地开发出标准、通用和易于扩展的安全加密应用程序。

CryptoAPI 是一组函数，为了完成数学计算，必须具有密码服务提供者模块（Cryptographic Service Provider，CSP）。Microsoft 通过 RSA Base Provider 在操作系统级提供一个 CSP，使用 RSA 公钥加密算法，更多的 CSP 可以根据需要增加到应用中。事实上，CSP 有可能与硬件设备（如智能卡）一起来进行数据加密。CryptoAPI 接口允许通过简单的函数调用来加密数据、交换公钥、哈希一个消息来建立摘要以及生成数字签名。CryptoAPI 还为许多高级安全性服务提供了密码操作，包括用于加密客户机/服务器消息，用于在各个平台之间传递机密数据和密钥的 PFX、代码签名等。

CryptoAPI 体系共由 5 部分组成，体系结构如图 29-2 所示。

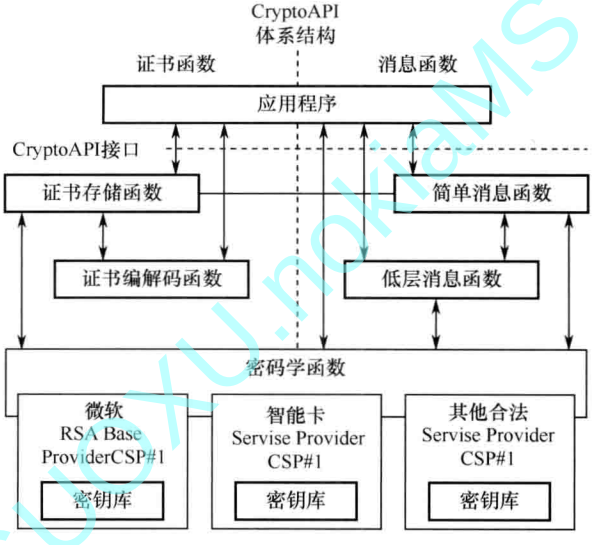

图 29-2 CryptoAPI 体系结构

（1）基本加密函数（Base Cryptographic Functions）

用于选择 CSP、建立 CSP 连接、产生密钥、交换及传输密钥等操作。

（2）证书编解码函数（Certificate Encode/Decode Functions）

用于数据加密、解密、哈希等操作。这类函数支持数据的加密/解密操作；计算哈希、签发和验证数字签名操作；实现证书、证书撤销列表、证书请求和证书扩展等编码和解码操作。

（3）证书库管理函数（Certificate Store Functions）

用于数字证书及证书库管理等操作。这组函数用于管理证书、证书撤销列表和证书信任列表的使用、存储、获取等。

（4）简单消息函数（Simplified Message Functions）

用于消息处理，比如消息编码/解码、消息加/解密、数字签名及签名验证等操作。它是把多个低层消息函数包装在一起以完成某个特定任务，方便用户使用。

（5）低层消息函数（Low-level Message Functions）

低层消息函数对传输的 PKCS#7 数据进行编码，对接收到的 PKCS#7 数据进行解码，并且对接收到的消息进行解码和验证。它可以实现简单消息函数的所有功能，且提供更大的灵活性，但需要更多的函数调用。

每类函数的命名前缀都有约定，前缀约定如下：基本加密函数 Crypt、证书编码与解码函数 Crypt、证书库管理函数 Store、简单消息函数 Message、低层消息函数 Msg。

2. CNG

Windows Vista 引入了新的加密 API 以替代旧的 CryptoAPI。旧的 CryptoAPI 存在于早期版本的 Windows NT 系列和 Windows 95 中。下一代加密技术（CNG）旨在长期替代 CryptoAPI，取代 CryptoAPI 提供的所有加密基元或服务。CNG 支持 CryptoAPI 提供的所有算法，而且应用更广泛并且包括许多新算法和更灵活的设计，从而为开发人员提供了对如何执行加密操作，以及算法如何协同工作以执行各种操作的更强的控制能力。

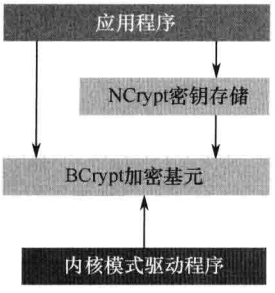

图 29-3　CNG 体系结构

图 29-3 说明了 CNG 的总体设计。BCrypt 是 CNG 的子集，提供加密基元，如随机数字生成、哈希函数、签名和加密密钥。NCrypt 也是 CNG 的子集，但它提供密钥存储工具，以支持永久保存非对称密钥和硬件（如智能卡）。从 BCrypt 和 NCrypt 的命名并不能了解太多内容。BCrypt 仅仅是头文件和为 CNG 提供基本服务的 DLL 的名称。在这种情况下，“B”代表“base”。同样，NCrypt 仅仅是头文件和提供高级别密钥存储功能的 DLL 的名称。“N”代表“new”。

29.5　Java 安全 API 规范

按照 Java 安全白皮书，Java 安全体系基本上分为 5 部分，分别是 Java 平台、Java 密码体系、Java 认证与授权、安全通信、PKI（公钥密码基础设施）。Java 安全技术包括一系列的 API、工具，以及常用密码算法、机制和协议。

1. Java 平台

Java 语言本身嵌入了安全特性，其中包括编译器和 JVM 对强数据类型的支持、自动内存管理、字节代码的验证机制以及独特的安全类加载方式，这些特性都是 Java 语言本身所具备的。

2. Java 密码体系

Java 密码体系依赖于 JCA 和 JCE，是两个非常重要的框架，它们提供了非常简洁通用的 API 接口。接口跟实现是完全分离的，即 Java 开发者可以采用 Sun 的实现方式，也可以接受其他的实现方式。

3. Java 认证与授权

Java 认证与授权用于两个目的：

① 针对用户认证，用于安全可靠地确定是谁当前正在执行 Java 代码。

② 针对用户授权，用于确保他们拥有所执行动作的访问控制权限。

JAAS（Java Authentication Authorization Service，Java 认证和授权 API）以插件方式工作，从而允许应用程序与底层身份验证技术相隔离。新的或更新的验证技术可以插入应用程序，而不需要修改应用程序本身。

4．安全通信

主要规范了标准安全通信协议（SSL、TLS、Kerberos、SASL 等）的 API 和实现。应用最多的是 SSL/TLS，其次是 Kerberos。

5．PKI（Public Key Infrastructure，公钥密码基础设施）体系

Java PKI 规范提供了管理密钥和证书的 API，实现的协议包括：X.509 规范、CRL（证书作废列表）、PKCS#11、PKCS#12、PKIX（RFC3280）、在线证书状态协议（OCSP）。

API 接口规范可分为多类，具体如表 29-1 所示。

表 29-1　Java API 接口分类

分类	Java 包
通用安全相关包	java.security javax.crypto java.security.cert java.security.spec javax.crypto.spec java.security.interfaces javax.crypto.interfaces java.security.cert
JAAS 相关包	javax.security.auth javax.security.auth.callback javax.security.auth.kerberos javax.security.auth.login javax.security.auth.spi javax.security.auth.x500 com.sun.security.auth com.sun.security.auth.callback com.sun.security.auth.login com.sun.security.auth.module
GSS-API 相关包	org.ietf.jgss com.sun.security.jgss
JSSE 相关包	javax.net javax.net.ssl
SASL 相关包	javax.security.sasl
基于 SSL/TLS 的 RMI 套接字包	javax.rmi.ssl
XML 数字签名相关包	javax.xml.crypto javax.xml.crypto.dom javax.xml.crypto.dsig javax.xml.crypto.dsig.dom javax.xml.crypto.dsig.keyinfo javax.xml.crypto.dsig.spec
智能卡 I/O 包	javax.smartcardio

29.6 CCID 规范

CCID（Integrated Circuit (s) Cards Interface Devices）是由几大国际级 IT 企业共同制定的一个标准，它提供了一种智能卡读写设备与主机或其他嵌入式主机实现相互通信的通用机制。可以从 http://www.usb.org 网站下载规范版本。

CCID 标准规定了 CCID 设备是一种芯片/智能卡接口设备，设备通过 USB 接口与主机或其他嵌入式主机连接，进行符合 CCID 标准的数据通信。同时设备通过符合 ISO/IEC 7816 标准协议的接口与智能卡进行通信，其结构如图 29-4 所示。

CCID 为所有通过 USB 连接的智能卡读写器定义了一个标准通信协议，使得相同的主机端驱动可以与任何符合 CCID 标准的智能卡读写器进行通信。

微软公司在其 Windows 2000 及以上的操作系统中提供并支持 CCID 驱动，使设备生产厂商可以轻松地开发使用符合 CCID 接口标准的设备。同时，CCID 接口标准支持 PC/SC 接口调用。在其他开源操作系统如 Linux 的众多版本上，也有许多开源的 CCID 驱动可供开发者和使用者使用。

CCID 与 PC/SC 的关系可以理解为载体与被载体的关系。PC/SC 是一个大的框架，CCID 只是针对 USB 通信的一个协议。PC/SC 是一个让卡片懂得用户需要卡片进行操作的一个协议，在通信中充当了 CCID 协议的消息内容。CCID 是让读卡器懂得 PC 发出的命令，从而规范其命令的一个协议，在通信中充当了 ISO/IEC-7816 命令的载体。

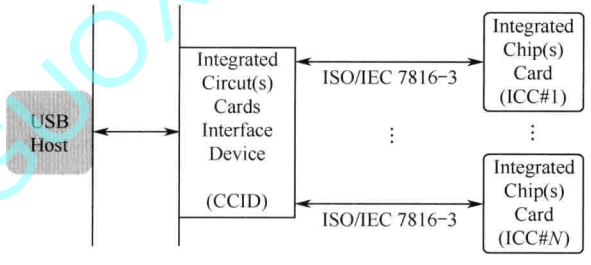

图 29-4 CCID 结构示意图

附录　主要参考资料

[1] 维基百科网站，zh.wikipedia.org，en.wikipedia.org
[2] OpenLdap 网站，www.openldap.org
[3] IETF RFC 网站，www.ietf.org/rfc
[4] CSDN 博客网站，blog.csdn.net
[5] Oracle 技术网站，www.oracle.com/technetwork
[6] Microsoft 开发者网站，msdn.microsoft.com
[7] 163 博客网站，http://www.blog.163.com/
[8] OpenSSL 网站，http://www.openssl.org
[9] Apache 网站，http://httpd.apache.org
[10] Tomcat 网站，http://tomcat.apache.org
[11] EJBCA 网站，http://www.ejbca.org
[12] ItEye 网站，www.iteye.com
[13] Codeguru 网站，www.codeguru.com
[14] 北京数字证书认证中心网站，www.bjca.org.cn
[15] 中国金融认证中心网站，www.cfca.com.cn
[16] 东方中讯数字证书认证网站，www.ezca.org
[17] 工信部电子认证业务规则规范网站，xxaqs.miit.gov.cn
[18] 国家商用密码管理办公室网站，www.oscca.gov.cn
[19] 卫生部网站，wsb.moh.gov.cn
[20] 互动百科网站，www.baike.com
[21] IC 卡标准-ISO-IEC 7816（1~15）
[22] Smart-Card Integrated Circuit(s) Card Interface Devices，www.usb.org
[23] 公钥密码标准，Public Key Cryptography Standard，PKCS#1~PKCS#15.
[24] 智能 IC 卡及智能密码钥匙密码应用接口规范.
[25] ZHANG Mingde etc. Research on Model of Trust Degrees for PKI.IAS 2009, ISBN 978-0-7695-3744-3, Aug. 2009, vol. 2: pp.647-650.
[26] ZHANG Mingde etc. Improved Approach on Modeling and Reasoning about PKI/WPKI. WiCOM 2010, ISBN: 978-1-4244-3709-2, Sep. 2010: pp.1-4.
[27] 张明德等. 改进的 PKI 可信度模型. 小型微型计算机系统，2012 年 2 月，第 33 卷第 2 期，P370-375.
[28] 张明德等. 应用安全中身份认证建模及推理方法. 小型微型计算机系统，2012 年 4 月，第 33 卷第 4 期，P754-758.
[29] 张明德等. 身份认证可信度研究. 计算机科学，2011 年 11 月，第 38 卷第 11 期，P43-47.