山东大学 软件 学院

Python密码学编程课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100150209 | 姓名： 杨佳庆 | | 班级： 21级网安班 |
| 实验题目：实验5：利用MAC和数字签名实现消息认证 | | | |
| 实验学时：2 | | 实验日期： 2023.10.10 | |
| 实验目的：   * 掌握基于对称密码实现消息认证的MAC设计原理及应用 * 掌握基于非对称密码实现消息认证的数字签名设计原理及应用 * 了解数字证书的结构和作用 | | | |
| 硬件环境：  PC机 | | | |
| 软件环境：  PyCharm | | | |
| 实验步骤与内容：   1. 利用cryptography中primitives的hashes、hmac模块实现给定消息的认证码生成及验证 2. 利用cryptography中primitives的cmac和algrithms.AES实现给定消息的认证码生成及验证 3. 利用cryptography中primitives的asymmetric.rsa生成给定消息的RSA数字签名并完成验证 4. 利用cryptography中primitives.asymmetric中的ec生成给定消息的椭圆曲线数字签名并完成验证 5. 利用Crypto库中的对应功能模块编程实现消息认证码和数字签名的生成和验证 6. 熟悉代码5-6，5-7，5-8，总结数字证书的内容结构、签发方式和作用。（可以在下次实验完成） | | | |
| 结论分析与体会：  在这个实验中，我们学习了如何使用Python的cryptography库实现非对称加密的一些基本操作。主要包括密钥生成、密钥对象和字节流之间的转换、以及基于RSA的加解密算法。  在密钥生成部分，我们使用了cryptography库中的rsa.generate\_private\_key()函数生成了RSA密钥对。通过指定公钥指数（public\_exponent）和密钥长度（key\_size），我们可以生成不同的密钥对。  然后，我们学习了如何使用cryptography库中的serialization模块将RSA密钥对象转为字节流表示，并可以将字节流转换回密钥对象。这种转换非常重要，因为密钥在网络传输或存储时通常以字节流的形式进行。  接下来，我们学习了如何使用gmpy2库和cryptography库中的RSA模块来实现简单的RAW-RSA加和解密。注意，这是一个非常不安全且易受攻击的实现。在实际应用中，我们应该使用更安全的填充方案来保护RSA加密。  在下一部分，我们学习了如何使用cryptography库中的PKCS1\_OAEP填充案来实现更安全的RSA加解密。PKCS1\_OAEP填充方案提供了更高的安全性，并且防止一些常见的RSA加密攻击。在实践中，我们应该使用更安全的加密库和填充方案来保护加密通信。  在最后两个部分，我们学习了共模攻击和同态攻击的概念。这些是高级的攻击技术，可以利用特定条件下的RSA实现中的漏洞来实现攻击。  通过这个实验，我们深入了解了非对称加密的基本原理、密钥、加解密操作、填充方案的选择等方面。我们还学习了一些基本的攻击技术和如何使用cryptography库来实现安全的加密操作。对于进一步加强我们对非对称加密算法的理解和应用具有很大帮助。 | | | |