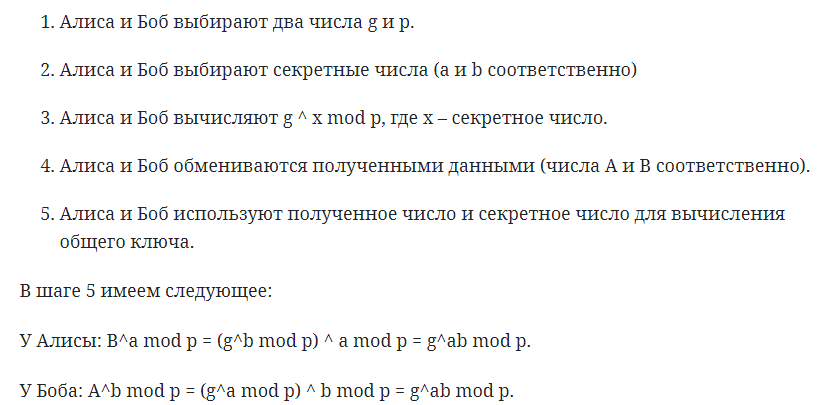
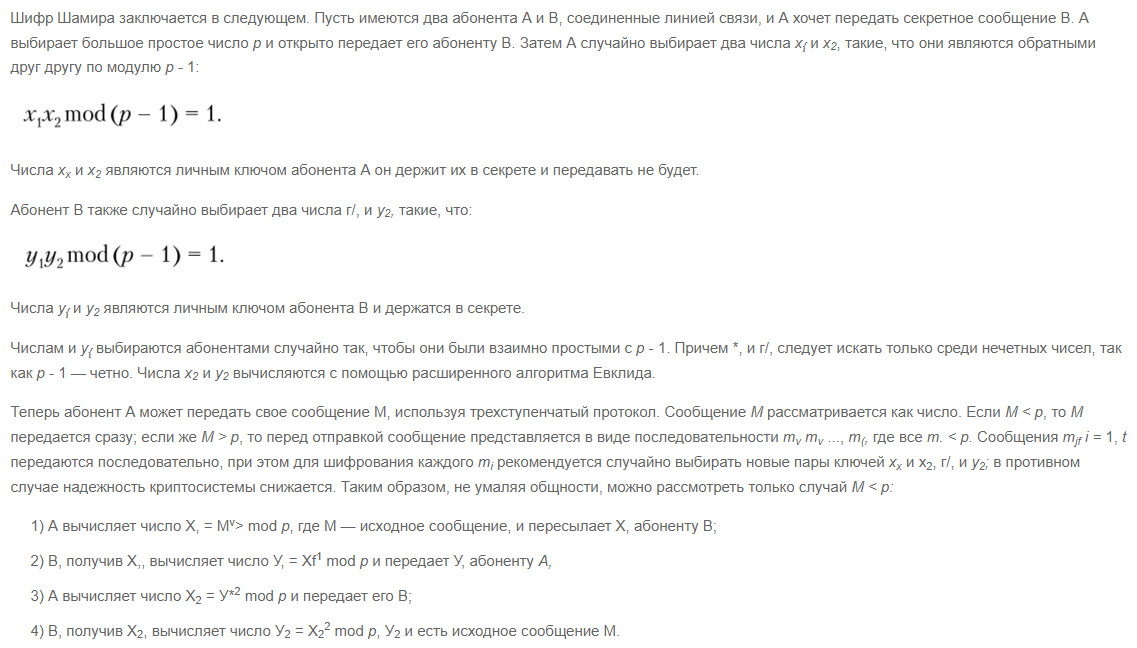
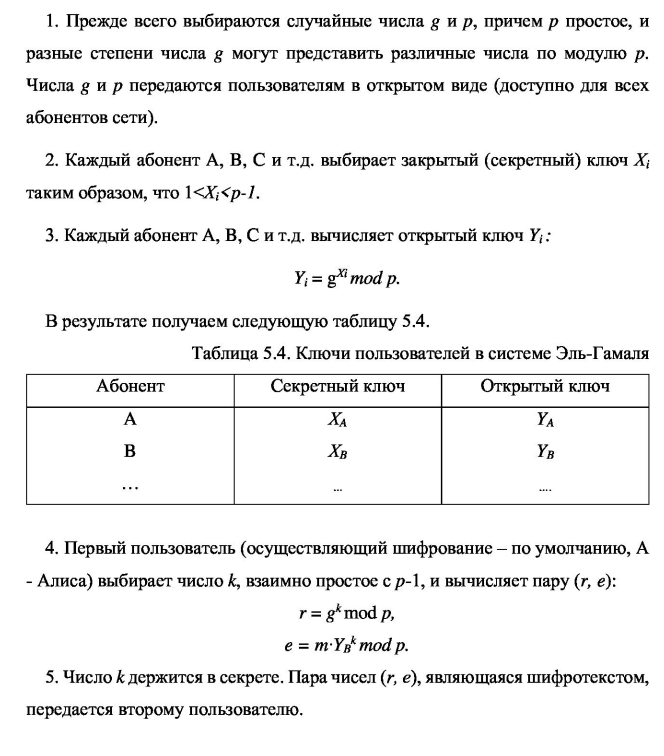
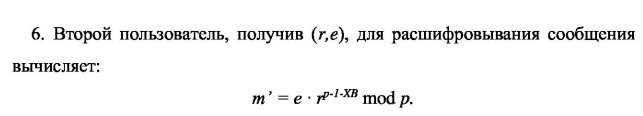
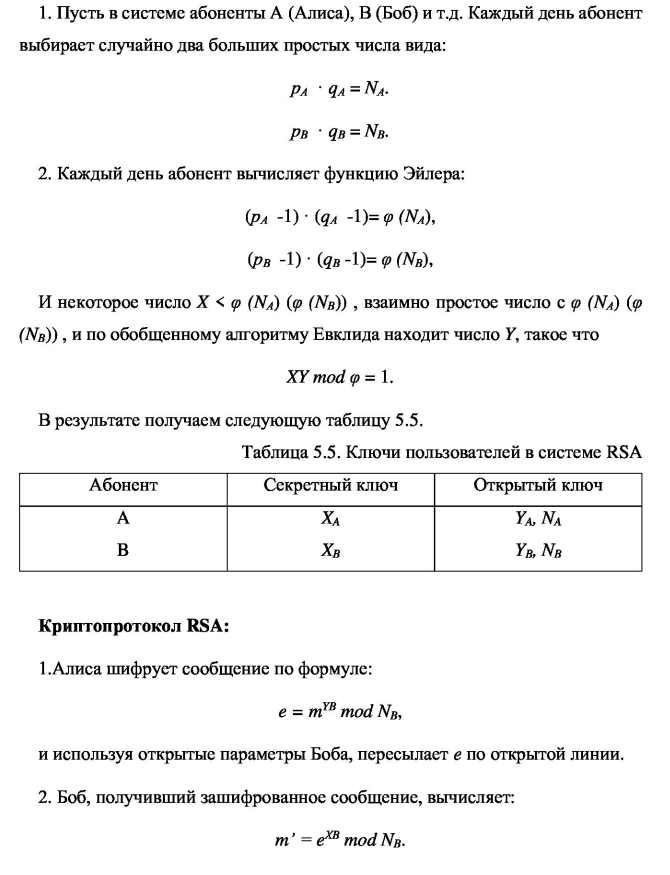
1. **Абстрактный тип данных**: Итак, что же будет, если слить понятия “тип данных” и “абстракция” воедино? Мы получим тип данных, который предоставляет нам некий набор операций, обеспечивающих поведение объектов этого типа данных, а также этот тип данных будет скрывать те данные, с помощью которых реализовано данное поведение.
2. **Класс** – это способ описания сущности, определяющий состояние и поведение, зависящее от этого состояния, а также правила для взаимодействия с данной сущностью. С точки зрения программирования класс можно рассматривать как набор данных и функций для работы с ними (методов). С точки зрения структуры программы, класс является сложным типом данных.  
   Зачем нужно: классы нужны для создания своих структур данных, которые будут содержать какую-то логику обработки. Вся логика хранится в описании класса, при этом оставляя в вызывающей программе лаконичные вызовы, без лишнего кода.
3. **Секции класса:**  
   Секция «public» (общедоступный). К элементам этой секции можно обращаться из любого модуля. Доступ возможен из методов этого класса, его потомков и вообще из любых процедур.  
   Секция «protected» (защищённый). К элементам этой секции можно обратиться из модуля, в котором описан класс. В другом модуле доступ возможен из методов «дружественного» класса или его потомков.  
   Секция «private» (частный). К элементам этой секции обратиться из других модулей нельзя. Доступ возможен из методов этого же класса, его потомков и процедур, расположенных в данном модуле.
4. **Односторонняя функция** — математическая функция, которая легко вычисляется для любого входного значения, но трудно найти аргумент по заданному значению функции.
5. **Асимметричное шифрование** — это метод шифрования данных, предполагающий использование двух ключей — открытого и закрытого. Открытый (публичный) ключ применяется для шифрования информации и может передаваться по незащищенным каналам. Закрытый (приватный) ключ применяется для расшифровки данных, зашифрованных открытым ключом.
6. **Криптографический протокол** — это абстрактный или конкретный протокол, включающий набор криптографических алгоритмов, часто являющихся последовательностью криптографических примитивов.
7. **Диффи-Хеллман**
8. **Шифр Шамира**
9. **Шифр Эль Гамаля**
10. **Шифр RSA**  
    

1. **Принципы создания объектной модели:**  
   Классы могут строится по ***иерархическому принципу***, когда один класс может быть подклассом другого класса. Используется при разработке структуры классов, когда сложные классы строятся на базе более простых путем добавления к ним новых характеристик и, возможно, уточнения имеющихся.   
   Принцип **инкапсуляции**характеризует сокрытие деталей внутреннего устройства классов от внешних по отношению к нему объектов или пользователей. То есть, взаимодействующему с классом объекту или пользователю не нужно знать, каким образом реализован тот или иной метод класса, чтобы им воспользоваться.  
   ***Модульность*** — это свойство системы, связанное с возможностью ее декомпозиции на ряд внутренне сильно сцепленных, но слабо связанных между собой подсистем (модулей).  
   **Абстрагирование** - процесс выделения абстракций в предметной области задачи.  
   ***Типизация*** — это ограничение, накладываемое на свойства объектов и препятствующее взаимозаменяемости абстракций различных типов.  
   ***Сохраняемость*** - это способность объекта существовать во времени, переживая породивший его процесс, и (или) в пространстве, перемещаясь из своего первоначального адресного пространства.  
   ***Параллелизм*** — наличие в системе нескольких потоков управления одновременно. Объект может быть активен, т. е. может порождать отдельный поток управления. Различные объекты могут быть активны одновременно.
2. **Экземпляр класса** – переменная, обладающая типом класса, благодаря которой мы взаимодействуем с классом.