Лабораторная работа 2 по основам криптографии.

1. Реализуйте stateless-сервис, предоставляющий объектный функционал для:
   - сложения двоичных полиномов (далее - элементов) из $GF(2^8)$;
   - умножения элементов из $GF(2^8)$ по заданному модулю;
   - взятия обратного элемента для элемента из $GF(2^8)$ по заданному модулю;
   - проверки двоичного полинома степени 8 на неприводимость над $GF(2^8)$;
   - генерации коллекции произвольного доступа всех неприводимых над $GF(2^8)$ двоичных полиномов степени 8.

   При попытке выполнения операции по приводимому над полем модулю, генерируйте (и перехватывайте в вызывающем коде) исключительную ситуацию. Значения элементов из $GF(2^8)$ и модулей над $GF(2^8)$ передавайте и возвращайте в виде однобайтовых значений (byte, [unsigned] char, …). При реализации функционала сервиса максимизируйте использование битовых операций.

2. На базе интерфейсов 3.1, 3.2, 3.3 из лабораторной работы 1 реализуйте класс, функционал которого позволяет выполнять [де]шифрование блока данных алгоритмом Rijndael при помощи вычисленных единожды из ключа шифрования, раундовых ключей. Реализация алгоритма должен поддерживать работу с блоками длиной 128/192/256 бит и ключами длиной 128/192/256 бит, а также предоставлять возможность настройки неприводимого над $GF(2^8)$ полинома.

3. Добавьте в класс из задания 2 внутренний функционал, позволяющий выполнять отложенные вычисления S-матриц, с их последующим использованием в трансформации SubBytes. Учтите, что конфигурирование матриц зависит от выбранного над $GF(2^8)$ неприводимого полинома.

4. Продемонстрируйте выполнение шифрования и дешифрования псевдослучайных последовательностей байтов и файлов (текстовых, музыкальных, изображений, видео) алгоритмом Rijndael с использованием различных режимов шифрования (используйте класс-контекст 3.4 из лабораторной работы 1), различных длин блока и ключа, а также с использованием различных неприводимых над $GF(2^8)$ двоичных полиномов степени 8.

5. Добавьте в класс-контекст 3.4 из лабораторной работы 1 возможность настройки типа набивки (PKCS7, ISO 10126, ANSI X.923) через объект перечисления, передаваемый в конструктор. Продемонстрируйте добавленные в реализацию класса-контекста режимы набивки посредством выполнения последовательно шифрования и дешифрования операций алгоритмами DES и Rijndael с различными режимами шифрования.