Задания к работе №2 по основам криптографии.

Все задания выполняются на объектно-ориентированном языке программирования.

Применение готовых реализаций алгоритмов защиты информации и библиотек, содержащих такие реализации, не допускается. Допускается использование существующих (входящих в ядро и сторонних) реализаций длинных целых и длинных вещественных чисел.

1. Спроектируйте и реализуйте stateless-сервис с компонентным функционалом для:
   - вычисления символа Лежандра;
   - вычисления символа Якоби;
   - вычисления НОД двух целых чисел при помощи алгоритма Евклида;
   - вычисления НОД двух целых чисел и решения соотношения Безу при помощи расширенного алгоритма Евклида;
   - выполнения операции возведения в степень по модулю.

   Для реализованного сервиса не допускается адаптирование функционала, предоставляемого ядром используемого ЯП и сторонними библиотеками.

   Продемонстрируйте работу реализованного функционала.

2. Спроектируйте интерфейс, предоставляющий описание функционала для вероятностного теста простоты (параметры метода: тестируемое значение, минимальная вероятность простоты в диапазоне [0.5, 1)). На базе спроектированного интерфейса и поведенческого паттерна проектирования “Шаблонный метод” реализуйте базовый абстрактный класс для вероятностного теста простоты, с возможностью кастомизации поведения одной итерации теста. С использованием сервиса, реализованного в задании 1, пронаследуйте базовый класс для реализации следующих вероятностных тестов простоты: Ферма, Соловея-Штрассена, Миллера-Рабина.

3. Спроектируйте и реализуйте объектный сервис, предназначенный для выполнения шифрования и дешифрования данных алгоритмом RSA. Сервис должен содержать объект вложенного (nested) сервиса для генерации ключей алгоритма RSA (контракт конструктора вложенного сервиса: используемый тест простоты (задаётся перечислением, тип которого является nested по отношению к типу сервиса для выполнения шифрования/дешифрования алгоритмом RSA), минимальная вероятность простоты в диапазоне [0.5, 1), битовая длина генерируемых проверяемых выбранным тестом простоты псевдослучайных чисел; параметры делегируются из конструктора сервиса-обёртки). При генерации ключей

обеспечьте невозможность применимости атаки Ферма и атаки Винера. Новую ключевую пару можно генерировать произвольное количество раз. Продемонстрируйте выполнение шифрования и дешифрования данных алгоритмом RSA посредством реализованных сервисов.

4. Для сервиса, реализованного в задании 3, реализуйте адаптер, позволяющий использовать алгоритм RSA для потокового шифрования (см. Задания к работе №1 по основам криптографии, задание 4). При выполнении операций шифрования обеспечьте невозможность применимости атаки Хастада.

5. Реализуйте сервис, предоставляющий компонентный функционал для выполнения атаки Ферма на открытый ключ алгоритма RSA. Для данного открытого ключа в качестве результата выполнения атаки необходимо получить найденное значение дешифрующей экспоненты, а также значение функции Эйлера от модуля RSA.

6. Реализуйте сервис, предоставляющий компонентный функционал для выполнения атаки Винера на открытый ключ алгоритма RSA. Для данного открытого ключа в качестве результата выполнения атаки необходимо получить найденное значение дешифрующей экспоненты, значение функции Эйлера от модуля RSA, а также коллекцию вычисленных во время атаки подходящих дробей для дроби, построенной из компонентов открытого ключа.