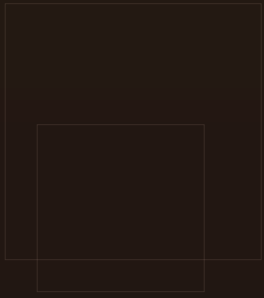


Уязвимость: при известном фрагменте открытого текста возможен криптоанализ.

Решение: гаммирование с обратной связью.

Гамма зависит от уже зашифрованных данных, что повышает стойкость.





# Реализация на Python

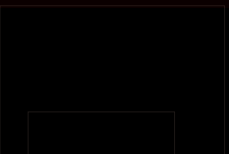
Словарь для преобразования букв в числа и обратно.

Алгоритм:

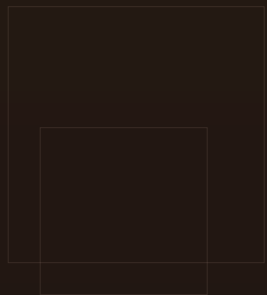
Преобразование текста и гаммы в числа.

Сложение по модулю 33 (для русского алфавита).

Обратное преобразование в текст.



# К о н т р о л ь н ы й   п р и м е р



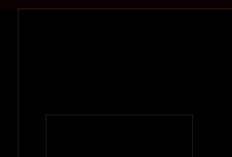
Текст: "примертекста"

Гамма: "гаммаключ"

Результат:

Зашифрованный текст.

Успешное расшифрование.





# Пример таблицы шифрования

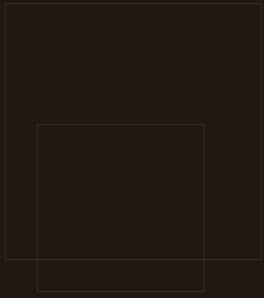
(На основе приложенной таблицы)

Преобразование букв в числа.

Сложение гаммы с текстом.

Применение модуля  $N$  и обратное преобразование.

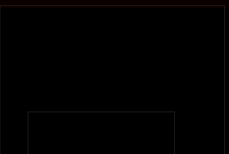




# С п и с о к л и т е р а т у р ы

Шифрование методом гаммирования.

Режим гаммирования в блочном алгоритме шифрования.





# Вопросы и заключение

Криптографические методы остаются актуальными для изучения основ защиты информации.

Реализация на Python позволяет наглядно понять алгоритмы.

Спасибо за внимание!

