**Лабораторная работа №1**

**Исследование симметричных алгоритмов шифрования**

**1. Базовые задания**

1. **Реализация алгоритма шифрования AES**:  
   Написать программу, реализующую алгоритм AES-128, и провести шифрование и дешифрование текстовых данных.  
   *Дополнительно*: сравнить время выполнения для разных размеров ключей (128, 192, 256 бит).
2. **Реализация алгоритма DES**:  
   Разработать программу для шифрования и дешифрования данных с использованием алгоритма DES.  
   *Дополнительно*: исследовать уязвимости DES к атаке полным перебором (brute-force).
3. **Сравнение AES и DES**:  
   Провести сравнительный анализ AES и DES по следующим параметрам: скорость шифрования, стойкость к атакам, размер ключа.

**2. Анализ и тестирование**

1. **Исследование режимов работы AES**:  
   Реализовать и сравнить различные режимы работы AES (ECB, CBC, CFB, OFB, GCM) на примере шифрования текстовых и бинарных данных.
2. **Анализ криптостойкости AES**:  
   Провести исследование криптостойкости AES к атакам (например, атака по сторонним каналам или атака на связанных ключах).
3. **Тестирование алгоритма на корректность**:  
   Написать тесты для проверки корректности реализации AES или DES, используя известные тестовые векторы (test vectors).

**3. Практические применения**

1. **Шифрование файлов**:  
   Разработать приложение для шифрования и дешифрования файлов с использованием AES.  
   *Дополнительно*: добавить возможность выбора режима работы (CBC, GCM).
2. **Шифрование сетевого трафика**:  
   Реализовать простой клиент-серверный протокол с использованием AES для шифрования передаваемых данных.
3. **Интеграция с базами данных**:  
   Написать скрипт для шифрования чувствительных данных в базе данных (например, паролей или персональных данных) с использованием AES.

**4. Исследовательские задания**

1. **Оптимизация AES**:  
   Исследовать возможности оптимизации AES для различных платформ (CPU, GPU) и сравнить производительность.
2. **Анализ влияния размера блока на безопасность**:  
   Исследовать, как размер блока влияет на криптостойкость симметричных алгоритмов (на примере AES и DES).
3. **Исследование гибридных схем шифрования**:  
   Разработать гибридную схему шифрования, сочетающую симметричный (AES) и асимметричный (RSA) алгоритмы.

**5. Творческие и углублённые задания**

1. **Реализация упрощённого AES**:  
   Разработать упрощённую версию AES (например, с уменьшенным количеством раундов) и исследовать её уязвимости.
2. **Визуализация работы AES**:  
   Создать визуализацию процесса шифрования AES (например, преобразование байтов на каждом раунде).
3. **Исследование альтернативных симметричных алгоритмов**:  
   Реализовать и проанализировать альтернативные симметричные алгоритмы (например, Blowfish, Twofish, ChaCha20) и сравнить их с AES.

**Лабораторная работа №2**

**Реализация асимметричных алгоритмов шифрования**

### ****1. Базовые задания****

1. **Реализация алгоритма RSA**:  
   Написать программу для генерации ключей, шифрования и дешифрования данных с использованием алгоритма RSA.  
   Дополнительно: реализовать padding (OAEP или PKCS#1 v1.5).
2. **Реализация алгоритма Эль-Гамаля**:  
   Разработать программу для шифрования и дешифрования данных с использованием алгоритма Эль-Гамаля.  
   Дополнительно: исследовать возможность использования алгоритма для цифровой подписи.
3. **Реализация алгоритма Диффи-Хеллмана**:  
   Написать программу для обмена ключами между двумя сторонами с использованием алгоритма Диффи-Хеллмана.  
   Дополнительно: исследовать уязвимость к атаке "человек посередине" (MITM).

### ****2. Анализ и тестирование****

1. **Сравнение RSA и Эль-Гамаля**:  
   Провести сравнительный анализ RSA и Эль-Гамаля по следующим параметрам: скорость шифрования, размер ключа, криптостойкость.
2. **Исследование криптостойкости RSA**:  
   Провести анализ уязвимостей RSA к атакам (например, атака Винера или атака на малый модуль).
3. **Тестирование корректности реализации RSA**:  
   Написать тесты для проверки корректности реализации RSA, используя известные тестовые векторы (test vectors).

### ****3. Практические применения****

1. **Шифрование файлов с использованием RSA**:  
   Разработать приложение для шифрования и дешифрования файлов с использованием RSA.  
   Дополнительно: добавить возможность выбора размера ключа (1024, 2048, 4096 бит).
2. **Реализация безопасного обмена сообщениями**:  
   Создать клиент-серверное приложение для безопасного обмена сообщениями с использованием RSA для шифрования и Диффи-Хеллмана для обмена ключами.
3. **Интеграция с базами данных**:  
   Написать скрипт для шифрования чувствительных данных в базе данных (например, паролей или персональных данных) с использованием RSA.

### ****4. Исследовательские задания****

1. **Оптимизация RSA**:  
   Исследовать возможности оптимизации RSA для различных платформ (CPU, GPU) и сравнить производительность.
2. **Исследование влияния размера ключа на безопасность**:  
   Провести анализ, как размер ключа влияет на криптостойкость RSA и Эль-Гамаля.
3. **Исследование гибридных схем шифрования**:  
   Разработать гибридную схему шифрования, сочетающую асимметричный (RSA) и симметричный (AES) алгоритмы.

### ****5. Творческие и углублённые задания****

1. **Реализация упрощённого RSA**:  
   Разработать упрощённую версию RSA (например, с уменьшенным размером ключа) и исследовать её уязвимости.
2. **Визуализация работы RSA**:  
   Создать визуализацию процесса шифрования и дешифрования RSA (например, преобразование чисел на каждом этапе).
3. **Исследование альтернативных асимметричных алгоритмов**:  
   Реализовать и проанализировать альтернативные асимметричные алгоритмы (например, ECC — Elliptic Curve Cryptography) и сравнить их с RSA.

**Лабораторная работа №3**

**Анализ и применение хэш-функций**

### ****1. Базовые задания****

1. **Реализация хэш-функции MD5**:  
   Написать программу для вычисления хэш-значения с использованием алгоритма MD5.  
   Дополнительно: провести анализ коллизий.
2. **Реализация хэш-функции SHA-256**:  
   Разработать программу для вычисления хэш-значения с использованием алгоритма SHA-256.  
   Дополнительно: сравнить производительность с MD5.
3. **Сравнение хэш-функций**:  
   Провести сравнительный анализ MD5, SHA-1 и SHA-256 по следующим параметрам: скорость вычисления, длина хэша, криптостойкость.

### ****2. Анализ и тестирование****

1. **Исследование коллизий в MD5**:  
   Провести эксперимент по поиску коллизий в MD5 (например, с использованием метода "birthday attack").
2. **Тестирование корректности реализации SHA-256**:  
   Написать тесты для проверки корректности реализации SHA-256, используя известные тестовые векторы (test vectors).
3. **Анализ устойчивости хэш-функций к атакам**:  
   Исследовать устойчивость MD5, SHA-1 и SHA-256 к атакам (например, атака на удлинение сообщения).

### ****3. Практические применения****

1. **Проверка целостности файлов**:  
   Разработать программу для вычисления и проверки хэш-сумм файлов (например, с использованием SHA-256).
2. **Хэширование паролей**:  
   Написать программу для безопасного хэширования паролей с использованием алгоритма bcrypt или PBKDF2.  
   Дополнительно: добавить "соль" (salt) для повышения безопасности.
3. **Реализация HMAC**:  
   Разработать программу для вычисления HMAC (Hash-based Message Authentication Code) с использованием SHA-256.

### ****4. Исследовательские задания****

1. **Исследование влияния длины входных данных на хэш**:  
   Провести эксперимент, как длина входных данных влияет на время вычисления хэш-функции.
2. **Анализ лавинного эффекта**:  
   Исследовать лавинный эффект в хэш-функциях (например, как небольшое изменение входных данных влияет на хэш-значение).
3. **Исследование гибридных схем хэширования**:  
   Разработать гибридную схему хэширования, сочетающую несколько алгоритмов (например, MD5 и SHA-256).

### ****5. Творческие и углублённые задания****

1. **Реализация упрощённой хэш-функции**:  
   Разработать упрощённую версию хэш-функции (например, с уменьшенным количеством раундов) и исследовать её уязвимости.
2. **Визуализация работы хэш-функции**:  
   Создать визуализацию процесса вычисления хэш-значения (например, преобразование данных на каждом этапе).
3. **Исследование альтернативных хэш-функций**:  
   Реализовать и проанализировать альтернативные хэш-функции (например, Blake2, Keccak) и сравнить их с SHA-256.

**Лабораторная работа №4**

**Разработка системы цифровой подписи на основе криптографических алгоритмов**

### ****1. Базовые задания****

1. **Реализация цифровой подписи на основе RSA**:  
   Написать программу для генерации ключей, создания и проверки цифровой подписи с использованием алгоритма RSA.  
   Дополнительно: реализовать padding (например, PSS).
2. **Реализация цифровой подписи на основе Эль-Гамаля**:  
   Разработать программу для создания и проверки цифровой подписи с использованием алгоритма Эль-Гамаля.  
   Дополнительно: исследовать возможность использования для шифрования.
3. **Реализация цифровой подписи на основе ECDSA**:  
   Написать программу для генерации ключей, создания и проверки цифровой подписи с использованием алгоритма ECDSA (Elliptic Curve Digital Signature Algorithm).

### ****2. Анализ и тестирование****

1. **Сравнение RSA и ECDSA**:  
   Провести сравнительный анализ RSA и ECDSA по следующим параметрам: скорость работы, размер ключа, криптостойкость.
2. **Исследование криптостойкости RSA-подписи**:  
   Провести анализ уязвимостей RSA-подписи к атакам (например, атака на малый модуль или атака Винера).
3. **Тестирование корректности реализации ECDSA**:  
   Написать тесты для проверки корректности реализации ECDSA, используя известные тестовые векторы (test vectors).

### ****3. Практические применения****

1. **Подпись документов**:  
   Разработать приложение для создания и проверки цифровой подписи текстовых документов с использованием RSA или ECDSA.  
   Дополнительно: добавить поддержку формата PDF или XML.
2. **Реализация системы аутентификации**:  
   Создать систему аутентификации на основе цифровой подписи (например, для входа в приложение).
3. **Интеграция с базами данных**:  
   Написать скрипт для подписи и проверки данных в базе данных (например, транзакций или записей).

### ****4. Исследовательские задания****

1. **Исследование влияния размера ключа на безопасность**:  
   Провести анализ, как размер ключа влияет на криптостойкость RSA и ECDSA.
2. **Анализ лавинного эффекта в цифровых подписях**:  
   Исследовать, как небольшое изменение входных данных влияет на цифровую подпись.
3. **Исследование гибридных схем подписи**:  
   Разработать гибридную схему подписи, сочетающую асимметричные и симметричные алгоритмы.

### ****5. Творческие и углублённые задания****

1. **Реализация упрощённой цифровой подписи**:  
   Разработать упрощённую версию цифровой подписи (например, с уменьшенным размером ключа) и исследовать её уязвимости.
2. **Визуализация работы цифровой подписи**:  
   Создать визуализацию процесса создания и проверки цифровой подписи (например, преобразование данных на каждом этапе).
3. **Исследование альтернативных алгоритмов подписи**:  
   Реализовать и проанализировать альтернативные алгоритмы цифровой подписи (например, EdDSA или Schnorr) и сравнить их с RSA и ECDSA.

**Лабораторная работа №5**

**Разработка системы цифровой подписи на основе криптографических алгоритмов**

**1. Базовые задания**

1. **Реализация атаки полным перебором (brute-force)**:  
   Написать программу для подбора пароля методом полного перебора.  
   *Дополнительно*: оценить время подбора для паролей разной длины.
2. **Реализация атаки по словарю**:  
   Разработать программу для подбора пароля с использованием словаря популярных паролей.  
   *Дополнительно*: исследовать эффективность словаря для разных типов паролей.
3. **Сравнение brute-force и атаки по словарю**:  
   Провести сравнительный анализ методов полного перебора и атаки по словарю по следующим параметрам: время подбора, эффективность.

**2. Анализ и тестирование**

1. **Исследование влияния длины пароля на стойкость**:  
   Провести эксперимент по подбору паролей разной длины (от 4 до 12 символов) и оценить время подбора.
2. **Исследование влияния сложности пароля на стойкость**:  
   Провести эксперимент по подбору паролей разной сложности (буквы, цифры, спецсимволы) и оценить время подбора.
3. **Тестирование эффективности радужных таблиц**:  
   Написать программу для генерации и использования радужных таблиц для подбора хэшей паролей.  
   *Дополнительно*: сравнить время подбора с brute-force.

**3. Практические применения**

1. **Подбор паролей к архивам**:  
   Разработать программу для подбора паролей к ZIP-архивам с использованием методов перебора и словаря.
2. **Подбор паролей к хэшам**:  
   Написать программу для подбора паролей по заданным хэшам (например, MD5 или SHA-256) с использованием радужных таблиц.
3. **Интеграция с базами данных**:  
   Написать скрипт для проверки стойкости паролей, хранящихся в базе данных, с использованием методов перебора.

**4. Исследовательские задания**

1. **Исследование эффективности радужных таблиц**:  
   Провести анализ эффективности радужных таблиц для разных алгоритмов хэширования (MD5, SHA-1, SHA-256).
2. **Анализ влияния salt на стойкость паролей**:  
   Исследовать, как использование **salt** влияет на сложность подбора паролей с использованием радужных таблиц.
3. **Исследование гибридных методов подбора**:  
   Разработать гибридный метод подбора паролей, сочетающий brute-force и атаку по словарю.

**5. Творческие и углублённые задания**

1. **Реализация оптимизированного brute-force**:  
   Разработать оптимизированную версию brute-force (например, с использованием многопоточности или GPU) и сравнить её производительность с классическим brute-force.
2. **Визуализация процесса подбора паролей**:  
   Создать визуализацию процесса подбора паролей (например, отображение перебираемых комбинаций в реальном времени).
3. **Исследование альтернативных методов подбора**:  
   Реализовать и проанализировать альтернативные методы подбора паролей (например, генетические алгоритмы) и сравнить их с brute-force и атакой по словарю.

**Лабораторная работа №6**

**Стеганография**

### ****1. Базовые задания****

1. **Реализация LSB-стеганографии в изображениях**:  
   Написать программу для скрытия текстового сообщения в изображении с использованием метода LSB (Least Significant Bit).  
   Дополнительно: реализовать извлечение сообщения.
2. **Реализация стеганографии в аудиофайлах**:  
   Разработать программу для скрытия текстового сообщения в аудиофайле с использованием LSB.  
   Дополнительно: исследовать влияние на качество звука.
3. **Сравнение LSB и DCT-методов**:  
   Провести сравнительный анализ методов LSB и DCT (Discrete Cosine Transform) для скрытия данных в изображениях.

### ****2. Анализ и тестирование****

1. **Исследование устойчивости LSB-метода**:  
   Провести эксперимент по обнаружению скрытых данных в изображении с использованием анализа гистограмм.
2. **Тестирование корректности реализации стеганографии**:  
   Написать тесты для проверки корректности скрытия и извлечения данных с использованием LSB.
3. **Анализ влияния размера сообщения на качество носителя**:  
   Исследовать, как размер скрываемого сообщения влияет на качество изображения или аудиофайла.

### ****3. Практические применения****

1. **Скрытие данных в текстовых файлах**:  
   Разработать программу для скрытия сообщения в текстовом файле с использованием пробелов и невидимых символов.
2. **Создание стеганографического чата**:  
   Написать клиент-серверное приложение для обмена сообщениями, скрытыми в изображениях.
3. **Интеграция с базами данных**:  
   Написать скрипт для скрытия данных в изображениях, хранящихся в базе данных.

### ****4. Исследовательские задания****

1. **Исследование методов обнаружения стеганографии**:  
   Провести анализ методов обнаружения скрытых данных (например, статистический анализ или машинное обучение).
2. **Анализ устойчивости к сжатию**:  
   Исследовать, как сжатие изображения (например, JPEG) влияет на сохранность скрытых данных.
3. **Исследование гибридных методов стеганографии**:  
   Разработать гибридный метод стеганографии, сочетающий LSB и DCT.

### ****5. Творческие и углублённые задания****

1. **Реализация стеганографии в видео**:  
   Разработать программу для скрытия данных в видеофайле с использованием LSB.  
   Дополнительно: исследовать влияние на качество видео.
2. **Визуализация процесса стеганографии**:  
   Создать визуализацию процесса скрытия данных в изображении (например, отображение изменённых пикселей).
3. **Исследование альтернативных методов стеганографии**:  
   Реализовать и проанализировать альтернативные методы стеганографии (например, использование нейронных сетей) и сравнить их с LSB.