# Лабораторная работа №11

**Изучение методов авторизации и аутентификации в настольных приложениях**

1. **Цель работы**
   1. Познакомиться с методами авторизации и аутентификации в настольных приложениях.
2. **Литература**
   1. Зверева В. П., Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. Образования / В. П. Зверева, А. В. Назаров. – М. : Издательский центр «Академия», 2018. – 256 с.
3. **Подготовка к работе**
   1. Повторить теоретический материал (см. п.2).
   2. Изучить описание лабораторной работы.
4. **Основное оборудование**
   1. Персональный компьютер.
5. **Задание**
   1. Разработать простое оконное приложение, использующее авторизацию пользователя через БД с распределением прав на три роли (роли определите самостоятельно)
   2. Разработать простое оконное приложение, использующее аутентификацию пользователя с использованием аутентификации Windows
   3. Модифицировать программу из п.5.1 так, чтобы приложение использовало для авторизации вместо пароля файл ключа ассиметричного шифрования (см. п.9)
   4. Составить отчет по проделанной работе.
6. **Порядок выполнения работы**
   1. Повторить теоретический материал п. 3.1;
   2. Выполнить задания 5.1-5.8
   3. Ответить на контрольные вопросы п. 8;
   4. Заполнить отчет п. 7.
7. **Содержание отчета**
   1. Титульный лист;
   2. Цель работы;
   3. Таблица п. 5.2
   4. Ответы на контрольные вопросы п. 6.3;
   5. Вывод по проделанной работе.
8. **Контрольные вопросы**
   1. Что такое аутентификация?
   2. Что такое авторизация?
9. **Приложение**

При регистрации пользователя используется алгоритм ассиметричного шифрования RSA. Публичным ключом шифруется некоторое сообщение и эти данные записывается в хранилище. Для авторизации необходимо расшифровать данное сообщение с использованием закрытого ключа, и если расшифровка прошла успешно, то пользователь авторизован.

Листинг 1 – Примерная логика регистрации и авторизации

|  |
| --- |
| // Генерация ключей  using (var rsa = RSA.Create())  {  byte[] publicKey = rsa.ExportRSAPublicKey();  byte[] privateKey = rsa.ExportRSAPrivateKey();  // Сохранение приватного ключа в файл  File.WriteAllBytes(privateKeyFilePath, privateKey);  // Генерация тестового токена для проверки при авторизации  byte[] testToken = RandomNumberGenerator.GetBytes(32);  // Шифрование тестового токена публичным ключом  rsa.ImportRSAPublicKey(publicKey, out \_);  byte[] encryptedToken = rsa.Encrypt(testToken, RSAEncryptionPadding.Pkcs1);  // Сохранение данных в базу  SaveUserToDatabase(username, publicKey, encryptedToken);  Console.WriteLine("Регистрация завершена. Сохраните приватный ключ.");  //Проверка валидности ключа путем расшифровки токена  try{  byte[] decryptedToken = rsa.Decrypt(testToken, RSAEncryptionPadding.Pkcs1);  Console.WriteLine("Успешная авторизация");  catch{  Console.WriteLine("Ошибка авторизации");  }  } |