<<精通PKI网络安全认证技术与编程实>>

图书基本信息

书名：<<精通PKI网络安全认证技术与编程实现>>

13位ISBN编号：9787115178459

10位ISBN编号：7115178453

出版时间：2008-7-1

出版时间：人民邮电出版社

作者：马臣云

页数：452

字数：766000

内容概要

PKI是解决开放式互联网络信息安全需求的成熟体系。
PKI体系支持身份认证，信息传输、存储的完整性，消息传输、存储的机密性，以及操作的不可否认性。
本书从实战出发，介绍了PKI应用开发过程和细节。
全书共32章，分6篇，主要内容包括PKI基础知识、OpenSSL开发、CrytoAPI开发、Java Security开发、电子商务网站应用、PKI技术应用等，涉及C语言、Java语言、JSP、ASP/ASP.NET、PHP等开发语言。
为了方便读者深入了解PKI，本书按照先原理、再讲解、再实战的方式进行，并且全部实例和软件都保存在随书赠送的光盘中。
本书适合PKI应用开发人员、企业网络管理人员以及大、中专院校师生阅读。

<<精通PKI网络安全认证技术与编程实>>

书籍目录

第1篇 PKI技术概述 第1章 PKI基础知识 1.1 PKI概述 1.2 什么是数字证书 1.2.1 数字认证的原理 1.2.2 数字认证是如何颁发的 1.3 为什么要使用数字证书 1.3.1 信息传输的保密性 1.3.2 交易者身份的确定性 1.3.3 发送信息的不可否认性 1.3.4 数据交换的完整性 1.4 加密技术 1.4.1 对称加密技术 1.4.2 非对称加密技术 1.5 数字签名技术 1.5.1 数字签名技术 1.5.2 时间戳技术 第2篇 OpenSSL开发 第2章 OpenSSL入门 2.1 OpenSSL概述 2.1.1 OpenSSL的组成 2.1.2 OpenSSL的优缺点 2.2 如何下载编译 2.2.1 Windows下编译OpenSSL 2.2.2 Linux下编译OpenSSL 2.3 如何搭建开发环境 2.3.1 Windows下搭建OpenSSL开发环境 2.3.2 Linux下搭建OpenSSL开发环境 2.4 小结 第3章 OpenSSL加密和解密 3.1 概述 3.2 函数介绍 3.2.1 初始化函数EVP_CIPHER_CTX_init 3.2.2 加密初始化函数EVP_EncryptInit_ex 3.2.3 数据加密Update函数EVP_EncryptUpdate 3.2.4 数据加密结束函数EVP_EncryptFinal_ex 3.2.5 解密初始化函数EVP_DecryptInit_ex 3.2.6 数据解密Update函数EVP_DecryptUpdate 3.2.7 数据解密结束函数EVP_DecryptFinal_ex 3.3 实例应用——数据加密 3.3.1 流程分析 3.3.2 实例实现 第4章 OpenSSL消息摘要 4.1 概述 4.2 函数介绍 4.2.1 初始化函数EVP_MD_CTX_init 4.2.2 设置摘要算法函数EVP_DigestInit_ex 4.2.3 摘要Update函数EVP_DigestUpdate 4.2.4 摘要结束函数EVP_DigestFinal_ex 4.2.5 计算摘要函数EVP_Digest 4.3 实例应用 4.3.1 流程分析 4.3.2 实例实现 第5章 OpenSSL签名和验证 5.1 函数介绍 5.1.1 签名初始化函数EVP_SignInit_ex 5.1.2 签名Update函数EVP_SignUpdate 5.1.3 签名结束函数EVP_SignFinal 5.1.4 验证初始化函数EVP_VerifyInit_ex 5.1.5 验证Update函数EVP_VerifyUpdate 5.1.6 验证结束函数EVP_VerifyFinal 5.2 实例应用 5.2.1 流程分析 5.2.2 实例实现 第6章 OpenSSL Base64编解和解码 6.1 函数介绍 6.1.1 Base64编码初始化函数EVP_EncodeInit 6.1.2 Base64编码Update函数EVP_EncodeUpdate 6.1.3 Base64编码结束函数EVP_EncodeFinal 6.1.4 Base64编码函数EVP_EncodeBlock 6.1.5 Base64解码函数EVP_DecodeBlock 6.1.6 Base64解码初始化函数EVP_DecodeInit 6.1.7 Base64解码Update函数EVP_DecodeUpdate 6.1.8 Base64解码结束函数EVP_DecodeFinal 6.2 实例应用 6.2.1 流程分析 6.2.2 实例实现 第7章 OpenSSL证书操作 7.1 函数介绍 7.1.1 DER编码转换为内部结构体函数d2i_X509 7.1.2 获得证书版本函数X509_get_version 7.1.3 获得证书序列号函数X509_get_serialNumber 7.1.4 获得证书颁发者信息函数X509_get_issuer_name 7.1.5 获得证书拥有者信息函数X509_get_subject_name 7.1.6 获得证书有效期的起始日期函数X509_get_notBefore 7.1.7 获得证书有效期的终止日期函数X509_get_notAfter 7.1.8 获得证书公钥函数X509_get_pubkey 7.1.9 创建和释放证书存储区函数X509_STORE_new、X509_STORE_free 7.1.10 向证书存储区添加证书函数X509_STORE_add_cert 7.1.11 向证书存储区添加证书吊销列表函数X509_STORE_add_crl 7.1.12 创建证书存储区上下文环境函数X509_STORE_CTX_new 7.1.13 释放证书存储区上下文环境函数X509_STORE_CTX_free 7.1.14 初始化证书存储区上下文环境函数X509_STORE_CTX_init 7.1.15 验证证书函数X509_verify_cert 7.2 实例应用 7.2.1 流程分析 7.2.2 实例实现 第8章 SSL/TLS编程 8.1 函数介绍 8.1.1 初始化SSL算法库函数SSL_library_init 8.1.2 初始化SSL上下文环境变量函数SSL_CTX_new 8.1.3 释放SSL上下文环境变量函数SSL_CTX_free 8.1.4 设置SSL证书函数SSL_CTX_use_certificate_file 8.1.5 设置SSL私钥函数SSL_CTX_use_PrivateKey_file 8.1.6 设置SSL证书函数SSL_CTX_use_certificate 8.1.7 设置SSL私钥函数SSL_CTX_use_PrivateKey 8.1.8 检查SSL私钥函数SSL_CTX_check_private_key 8.1.9 新建SSL句柄函数SSL_new 8.1.10 释放SSL句柄函

数SSL_free 8.1.11 设置socket句柄函数SSL_set_fd 8.1.12 建立SSL链接函数SSL_connect 8.1.13 接受SSL链接函数SSL_accept 8.1.14 获得SSL链接使用的证书SSL_get_peer_certificate 8.1.15 发送SSL数据函数SSL_write 8.1.16 读取SSL数据函数SSL_read 8.2 实例应用 8.2.1 流程分析 8.2.2 实例实现 第9章 开发实例——文件保险箱 9.1 功能预览 9.1.1 文件加密 9.1.2 文件解密 9.2 流程分析 9.2.1 文件加密函数Encrypt_File 9.2.2 文件解密函数Decrypt_File 9.3 功能实现 第10章 开发实例——安全通信软件 10.1 功能预览 10.2 流程分析 10.2.1 服务端流程分析 10.2.2 客户端流程分析 10.3 功能实现 10.3.1 服务端 10.3.2 客户端 第11章 开发实例——安全报文系统 11.1 功能预览 11.1.1 发送方产生安全报文 11.1.2 接收方解密安全报文 11.2 流程分析 11.2.1 发送方流程分析 11.2.2 接收方流程分析 11.3 功能实现 11.3.1 发送方 11.3.2 接收方 第3篇 CrytoAPI开发 第12章 CryptoAPI开发入门 12.1 CryptoAPI的组成 12.2 CryptoAPI的优缺点 12.3 如何搭建开发环境 第13章 密码服务提供者CSP函数 13.1 函数介绍 13.1.1 连接CSP函数CryptAcquireContext 13.1.2 枚举CSP函数CryptEnumProviders 13.1.3 获得默认CSP函数CryptGetDefaultProvider 13.1.4 设置默认CSP函数CryptSetProvider 13.1.5 获得CSP参数属性函数CryptGetProvParam 13.1.6 设置CSP参数函数CryptSetProvParam 13.1.7 断开CSP函数CryptReleaseContext 13.2 实例应用 13.2.1 流程分析 13.2.2 实例实现 第14章 密钥的产生和交换函数 14.1 函数介绍 14.1.1 生成函数CryptGenKey 14.1.2 派生密钥函数CryptDeriveKey 14.1.3 销毁密钥函数CryptDestroyKey 14.1.4 复制密钥函数CryptDuplicateKey 14.1.5 导出密钥函数CryptExportKey 14.1.6 导入密钥函数CryptImportKey 14.1.7 获得密钥参数函数CryptGetKeyParam 14.1.8 获得密钥参数函数CryptSetKeyParam 14.1.9 获得密钥参数函数CryptGenRandom 14.2 实例应用 14.2.1 流程分析 14.2.2 实例实现 第15章 数据的加密和解密函数 15.1 函数介绍 15.1.1 数据加密函数CryptEncrypt 15.1.2 数据解密函数CryptDecrypt 15.2 实例应用 15.2.1 流程分析 15.2.2 实例实现 第16章 哈希和数字签名函数 16.1 函数介绍 16.1.1 创建哈希函数CryptCreateHash 16.1.2 销毁哈希CryptDestroyHash 16.1.3 复制哈希函数CryptDuplicateHash 16.1.4 获得哈希参数函数CryptGetHashParam 16.1.5 设置哈希参数函数CryptSetHashParam 16.1.6 哈希会话密钥函数CryptHashSessionKey 16.1.7 哈希数据函数CryptHashData 16.1.8 对哈希签名函数CryptSignHash 16.1.9 对哈希验证签名函数CryptVerifySignature 16.2 实例应用 16.2.1 流程分析 16.2.2 实例实现 第17章 证书和证书库函数 17.1 函数介绍 17.1.1 打开证书库函数CertOpenStore 17.1.2 关闭证书库函数CertCloseStore 17.1.3 从证书库枚举证书函数CertEnumCertificatesInStore 17.1.4 从证书库查找证书函数CertFindCertificateInStore 17.1.5 创建证书句柄函数CertCreateCertificateContext 17.1.6 释放证书句柄函数CertFreeCertificateContext 17.1.7 获得证书句柄属性函数CertGetCertificateContextProperty 17.1.8 设置证书句柄属性函数CertSetCertificateContextProperty 17.1.9 获得证书主题名称函数CertGetNameString 17.2 实例应用 17.2.1 流程分析 17.2.2 实例实现 第18章 开发实例——文件保险箱 18.1 功能预览 18.1.1 文件加密 18.1.2 文件解密 18.2 流程分析 18.2.1 文件加密函数Encrypt_File 18.2.2 文件解密函数Decrypt_File 18.3 功能实现 第19章 开发实例—安全报文系统 19.1 功能预览 19.1.1 安全报文发送 19.1.2 安全报文接收 19.2 流程分析 19.2.1 发送方流程分析 19.2.2 接收方流程分析 19.3 功能实现 19.3.1 发送方 19.3.2 接收方 第4篇 Java Security开发 第20章 Java Security开发入门 20.1 设计原理和体系结构 20.1.1 设计原理 20.1.2 体系结构 20.2 主要概念 20.2.1 引擎类和算法 20.2.2 实现和提供者 20.2.3 获得实现实例的factory（工厂）方法 20.3 主要类和接口 20.4

搭建开发环境　第21章　Java消息摘要　21.1　MessageDigest类函数介绍　21.1.1　构造方法　21.1.2　生成实例对象函数getInstance（1）　21.1.3　生成实例对象函数getInstance（2）　21.1.4　获得密码服务提供者函数getProvider　21.1.5　计算摘要函数update（1）　21.1.6　计算摘要函数update（2）　21.1.7　计算摘要函数update（3）　21.1.8　计算摘要函数update（4）　21.1.9　完成计算摘要函数digest（1）　21.1.10　完成计算摘要函数digest（2）　21.1.11　完成计算摘要函数digest（3）　21.1.12　比较摘要值函数isEqual　21.1.13　对象重置函数reset　21.1.14　获得摘要算法函数getAlgorithm　21.1.15　获得摘要值长度函数getDigestLength　21.2　实例应用　21.2.1　流程分析　21.2.2　实例实现　第22章　Java加密和解密　22.1　KeyGenerator类函数介绍　22.1.1　构造方法　22.1.2　生成实例对象函数getInstance（1）　22.1.3　生成实例对象函数getInstance（2）　22.1.4　获得对象密码算法函数getAlgorithm　22.1.5　获得密码服务提供者函数getProvider　22.1.6　初始化密钥生成器函数init（1）　22.1.7　初始化密钥生成器函数init（2）　22.1.8　初始化密钥生成器函数init（3）　22.1.9　初始化密钥生成器函数init（4）　22.1.10　初始化密钥生成器函数init（5）　22.1.11　生成密钥函数generateKey　22.2　Cipher类函数介绍　22.2.1　构造方法　22.2.2　生成实例对象函数getInstance（1）　22.2.3　生成实例对象函数getInstance（2）　22.2.4　获得密码服务提供者函数getProvider　22.2.5　获得密码算法函数getAlgorithm　22.2.6　获得密码算法分组长度函数getBlockSize　22.2.7　获得输出数据的长度函数getOutputSize　22.2.8　获得初始化向量函数getIV　22.2.9　密码对象初始化函数init（1）　22.2.10　密码对象初始化函数init（2）　22.2.11　密码对象初始化函数init（3）　22.2.12　密码对象初始化函数init（4）　22.2.13　计算加密或解密函数update（1）　22.2.14　计算加密或解密函数update（2）　22.2.15　计算加密或解密函数update（3）　22.2.16　计算加密或解密函数update（4）　22.2.17　结束加密或解密函数doFinal（1）　22.2.18　结束加密或解密函数doFinal（2）　22.3　实例应用　22.3.1　流程分析　22.3.2　实例实现　第23章　Java数字签名和验证　23.1　KeyPairGenerator类函数介绍　23.1.1　构造方法　23.1.2　获得密码算法函数getAlgorithm　23.1.3　生成实例对象函数getInstance（1）　23.1.4　生成实例对象函数getInstance（2）　23.1.5　密码对象初始化函数initialize（1）　23.1.6　密码对象初始化函数initialize（2）　23.1.7　生成非对称密钥对函数genKeyPair和generateKeyPair　23.2　Signature类函数介绍　23.2.1　构造方法　23.2.2　获得签名对象算法函数getAlgorithm　23.2.3　生成实例对象函数getInstance（1）　23.2.4　生成实例对象函数getInstance（2）　23.2.5　初始化验证对象函数initVerify（1）　23.2.6　初始化验证对象函数initVerify（2）　23.2.7　初始化签名对象函数initSign（1）　23.2.8　初始化签名对象函数initSign（2）　23.2.9　更新签名或验证数据函数update（1）　23.2.10　更新签名或验证数据函数update（2）　23.2.11　更新签名或验证数据函数update（3）　23.2.12　签名函数sign（1）　23.2.13　签名函数sign（2）　23.2.14　验证签名函数verify（1）　23.2.15　验证签名函数verify（2）　23.3　实例应用　23.3.1　数字签名实现　23.3.2　数字签名验证实现　23.3.3　实例实现　第24章　keytool和证书类　24.1　keytool命令介绍　24.1.1　产生密钥对命令genkey　24.1.2　向密钥仓库导入证书命令import　24.1.3　导出证书请求命令certreq　24.1.4　导出证书命令export　24.1.5　枚举仓库数据命令list　24.1.6　管理密钥仓库命令storepasswd　24.1.7　管理密钥仓库命令keypasswd　24.1.8　管理密钥仓库命令delete　24.2　X509Certificate类函数介绍　24.2.1　构造方法　24.2.2　检查证书有效期函数checkValidity（1）　24.2.3　检查证书有效期函数checkValidity（2）　24.2.4　获得证书版本函数getVersion　24.2.5　获得证书序列号函数getSerialNumber　24.2.6　获得证书颁发者函数getIssuerX500Principal　24.2.7　获得证书主题信息函数getSubjectX500Principal　24.2.8　获得证书有效起始日期函数getNotBefore　24.2.9　获得证书有效期终止日期函数getNotAfter　24.2.10　获得DER编码的证书内容函数getTBSCertificate　24.2.11　获得证书签名值函数getSignature　24.2.12　获得证书签名算

法名称函数getSigAlgName 24.2.13 获得证书密钥用途函数getKeyUsage 24.3 X509CRL类函数介绍 24.3.1 构造方法 24.3.2 getEncoded 24.3.3 验证CRL签名函数verify 24.3.4 获得CRL版本函数getVersion 24.3.5 获得CRL颁发者函数getIssuerX500Principal 24.3.6 获得CRL本次更新时间函数getThisUpdate 24.3.7 获得CRL下次更新时间函数getNextUpdate 24.3.8 获得被吊销的证书函数getRevokedCertificate（1） 24.3.9 获得被吊销的证书函数getRevokedCertificate（2） 24.3.10 获得被吊销的证书函数getRevokedCertificate（3） 24.3.11 获得DER编码的CRL信息函数getTBSCertList 24.3.12 获得签名值函数getSignature 24.3.13 获得签名算法名称函数getSigAlgName 24.4 实例应用 24.4.1 流程分析 24.4.2 实例实现 第25章 Java开发实例—文件保险箱 25.1 功能预览 25.2 流程分析 25.3 功能实现 第26章 Java开发实例——安全报文系统 26.1 功能预览 26.1.1 安全报文发送 26.1.2 安全报文接收 26.2 流程分析 26.2.1 发送方流程分析 26.2.2 接收方流程分析 26.3 功能实现 26.3.1 密钥和证书keystore的生成方法 26.3.2 安全报文发送方 26.3.3 安全报文接收方 第5篇 PKI电子商务网站应用 第27章 ASP/ASP.Net电子商务网站应用 27.1 配置IIS的SSL服务器证书 27.1.1 生成证书请求 27.1.2 安装证书 27.1.3 启用SSL 27.2 基于数字证书的用户身份认证 27.2.1 基于数字证书的用户身份认证的方法 27.2.2 ASP/ASP.NET页面获取客户端证书的方法 27.3 数据签名处理—基于 CAPICOM的应用 27.3.1 CAPICOM简介 27.3.2 CAPICOM对象——Certificate对象 27.3.3 CAPICOM对象——Certificates对象 27.3.4 CAPICOM对象——CertificateStatus对象 27.3.5 CAPICOM对象——Store对象 27.3.6 CAPICOM对象——SignedData对象 27.3.7 CAPICOM对象——Signer对象 27.3.8 CAPICOM对象——Signers对象 27.3.9 CAPICOM对象——EnvelopedData对象 27.3.10 CAPICOM对象——Recipients对象 27.3.11 CAPICOM对象——Algorithm对象 27.3.12 CAPICOM对象——其他对象 27.3.13 如何在客户端安装部署和调用 27.3.14 如何在服务器端安装部署和调用 27.4 基于自开发控件应用 27.4.1 开发ActiveX控件 27.4.2 如何在客户端部署和调用 27.4.3 代码示例 27.5 开发实例——安全登录 27.5.1 登录处理页面（login.apsx.cs） 27.5.2 用户页面（main.apsx.cs） 27.5.3 出错显示页面（err.apsx.cs） 27.5.4 测试功能 27.6 开发实例——订单签名 27.6.1 签名页面前台（Sign.aspx） 27.6.2 签名页面后台（Sign.aspx.cs） 27.6.3 签证签名页面前台（verifySign.aspx） 27.6.4 验证签名后台页面（verifySign.aspx.cs） 27.7 小结 第28章 JSP电子商务网站应用 28.1 配置JSP Web服务器的SSL证书 28.1.1 生成证书请求文件（CSR） 28.1.2 导入证书 28.1.3 设置Tomcat支持SSL 28.1.4 使用浏览器访问SSL服务器 28.2 基于数字证书的用户身份认证 28.2.1 基于数字证书的用户身份认证的方法 28.2.2 JSP页面获取客户端证书的方法 28.3 数据签名处理 28.3.1 JSP前台提交签名 28.3.2 JSP后台处理签名 28.4 开发实例——安全登录 28.4.1 SSL登录处理页面（login.jsp） 28.4.2 用户主页面（main.jsp） 28.4.3 出错处理页面（err.jsp） 28.4.4 测试代码 28.5 开发实例——订单签名 28.5.1 签名页面（Sign.jsp） 28.5.2 验证签名页面（verifySign.jsp） 第29章 PHP电子商务网站应用 29.1 配置Apache的SSL证书 29.1.1 安装Apache+PHP+SSL 29.1.2 配置Apache的SSL证书 29.2 基于数字证书的用户身份认证 29.2.1 基于数字证书的用户身份认证的方法 29.2.2 PHP页面获取客户端证书的方法 29.3 数据签名处理 29.3.1 PHP前台提交签名 29.3.2 PHP后台处理签名 29.4 开发实例——安全登录 29.4.1 登录页面（login.php） 29.4.2 用户主页面（main.php） 29.4.3 出错处理页面（err.php） 29.4.4 测试代码 29.5 开发实例——订单签名 29.5.1 签名页面（Sign.php） 29.5.2 验证签名页面（verifySign.php） 29.5.3 测试代码 第6篇 其他PKI技术应用 第30章 颁发和获取数字证书 30.1 利用OpenSSL颁发数字证书 30.1.1 准备工作 30.1.2 建立根证书 30.1.3 颁发用户证书 30.2 利用Windows证书服务

颁发 数字证书 30.2.1 准备工作 30.2.2 安装证书服务并设置CA 30.2.3 提交证书请求 30.2.4 证书颁发机构处理请求 30.2.5 下载证书 30.3 通过CA机构获取数字证书 第31章 安全电子邮件应用指南 31.1 Foxmail安全电子邮件应用 31.1.1 为Foxmail邮箱账户配置证书 31.1.2 发送和阅读安全电子邮件 31.2 Outlook安全电子邮件应用 31.2.1 为Outlook邮箱账户配置证书 31.2.2 发送和阅读安全电子邮件 第32章 代码签名应用指南 32.1 什么是代码签名 32.2 Windows应用程序代码签名 32.2.1 申请代码签名证书 32.2.2 使用SignCode.exe对代码签名 32.2.3 查看代码签名证书 32.3 Java代码签名 32.3.1 下载签名工具 32.3.2 申请签名证书 32.3.3 执行代码签名 32.3.4 验证Java代码签名 32.4 移动代码签名 32.4.1 主流移动操作系统对代码签名的要求 32.4.2 代码签名的操作方法

章节摘录

第1篇　PKI技术概述　　第1章　PKI基础知识　　本章将介绍PKI的基础知识、PKI的用途、数字证书等技术，并介绍一下PKI常用术语。
由于本书专注实战，所以这里对密码理论知识不做深入介绍。
1.1　PKI概述　　PKI是Public Key Infrastructure的缩写，即公开密钥基础设施，它是国际上解决开放式互联网络信息安全需求的一套体系。
PKI体系支持身份认证，信息传输、存储的完整性，消息传输、存储的机密性，以及操作的不可否认性。
“基础设施”的作用，就是只要遵从必要的原则，不同的实体都可以方便地使用基础设施提供的服务。
使用PKI安全基础设施就像将电器接通电源一样简单。
PKI的核心是认证中心（cA）。
CA就像公安局发放身份证一样，发放一个叫“数字证书”的身份证明。
这个数字证书包含了用户身份的部分信息，以及用户持有的公钥。
像公安局对身份证盖章一样，CA利用本身的私钥为数字证书加上了数字签名。
PKI的核心技术基础是公钥密码学的“加密”和“签名”技术。
1.2　什么是数字证书　　数字证书就是网络通信中标志通信各方身份信息的一系列数据，其作用类似于现实生活中的身份证。
它是由一个权威机构发行的，人们可以在交往中用它来识别对方的身份。
最简单的证书包含一个公开密钥、名称以及证书授权中心的数字签名。
一般情况下证书中还包括密钥的有效时间、发证机关（证书授权中心）的名称、该证书的序列号等信息，证书的格式遵循ITUTX.509国际标准，如图1.1所示。
一个标准的X.509数字证书包含以下一些内容。
证书的版本号。
证书的序列号。
每个证书对于特定的CA来说都有一个唯一的证书序列号。
证书所使用的签名算法。
证书的发行机构名称，命名规则一般采用X.500格式。
证书的有效期。
现在通用的证书一般采用UTC时间格式，它的计时范围为l950～2049。
证书所有人的名称，命名规则一般采用X.500格式。
证书所有人的公开密钥。
证书发行者对证书的签名。
在Windows下可以方便地查看证书，打开证书可以看到类似图l.2所示的证书内容。
1.2.1　数字认证的原理　　数字证书采用公钥体制，即利用一对互相匹配的密钥进行加密、解密。
每个用户可以设定一把特定的仅为本人所知的私有密钥（私钥），用它进行解密和签名；同时设定一把公共密钥（公钥）并由本人公开，为一组用户所共享，用于加密和验证签名。
当发送一份保密文件时，发送方使用接收方的公钥对数据加密，而接收方则使用自己的私钥解密，这样信息就可以安全无误地到达目的地了。
通过数字的手段保证加密过程是一个不可逆过程，即只有用私有密钥才能解密。
在公开密钥密码体制中，常用的一种是RSA加密算法。
其数学原理是将一个大数分解成两个质数的乘积，加密和解密用的是两个不同的密钥。
即使已知明文、密文和加密密钥（公钥），想要推导出解密密钥（私钥），在计算上是不可能的。
按现在的计算机技术水平，要破解目前采用的1024位RSA密钥，需要上千年的计算时间。
公开密钥技术解决了密钥发布的管理问题，商家可以公开其公开密钥，而保留其私有密钥。

购物者可以用人人皆知的公开密钥对发送的信息进行加密，安全地传送给商家，然后由商家用自己的私有密钥进行解密。

如果用户需要发送加密数据，则需要使用接收方的数字证书（公钥）对数据进行加密，而接收方则使用自己的私钥进行解密，从而保证数据的安全保密性。

另外，用户可以通过数字签名实现数据的完整性和有效性，只需采用私钥对数据进行加密处理，由于私钥仅为用户个人拥有，因此能够保证签名文件的唯一性，即保证数据由签名者自己签名发送，签名者不能否认或难以否认；数据自签发到接收这段过程中未曾作过任何修改，签发的文件是真实的。

1.2.2　数字认证是如何颁发的　　数字证书是由认证中心颁发的，该证书是认证中心与用户建立信任关系的基础。

在用户使用数字证书之前必须首先下载和安装。

认证中心是一家能向用户签发数字证书以确认用户身份的管理机构。

为了防止数字凭证的伪造，认证中心的公钥必须是可靠的。

认证中心必须公布其公钥，或由更高级别的认证中心提供一个电子凭证来证明其公钥的有效性。

后一种方法导致了多级别认证中心的出现。

数字证书颁发过程如下：用户产生了自己的密钥对，并将公钥及部分个人身份信息传送给认证中心。

认证中心在核实身份后，将执行一些必要的步骤，以确信请求确实由用户发送而来。

然后，认证中心将发给用户一个数字证书。

该证书内附有用户和他的密钥等信息，还附有对认证中心公钥加以确认的数字证书。

当用户想证明其公钥的合法性时，就可以提供该数字证书。

1.3　为什么要使用数字证书　　虽然因特网电子商务系统技术可以使在网上购物的顾客能够极其方便地获得商家和企业的信息，但同时也增加了对某些敏感或有价值的数据被滥用的风险。

买方和卖方都必须保证在因特网上进行的一切金融交易运作都是真实可靠的，并且顾客、商家和企业等交易各方都具有绝对的信心，因而因特网电子商务系统必须保证具有十分可靠的安全保密技术，也就是说，必须保证网络安全的四大要素，即信息传输的保密性、数据交换的完整性、发送信息的不可否认性、交易者身份的确定性。

1.3.1　信息传输的保密性　　交易中的商务信息均有保密的要求。

例如，若信用卡的账号和用户名被人知悉，就可能被盗用；订货和付款的信息被竞争对手获悉，就可能丧失商机。

因此，在电子商务的信息传播中，数据一般均有加密的要求。

1.3.2　交易者身份的确定性　　网上交易的双方很可能素昧平生，相隔千里。

要使交易成功，首先要确认对方的身份。

商家要考虑客户端是不是骗子，而客户也会担心网上的商店是一个欺骗消费者的黑店。

因此，能方便而可靠地确认对方身份是交易的前提条件。

1.3.3　发送信息的不可否认性　　由于商情的千变万化，交易一旦达成是不能被取消的，否则必然会损害一方的利益。

例如，订购黄金时，订货时金价较低，但收到订单后金价上涨了。

这时，如果收单方否认收到订单的实际时间，甚至否认收到订单的事实，则订货方就会蒙受损失。

因此，电子交易通信过程的各个环节都必须是不可否认的。

编辑推荐

PKI，利用公钥加密技术，解决电子商务信息安全需求的成熟体系。
内容充实，技术全面，覆盖了常见的PKI应用开发技术注重应用。
强调实战，填补了PKI类书籍只重理论没有实战的空白。
以“步骤+代码”的方式进行讲解，让初学者快速入门实例典型、代码丰富，有极大的应用价值
博客专栏支持，解惑答疑，深入交流。
本书直接从实战出发，介绍了PKI应用开发过程和细节。
本书介绍了PKI应用开发常用的技术，包括OpenSSL开发、CrytoAPI开发、Java Security开发、电子商务网站应用、PKI相关技术应用等，涉及C语言、Java语言、Web开发语言(JSP、ASP/ASP.NET、PHP)，
每个系列都是按照先原理、再讲解、再实战的方式进行。
力求读者学完本书后，可进行项目实践。

版权说明