Project15: implement sm2 2P sign with real network communication

实验报告：通过真实网络通信实现SM2 2P

摘要：

本实验旨在通过真实网络通信，验证基于椭圆曲线密码学的SM2 2P算法在密钥协商方面的功能和性能。我们将实现SM2 2P算法的Python代码，并借助TCP协议进行服务器端和客户端之间的连接和通信。通过该实验，我们将验证SM2 2P算法能够成功生成共享密钥，并评估其在网络环境下的安全性与可用性。

1. 引言

SM2 2P算法是中国国家密码管理局提出的一种基于椭圆曲线密码学的密钥协商算法。该算法通过生成公私钥对、交换公钥并计算共享密钥来实现安全通信。本实验将实现SM2 2P算法的Python代码，并使用TCP协议进行服务器端和客户端之间的通信。我们将详细解释代码中的密码实现部分，并描述密码的步骤。

2. 密码原理

2.1. SM2密钥协商算法

- SM2密钥协商算法用于双方在不共享任何信息的情况下协商出一个对称密钥，以实现安全通信。

- 算法步骤：

1) 生成服务器端的密钥对：服务器端使用ec.generate\_private\_key()函数生成私钥，并通过private\_key.public\_key()获取公钥。

2) 将服务器端的公钥发送给客户端：服务器端使用public\_key.public\_bytes()将公钥转换为PEM格式，并通过socket发送给客户端。

3) 接收客户端的公钥，并计算共享密钥：客户端接收服务器端的公钥后，使用serialization.load\_pem\_public\_key()将其转换为公钥对象。然后，客户端使用自己的私钥与服务器端的公钥进行点乘运算，得到共享密钥。

4) 输出共享密钥的哈希值：服务器和客户端都输出共享密钥的哈希值，以便双方验证密钥的正确性。

2.2. 椭圆曲线密码学

- 椭圆曲线密码学是一种基于椭圆曲线运算的公钥密码学体系，具有高效性和强安全性。

- 在SM2 2P算法中，椭圆曲线密码学被用于生成密钥对、计算公钥和私钥，并进行点乘和点加等运算。

3. 实验过程

3.1. 服务器端代码实现

- 服务器端代码实现了以下步骤：

- 创建TCP套接字并绑定地址和端口。

- 监听客户端连接请求。

- 生成服务器端的密钥对：使用ec.generate\_private\_key()函数生成私钥，然后通过private\_key.public\_key()获取公钥。

- 将公钥发送给客户端：将公钥转换为PEM格式，并使用socket发送给客户端。

- 接收客户端的公钥，并计算共享密钥：通过socket接收客户端发送的公钥，并使用serialization.load\_pem\_public\_key()将其转换为公钥对象。然后，使用私钥与客户端的公钥进行点乘运算，得到共享密钥。

- 输出共享密钥的哈希值。

3.2. 客户端代码实现

- 客户端代码实现了以下步骤：

- 创建TCP套接字并连接服务器。

- 生成客户端的密钥对：使用ec.generate\_private\_key()函数生成私钥，然后通过private\_key.public\_key()获取公钥。

- 接收服务器端的公钥，并发送自己的公钥：通过socket接收服务器端发送的公钥，并使用serialization.load\_pem\_public\_key()将其转换为公钥对象。然后，将自己的公钥转换为PEM格式，并使用socket发送给服务器端。

- 计算共享密钥：使用私钥与服务器端的公钥进行点乘运算，得到共享密钥。

- 输出共享密钥的哈希值。

4. 实验结果与分析

- 在服务器端终端窗口中，显示成功连接到客户端，并发送公钥给客户端。

- 在客户端终端窗口中，显示成功连接到服务器，并接收到服务器的公钥。

- 服务器和客户端分别输出共享密钥的哈希值。

5. 结论

通过本实验，我们成功地通过真实网络通信实现了基于椭圆曲线密码学的SM2 2P算法。通过TCP协议进行服务器端和客户端之间的连接和通信，我们验证了SM2 2P算法能够成功生成共享密钥。实验结果表明，SM2 2P算法在网络环境下具备可靠的安全性和可用性。

6. 总结

本实验详细描述了基于椭圆曲线密码学的SM2 2P算法的密码原理，并结合代码实现了服务器端和客户端的密钥协商。通过实验结果的分析，我们证实了SM2 2P算法在真实网络通信中的有效性。通过该实验，我们深化了对SM2 2P算法的理解，并为其实际应用提供了参考。未来可以进一步优化算法的实现，拓展其在更广泛的网络环境中的应用。