Лабораторная работа 3. Исследование криптографических модулей Python

Цель работы: изучение основных принципов криптографии и практическое применение криптографических модулей Python для шифрования, дешифрования, хеширования данных.

Порядок выполнения работы:

I. Теоретическое исследование

1) Изучить теорию алгоритма шифрования, связанного с вашим вариантом.
2) Ответить на вопросы:
   - Является ли алгоритм симметричным или асимметричным?
   - Использует ли он блочное или поточное шифрование?
   - Какие параметры (ключи, векторы инициализации, длина блока) используются?

II. Практическое задание

3) Установить нужный модуль Python.
4) Написать программу для решения задачи (согласно варианту):
   - генерации ключей (если необходимо);
   - шифрования текста (или файлов);
   - дешифрования данных (если применимо).
5) Протестировать программу на разных входных данных.

III. Подготовка отчета и защита

6) Оформить отчет в MS Word, который включает:
   - Теоретическое описание алгоритма
   - Программный код для решения задачи
   - Результаты работы программы (пример входных и выходных данных)
   - Выводы (оценка работы алгоритма).

№ п/п	Критерии оценки	Макс. балл
1.	Теоретическое обоснование: описание алгоритмов, их характеристик и области применения.	1
2.	Корректность работы программы и качество кода: программа успешно выполняет шифрование и дешифрование данных, работает без ошибок, код хорошо структурирован, содержит комментарии и понятные переменные.	3
3.	Оформление отчета: логичное и грамотное описание выполненной работы, наличие скриншотов тестирования, оценка работы алгоритма, сравнительный анализ скорости и стойкости методов (при необходимости).	1
4.	Защита работы	1
	Итоговый балл	**6**

Варианты заданий

№ вар.	Задание
1.	**Исследование модуля hashlib** Написать программу для хеширования паролей с использованием алгоритмов MD5, SHA-256 и SHA-512. Сравнить длины хешей и время их вычисления.
2.	**Исследование модуля cryptography** Написать программу для шифрования и дешифрования текста с использованием алгоритма AES в режиме CBC. Реализовать генерацию ключа и вектора инициализации (IV).
3.	**Исследование модуля pillow** Реализовать программу, которая скрывает текстовое сообщение в изображении и извлекает его.
4.	**Исследование модуля stegano** Реализовать программу, которая скрывает текстовое сообщение в видео и извлекает его.
5.	**Исследование модуля cryptography** Написать программу, которая генерирует пару ключей RSA, шифрует текст с использованием открытого ключа и расшифровывает с использованием закрытого ключа.
6.	**Исследование модуля хеширования файлов hashlib** Написать программу, которая вычисляет контрольную сумму файла (MD5, SHA-256) и проверяет целостность данных после передачи файла.
7.	**Исследование модуля pycryptodome** Реализовать шифрование и дешифрование сообщений с использованием Blowfish. Исследовать зависимость длины зашифрованного текста от размера блока.
8.	**Исследование модуля steganography** Реализовать программу, которая скрывает текстовое сообщение в изображении и извлекает его.
9.	**Исследование работы HMAC (hmac, hashlib)** Написать программу для проверки целостности сообщений с использованием HMAC на основе SHA-256.
10.	**Исследование работы алгоритма SHA-3 (hashlib)** Написать программу, сравнивающую скорость работы SHA-256 и SHA-3-256 при хешировании одного и того же текста.
11.	**Реализация скрытия данных в аудиофайлах (pydub, wave)** Разработать программу, скрывающую текстовые данные в аудиофайле методом LSB (наименее значащего бита).
12.	**Исследование работы алгоритма SHA-3 (hashlib)** Написать программу, сравнивающую скорость работы SHA-256 и SHA-3-256 при хешировании одного и того же текста.
13.	**Исследование работы алгоритма PBKDF2 (hashlib, os)** Написать программу, использующую PBKDF2 для безопасного хранения паролей. Сравнить устойчивость к атакам при разном количестве итераций.

№ вар.	Задание
14.	**Исследование модуля secrets: генерация случайных криптографически стойких ключей** Использовать модуль secrets для генерации безопасных паролей и ключей. Реализовать генератор паролей с разными уровнями сложности.
15.	**Исследование модуля cryptography** Генерация пары ключей ECC и шифрование: использовать эллиптические кривые (ECC) для генерации ключей, подписания и проверки подписи сообщения.
16.	**Исследование модуля pycryptodome** Написать программу для шифрования и дешифрования файлов с использованием AES в режиме GCM.
17.	**Исследование модуля zxcvbn** Исследование устойчивости паролей: использовать библиотеку zxcvbn для анализа безопасности паролей. Реализовать инструмент, оценивающий сложность введенного пароля.
18.	**Исследование модуля pycryptodome** Написать программу для генерации общего ключа двумя сторонами с использованием алгоритма Диффи-Хеллмана.
19.	**Исследование модуля cryptography** Использовать модуль cryptography для создания самоподписанного сертификата X.509 и его валидации.
20.	**Исследование модуля cryptography** Использовать модуль cryptography.fernet для симметричного шифрования файла.
21.	**Реализация скрытия данных в аудиофайлах (pydub, wave)** Разработать программу, скрывающую текстовые данные в аудиофайле методом LSB (наименее значащего бита).
22.	**Исследование модуля steganography** Реализация скрытия данных в текстовом файле**.** Написать программу, которая встраивает скрытое сообщение в текстовый файл, изменяя незначащие символы (например, пробелы, табуляции).
23.	**Исследование модуля stegano** Реализовать программу, которая скрывает текстовое сообщение в изображении и извлекает его.
24.	**Исследование модуля pyotp, qrcode** Разработать систему генерации одноразовых паролей (TOTP) с использованием библиотеки pyotp. Реализовать генерацию QR-кода для аутентификации.