支持移动多媒体通信的安全中间件的研究

摘 要

随着计算机技术和通信技术的不断发展，使计算机应用得到空前的普及和因特网的快速延伸；信息的传输也从有线向无线发展，无线网络所传输的信息也已从简单的语音信息、文字信息到复杂的多媒体信息；电子商务也已向无线延伸；所有的这一切在方便了人们的生活的同时极大地推动了社会信息化的发展。然而，信息安全问题始终没有得到很好的解决；这里面所涉及的因素很多，比如提供信息安全的密码学理论本身的复杂性与难懂性，应用环境的复杂性与多样性等等；特别是在无线网络环境中，安全问题尤为突出，它所涉及的问题也更加复杂，如无线接口的开放性，网络带宽的有限性，无线终端计算能力不足，还有如吞吐量、大规模并发等效率、性能、成本等等。这些都不为一般人所能掌握，也没有这个必要。信息安全问题已经成为制约社会信息化进一步发展的主要瓶颈。

现在，软件的“开放性”原则已成为共识，应用软件的开发和集成也从开始的手工作坊式阶段，到工厂化、流程作业化、模块装配化、自动化方向发展，分工越来越细，也越来越明确。中间件技术的正是迎合这种趋势而出现的，它处于系统软件和应用软件之间，属于可复用软件的范畴，具有标准的程序接口和协议，可实现不同硬件和操作系统平台上的数据共享和应用互操作。总的作用是为处于自己上层的应用软件提供运行与开发

的环境，帮助用户灵活、高效地开发和集成复杂的应用软件。

支持移动多媒体通信的安全中间件主要是整合移动网络中的安全技术和中间件技术，来屏蔽其安全技术的复杂性，并且综合考虑移动网络环境中的局限性等等，向一般用户提供具有标准的应用程序接口（API），让他们从这些复杂的细节中解脱出来，更专注于他的本职工作。从而使安全技术真正易用，易普及。

本文主要是对支持移动多媒体通信的安全中间件进行研究，并取得了一些有意义的研究成果，主要内容包括：

1. 对支持移动多媒体通信的安全中间件的整体框架进行设计；
2. 给出安全中间件中的通用管理器（CSM）模块和资源信息服务器 RIS 模块大概设计；
3. 给出安全中间件中的安全服务层（SSL）中的 3GPP 和 WAP 安全服务的设计与实现并给出 IPSec 的实现框架；
4. 给出安全服务提供者模块(SSPM)中的密码服务提供者（CSP）和证书服务提供者模块（CertSP）设计与实现；
5. 最后提出一种混沌序列在 3G 安全通信中的应用方法，并给出了它的优点和复杂度。

关键字：移动多媒体通信　安全中间件　信息安全　API　3G

RESEARCH ON THE SECURITY MIDDLEWARE OF SUPPORTING MOBILE MULTIMEDIA COMMUNICATIONS

ABSTRACT

Along with developing of the computer and communication technologies, computer application has obtained the unprecedented popularization and the Internet also has gained the rapid extension; The main Carrier of the information transmission has been already from wired to wireless.the information transmitted by the wireless network has already been from simple voice messages, text messages to complex multimedia style. The E-commerce entered the field of wireless. All of these have facilitated people's life enormously and greatly promoted the development of information society at the same time. However, the problem of information security is unsolved until now, because this problem involves many factors. For instance, the cryptology theory which provides the information security is very difficult to understand and the application environment is also very complex and so on. Especially in the wireless network environment, security problem is outstanding, which involves the question is more complex. For example, wireless interface's openness, the network bandwidth is limited, wireless terminal computation ability is insufficient, and throughput, large-scale, efficiency, performance, cost and so on.

All of these issues are impossible to grasp and to resolve for the general user. Security problem has become the main bottleneck which restricts the further development of social information.

"Openness" in the software principle is now popular. The development and integration of application software is from the beginning of the hands workshop-style stage, to the factory, the flow of operations, module assembly, and automation. And the division is much smaller and clearer. The middleware technology is emerging to cater to such trend.It is between the system software and application software by virtue of the standard procedures and interfaces agreement, and could realize that different hardware and operating system platform data share and apply of each other, It could provide operational and development environment for their own software applications, help users flexible, efficient development and integration of complex software applications.

The main fuction of the Security middleware of supporting mobile multimedia communications is. integrates the mobile network security technology and middleware technology to shield theirs complexities, such as the algorithm complexity, the network protocol complexity, and the mobile network environment limitations.On the other hand ,it provides a standard Application Programming Interfaces (API) for the general users. And then the users can free from the complicated details and concentrate on his official duty work. After that, security technology will be really easy to use, and will be popularly in practice by the security middleware.

In this paper, the main research objective is study the security middleware of supporting mobile multimedia communications. Some significant conclusions are drawn. The main content included:

1. Design the architecture of security middleware of supporting mobile multimedia communications.
2. Give the Common Security Manager (CSM) module and the Resources Information Server (RIS) module.
3. Design and realize the 3GPP and WAP security services in the Security Service Layer, and the IPSec architecture is also to be given in the latter.
4. Design and realize the Cipher Service Provider and the Certificate Service Provider (CertSP) in the Security Service Provider Module (SSPM).
5. Propose one application method which chaos sequence in the 3G Security communications, and give its merit and complexity.

KEY WORDS: Mobile Multimedia Communications; Security Middleware; Information Security; API; 3G

第一章 绪 论

1.1 移动通信技术的发展及其安全技术

从20世纪70年代以来，移动通信进入了飞速发展时期，它经历了以模拟技术为特征的第一代模拟蜂窝移动通信系统（其典型代表为 Advanced Mobile Phone Service: AMPS）到以数字技术为特征的第二代数字蜂窝移动通信系统（其典型代表为全球移动通信系统 GSM：Global System for Mobile communication，（早期的 GSM 代表 Group Special Mobile））以及在 21 世纪初兴起的以多媒体技术为特征的第三代移动通信系统（分别有三种标准:WCDMA,CDMA2000,和 TD-SCDMA）。第三代移动通信系统正在实现着人们期待已久的无论何人（Whoever）在何时(Whenever)何地(Wherever) 都能和另一个人（Whomever）进行任何方式(Whatever)的通信的 5W 梦想。

第一代蜂窝移动通信网络基于模拟通信技术，采用频分复用（FDMA，Frequency Division Multiple Access）模式，刚开始网络容量基本上可以满足当时移动通信用户的需要。但是，由于不同的网络工作在不同的无线频段而且信道带宽也不尽相同，这使得第一代移动通信网络之间不能互联互通；随着模拟蜂窝电话的迅速发展也开始显现出其它缺点，特别是在人口密集的大城市，由于模拟式蜂窝电话采用的频分多址技术造成频率资源严重不足；再加第一代蜂窝移动通信系统没有采取密码技术来保护无线网络访问的安全性，使得移动用户的通话信息很容易被窃听，移动用户的身份也很容易被假冒，从而给网络用户和网络运营商造成巨大的经济损失[1]。

进入20世纪80年代末期，基于模拟技术的第一代蜂窝移动通信系统已经显得过时了，而此时随着大规模集成电路的进步、微型计算机、微处理器和数字信号处理技术的大量应用推动了数字通信技术在第二代蜂窝移动通信网络中的应用。例如，先进的数字语音编码技术在保证话音质量的前提下可以大大减少通信带宽的需要，从而可以提高网络频段的利用率，并且在频分复用（FDMA）的基础上又采用了时分多址（TDMA，Time Division Multiple Access）来增加网络容量；差错控制技术可以用来增强网络抗干扰能力，使得基站可以以低功率发射；而且，数字通信技术利于引入密码学技术来保护通信的安

全，比如数字加密技术可以保护数字化了的用户语音、数据和网络信令，身份认证技术可以鉴别移动用户的身份，有效地防止身份假冒。因此，基于数字技术的第二代蜂窝移动通信网络比基于模拟技术的第一代蜂窝移动通信网络不仅性能提高，容量增加，而且安全。20 世纪 90 年代，第二代蜂窝移动通信网开始在世界各地投入运营。1990 年泛欧洲的数字蜂窝移动通信网（GSM）率先在西欧各国开始运营，它使得欧洲摆脱了第一代模拟蜂窝移动通信网络体制众多互不相通的困境。后来，澳大利亚、中国以及一些中东国家陆续采用了 GSM 网络，使得 GSM 网络成为世界上覆盖范围最大的移动通信网络。虽然 AMPS 是美国的唯一的第一代蜂窝移动通信标准，但是美国第二代数字蜂窝移动通信网络及数字化的 AMPS（也称之为 DAMPS）却存在着两种互不兼容的复用技术标准：1992 年，电信工业协会（TIA：Telecommunication Industry Association）的 TR45-3 子委员会提出了基于 TDMA 技术的 IS-54 标准(1996 年更新版本为 IS-136)，它可以将 AMPS 网络容量提高 3 倍；1993 年，TIA 的 TR54-5 子委员会提出了基于 CDMA(Code Division Multiple Access，码分多址)技术的 IS-95，它不仅可以增加网络容量，而且增强了网络的灵活性，支持宽带信号处理，便于向第三代过渡。

世界上覆盖范围最大的移动通信网络系统 GSM 在密码技术的保护下，其网络的安全性比第一代移动通信系统改善了很多，然而，GSM 网络也存在着安全漏洞，其采用的基于算法保密的大多数保密算法（比如 A5/1，A5/2）已经被破译[2,3]。此外密码分析结果[4,5]还表明：GSM 网络的密码方案不能抵抗来自攻击者的主动攻击。并且在消息完整性检测方面，没有采用任何完整性检测技术来保护无线信道中传送的信令消息的完整性，这样攻击者可以篡改信令信息从而达到攻击目的。

第二代蜂窝移动通信网络不仅可以提供语音通话业务，也可以提供一些低速率的数据传输业务。然而，20 世纪 90 年代末期，随着因特网与移动通信网的融合，低速率数据传输业务已经无法满足移动用户需求，对高速数据传输业务的需求推动着移动通信网络走向第三代。随着移动通信的不断发展，基于移动通信网络的业务也越来越多，如电子邮件、短信息、移动因特网、移动电子商贸以及视频传输等等，这些业务的广泛应用需要一个全球统一的移动通信网络标准。因此，国际电信联盟 ITU 在 20 世纪 90 年代就倡导制定一个全球统一的第三代蜂窝移动通信网络标准——未来公共陆地移动电信网络（FPLMTS，Future Public Land Mobile Telecommunication Network）。2000 年，FPLMTS 改名为国际移动电信 IMT2000，它将最终由两个第三代伙伴计划 3GPP 和 3GPP2 分别制定的第三代蜂窝移动通信网络标准 WCDMA 和 CDMA2000 融合而成。

3GPP 是由欧洲、中国、日本、韩国、和美国得电信标准组织于 1998 年 10 月联合成立的第三代伙伴计划（3GPP，the Generation Partnership Project），旨在制定一种以改进的 GSM 核心网络为基础的第三代蜂窝移动通信网络标准 WCDMA。3GPP2 是由美国、中国、韩国和日本等国的电信标准组织于 1998 年成立的另一个第三代伙伴计划，旨在制定一种以 IS-95 核心网络为基础的第三代蜂窝移动通信网络标准 CDMA2000。WCDMA 和 CDMA2000 有许多相似特征，然而 WCDMA 向后兼容 GSM 网络，而 CDMA2000 向后兼容 IS-95 网络，第三代移动通信网络与他们兼容互通，甚至可以利用部分已有的核心网络设施，使得网络运营商能够避免重复投资，最终保证第二代移动通信网络平滑地向第三代过渡。第三代与前面两代的最大特点是能提供宽带的移动多媒体通信服务。

3GPP 网络采纳了 GSM 网络中已经被实践证明安全而可行的网络安全访问业务，如移动用户身份保密性、移动用户身份认证性、移动用户数据和语音以及部分信令的保密性。为了消除 GSM 网络中存在的安全漏洞，3GPP 网络增加了一些新的安全业务，如网络端认证性和信令数据的完整性检测业务等；并设计了新的安全、灵活的身份认证算法、加密算法和完整性检测算法。

鉴于 GSM 标准在密码算法设计上的教训，3GPP 标准制定者采用了 128bit 密钥的分组密码算法 KASUMI[6]和 AES[7][8]来设计身份认证算法、加密算法和数据完整性检测算法，并将算法公开以便密码研究人员分析它们的安全性。在现在为止，有关 KASUMI 和 AES 算法的研究结果[7]并没有发现它们存在任何设计问题。因此，密码学界普遍认为 3GPP 网络的安全性比 GSM 网络提高了很多。此外，3GPP 网络还设有密码提示功能，使得用户知道自己是否在通信过程中使用密码功能，提高了透明性。

1.2 无线应用协议(WAP)及其安全技术

20 世纪 90 年代中后，由于移动通信技术和因特网技术飞速发展，移动通信可以帮助人们解脱有线网络的束缚，因特网拥有非常丰富的网络资源从而可以为移动用户提供非常丰富的内容服务，人们很自然地产生了移动因特网的想法。然而，移动因特网并不是因特网和移动通信网的简单结合，因为它们之间不能相通。在因特网方面，大多数因特网技术是针对计算机网络而设计的。计算机的计算能力强大，计算资源丰富，计算机

网络的传输速度非常快（可达上百 M bit/s）且可靠性高；而且，基于高速计算机网络而设计的因特网，不仅网页包含大量的图文信息，而且网络协议也非常复杂，需要大量的运算。而在移动通信网方面，移动设备受体积和功耗的限制，其计算能力和资源都十分有限，并且显示屏比较小。移动通信网络受无线带宽限制，其数据传输速率相对比较低，通信可靠性有线，而且有些移动通信网络仍然使用电路交换技术，用户在建立了连接之后将在整个通信期间完全占用此链路，并为此付费。因此要建立移动因特网就必须设计适合移动因特网特点的网络协议和网络资源。

此外，移动通信网络标准不一和移动设备种类繁多，在很大程度上妨碍了移动因特网标准的制定。为了消除这些障碍，1997 年以诺基亚、爱立信、摩托罗拉和 Unwired Planet 为首的无线设备制造商组成了 WAP 论坛，旨在制定统一的无线服务应用框架和网络协议。从 1998 年，WAP 论坛公布 WAP 规范 1.0 版以来，鉴于第二代半（2.5G）和第三代移动通信网络数据传输速率得到显著的提高（如 GPRS 速率可达 384kbit/s，3GPP 为 2Mbit/s），WAP 论坛于 2001 年公布了 WAP 规范 2.0 版，它在 1.0 的基础上增加了对因特网协议如 HTTP、TLS 和 TCP 的支持，向因特网更靠近了一步。

WAP 协议规范中为安全服务专门设置了一个安全协议层 WTLS（Wireless Transport Layer Security）。鉴于移动通信网络带宽窄而且移动设备计算能力低的特点，WTLS 对因特网传输层的安全协议（TLS Transport Layer Security，即 SSL 的继任者）进行了优化，并增加了计算复杂性低的椭圆曲线密码算法。进行优化后的 WTLS 仍然可以实现数据保密性、数据完整性和身份认证性的密码特性。WAP 规范 2.0 版还增加了对 TLS 和 IPSec 的支持。

可以看出:无线应用协议 WAP（Wireless Application Protocol）是飞速发展的因特网技术和移动通信技术相结合的产物。它不仅为无线设备提供因特网服务制定了有效的网络协议、而且提供了开发各种无线通信网络的应用框架。

1.3 信息安全技术的发展

近几年来，随着移动通信技术的发展和因特网的广泛应用，涌现出了电子商务、电子政务以及众多网上交易活动，给网络中的信息安全提出了较高的要求，如在交易过程中对方身份的识别、相关信息数字签名处理、数据的加密等，又由于在移动通信网络中，

移动站与固定网络之间的所有通信都是通过无线信道来传输的，而无线信道是开放的，任何具有适当无线设备的人均可以通过窃听无线信道而获得其中传输的信息，甚至可以修改、插入、删除或重传无线信道中的信息，达到假冒移动用户的身份以欺骗网络端的目的。这些都促进了信息安全技术的发展。

信息安全技术主要是针对安全威胁实施安全服务。安全威胁主要有四个：信息窃听、假冒攻击、信息篡改和重传以及服务后抵赖。 分别对应的安全服务技术为机密性服务、身份认证性、数据完整性和服务不可否认性（采用数字签名技术）。

1.4 安全中间件的产生的背景和特点

随着信息化的发展，人们通过有线或无线网络传输的信息也越来越多，并且涉及到各行各业，所传输的信息有机密的也有不可否认的，这样信息安全的问题就越来越受到人们的关注。虽然提供信息安全的基本技术已经得到很好的解决，关键是如何应用，但是信息安全技术的应用涉及到各种各样的知识，比如在安全的代价、易用性、互操作性等等问题上存在着巨大的障碍。应用于无线网络上的安全技术更涉及到无线网络自身所存在的缺点（比如网络带宽窄、移动终端计算能力不足、内存和能量有限等）所导致的问题更加复杂。所有的这些都不是一般应用人员所能掌握的，也没有必要人人都掌握。

从软件技术的发展的特点来看，软件的“开放性”原则已成为共识。应用软件的开发也由初期的不同的厂商各自为政，自主开发，逐步的向标准化、可重用、模块化等人们共同追求的方向迈进。可以肯定的是软件的模块化、可移植性、互操作性是软件发展的必然趋势。而在上世纪九十年代中后期出现的中间件技术正是这一特征的代表，有专家指出，中间件是继操作系统软件和数据库系统软件之后的又一次软件业的革命，对软件的产业化也有着相当重要的积极作用。

基于以上两点，安全中间件作为融合了安全技术和中间件技术的一种保障信息安全的软件应运而生。

中间件[14]（Middleware）是基础软件的一大类，属于可复用软件的范畴。中间件是位于硬件、操作系统平台和应用程序之间的通用服务系统，具有标准的程序接口和协议，可实现不同硬件和操作系统平台上的数据共享和应用互操作等，总的作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户灵活、高效地开发和集成复杂的应用

软件。

安全中间件借助中间件的概念，并采用其许多成熟的中间件技术和安全技术来屏蔽安全的复杂性，如算法复杂性、模块间和模块内部的安全、体系结构安全、基于组件的安全及其效率等等，从而使安全技术真正易用、易普及，将电子商务真正实用化。

无线安全中间件是应用于无线通信网络之上的一种安全中间件，它的设计必须适合无线网络的特点和移动终端的局限性。支持移动多媒体通信的安全中间件是一种无线安全中间件，其主要的应用平台有：（1）移动终端、（2）移动网络中的网络端平台、（3）第三方内容服务商平台。本文以通用模型进行叙述，应用于不同的平台可以根据其特性进行删简，比如应用于移动终端，就应该更简练高效。

1.5 国内外无线安全中间件的研究状况

安全中间件的概念是最近几年才被人们的提出的。国外已有相关的产品推出，目前国内只有为数不多的厂商或科研机构从事这方面的研发。在其应用上来看，大多数主要是在有线网络上的，涉及电子商务、电子政务等方面，但在国内投入使用的不多见。对于无线安全中间件的研究报道，国外国内都比较少见。主要有：

国外：2000年欧洲提出了和SESAME[9] 和2002年美国的UNIVERSITY OF ILLINOIS（伊利诺斯州大学）在SESAME基础上发展的Tiny SESAME[10,11,12]，不过它只提供无线终端接入无线网络的安全，没有涉及到无线终端与internet互联的安全。

国内：浙江工业大学软件开发环境重点实验室的蒋融融，江颉，陈铁明在《无线安全中间件的研究与设计》[13]一文中提出了一个无线安全中间件模型。但是它只考虑了WAP安全部分，没有涉及无线终端接入无线服务网络的安全。

1.6 课题的来源、研究目的和主要工作

本课题来源于广西科学研究与技术开发计划资助项目(桂科攻 0428006-4)《通信与信息安全产品－支持移动多媒体通信的中间件平台的开发》。

本论文研究目的在于整合安全技术与中间件技术使其适合应用于移动通信网络，以中间件平台形式向上层用户提供统一的 API 接口，使安全技术更加易用，为信息化的

进一步发展扫清障碍。这是推动计算机技术与无线通信技术的进一步结合发展，实现网上信息的无地域限制的安全传输，充分发挥网上海量信息的优势（即丰富的内容），推动信息化的进一步发展的核心问题。

主要工作：研究移动通信安全（主要是3G安全）和无线终端接入internet的安全（WAP的WTLS）（特别是移动电子商务的安全）问题，以及IPSec安全原理及其应用；在参考有线网络的安全中间件的基础上致力于提出一个适合于无线通信网络的支持移动多媒体通信的安全中间件模型；并实现最主要的安全服务：如信息的加解密，数据的完整性、数据的签名和数字证书等等。

1.7 论文结构

本论文共分七章，各章的内容安排如下：

第一章 简要介绍移动通信的发展及其安全技术和WAP的安全性以及安全中间件的概念及特性，以及无线安全中间件技术的国内外研究现状和研究意义。

第二章 介绍密码学理论的相关概念和主要的加密算法及相关技术，为后续章节提供密码学理论基础。

第三章 介绍支持移动多媒体通信的安全中间件的系统构成，然后简单介绍了其中的通用管理器CSM模块和资源信息服务器RIS模块。

第四章 详细介绍了安全服务层（SSL）模块的设计和实现。

第五章 详细介绍了安全服务提供者模块(SSPM)的设计和实现。

第六章 详细介绍了本文提出的一种混沌序列在3G安全通信中的应用方法

第七章 总结和展望，总结全文并提出今后进一步的研究思路。

论文最后是参考文献、攻读硕士学位期间发表的论文以及致谢。

第二章　安全中间件的密码学理论基础

安全中间件主要是向上层用户提供信息安全服务应用程序接口（API），是一种将密码学技术和业务安全管理规则以软件实现的一个组合形式。其中密码学技术起着非常重要的作用，依据密码学技术来实现信息安全的四大目标：数据的机密性、数据的完整性、认证（实体和数据源）、不可抵赖性；对应采用的密码学技术为：加密技术、消息认证/数据完整性技术、身份识别/实体认证技术、数字签名技术。本章首先介绍密码学的基本概念，然后再分别介绍：加密技术、消息认证/数据完整性技术、身份识别/实体认证技术、数字签名技术以及算法安全性等概念。

2.1 密码学的基本概念

密码技术（cryptology）是信息安全技术的核心[15]，它主要由密码编码技术（cryptography）和密码分析技术（cryptanalysis）两个分支组成，是数学的一个分支学科。密码编码技术的主要任务是寻求产生安全性高的有效密码算法和协议，使其满足对消息进行加密或认证的要求，也称为密码编码学，从事此行的叫密码编码者(cryptographer)。密码分析技术的主要任务是破译密码或伪造认证信息，以实现窃取机密信息或进行诈骗破坏活动，也称为密码分析学，从事此行的叫密码分析者（cryptanalyst）。这两个分支既相互对立又相互依存，正是由于这种对立又统一关系，推动了密码学自身的快速发展。目前人们将密码理论与技术分成两大类，一类是基于数学的密码理论与技术，包括公钥密码、分组密码、序列密码、认证码、数字签名、Hash函数、身份识别、密钥管理、PKI 技术、VPN 技术等；另一类是非数学的密码理论与技术，包括信息隐藏、量子密码、基于生物特征的识别理论与技术等。

假设Alice与Bob要进行安全通信，Alice作为发送者(Sender)，Bob作为接受者(Receiver)。Alice想发送消息给Bob，那么Alice必须确信窃听者Malice不能阅读他发送

给Bob的消息（Message），这样，Alice必须用某种方法把消息伪装起来以隐藏它的内容，这个过程就称为加密(Encryption)。消息(Message)被称为明文(Plaintext)．被加密过的消息称为密文(Ciphertext)，而Bob想理解这个消息，必须把密文转换为明文，这个过程就称为解密(Decryption)。

密码算法(Algorithm)也叫做密码（Cipher），是用于加密和解密的数学函数（Function）。（一般情况下，有两个相关的函数：一个用于加密，另一个用于解密）。密码算法分为两大类：公开的密码算法和受限制的算法。

如果算法的保密性是基于保持算法本身的秘密，这种算法称为受限制的算法。受限制的算法具有历史意义，但按现在的标准，它们的保密性已远远不够。大的或经常变换的用户组织不能使用它们，因为每有一个用户离开这个组织，其它的用户就必须改换另外不同的算法。如果有人无意暴露了这个秘密，所有人都必须改变他们的算法。更糟的是，受限制的密码算法不可能进行质量控制或标准化。每个用户组织必须有他们自己的唯一算法。这样的组织不可能采用流行的硬件或软件产品。但窃听者却可以买到这些流行产品并学习算法，于是用户不得不自己编写算法并予以实现，如果这个组织中没有好的密码学专家，那么他们就无法知道他们是否拥有安全的算法。尽管有这些主要缺陷，受限制的算法对低密级的应用来说还是很流行的，用户或者没有认识到或者不在乎他们系统中内在的问题（在GSM系统中开始采用的就是受限制的算法）。

如果一个密码算法的保密性是基于密钥（Key）的安全，而不是基于算法的细节的安全性，算法的保密性与算法本身是否保密无关，这样的密码算法称为公开的密码算法。这就意味着算法可以公开，也可以被分析，可以大量生产使用算法的产品，即使偷听者知道你的算法也没有关系；如果他不知道你使用的具体密钥，他就不可能阅读你的消息。

现代密码系统的安全性都是基于密钥的安全性，而不是基于算法的安全性，一个典型的密码系统（Cryptosystem）由算法、以及所有可能的明文、密文和密钥组成。

明文用 M 或 P 表示，称为消息空间的集合。它可以是各种各样的信息或符号，比如可以是位序列、文本文件、位图、数字化的语言序列或数字化的视频图像等等。在计算机系统中，M 一般都是简单的二进制数据。明文可被传送或存储，无论在哪种情况，M 指待加密的消息。

密文用 C 表示，它也是二进制数据，通常的情况下 C 和M一样大，但是也可以采用压缩算法和加密算法相结合，这样经过加密和压缩的到的 C 就会比 M 小（压缩比例取决于压缩算法）；仅通过加密得到的 C 有时会稍大。

加密解密变换[16]

- Key 表示一个称为密钥空间的集合，其中的元素 k 称为密钥。
- 每个 $k_e \in Key$ 唯一地确定从 M 到 C 的一个双射，记为 E_k，称为加密函数或加密变换。如果是可逆变换，即每个不同的密文都可以恢复成唯一的明文，那么 E_k 必须是双射的。
- 对于 $k_d \in Key$，D_k 表示从C到M的一个双射（即 D_k：$C \to M$）。D_k 称为解密函数或解密变换。
- 一种加密方案包括一个加密变换集 $\{E_k,\ k_e \in Key\}$ 和一个相应的解密变换集 $\{D_{k_d} : k_d \in Key\}$，且有如下性质：对每个 $k_e \in Key$，存在唯一的 $k_d \in Key$，使得 $D_{k_d} = E_{k_e}^{-1}$，即 $D_{k_d}(E_{k_e}(m)) = m$ 对于所有的 $m \in M$ 成立。
- 要构造一种加密方案，需要选取明文空间 M、密文空间 C、密钥空间 Key、加密变换集 $\{E_{k_e} : k_e \in Key\}$，以及一个相应的解密变换集 $\{D_{k_d} : k_d \in Key\}$。

采用数学公式表示加密解密过程如下：

加密函数 E_{k_e}（Encrypt）作用于明文 m 得到密文 c：

$$E_{k_e}(m) = c \qquad \{m \in M, c \in C, k_e \in Key\} \tag{2-1}$$

相反，解密函数 D_{k_d} (Decrypt) 作用于 c 产生 m，数学公式表示如下：

$$D_{k_d}(c) = m \qquad \{m \in M, c \in C, k_e \in Key\} \tag{2-2}$$

那么，加密解密全过程解析如下：先对明文消息 m（Message）进行加密（Encrypt）得到密文 c（Ciphertext），然后再对密文进行解密（Decrypt）恢复明文，按数学公式表示如下：将（2-1）式代入（2-2）得到公式（2-3），显而易见（2-3）成立。

$$D_{k_d}(E_{k_e}(m)) = m \qquad \{m \in M, c \in C, k_e \in Key\} \tag{2-3}$$

2.2 主要的加密算法及相关技术

现代密码学中，主要的加密算法可以分为三个类型：对称密钥算法、非对称密钥算法、无密钥算法。其中对称密钥算法和非对称密钥算法结合得到混合密码体制算法。

2.2.1 对称密钥算法（Symmetric-key Algorithms）

对称密钥算法，有时又称为传统密码算法，消息加密的历史就是应用对称加密算法的历史。这里所说的“对称”是指加密和解密都采用同一个密钥，所以这些算法也叫秘密密钥算法或单密钥算法，它要求发送者和接收者在安全通信之前，商定一个密钥。也就是说，在通信之前，通信双方必须通过一个安全的通道来交换密钥。对称算法的安全性依赖于密钥，泄漏密钥就意味着任何人都能对消息进行加/解密。只要通信需要保密，密钥就必须保密。

对称密钥算法的加密和解密表示如图2-1所示：

$$E_k(m)=c; \qquad D_k(c)=m$$

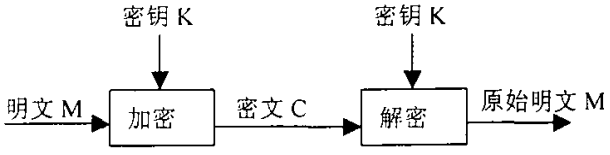


图 2-1 使用对称密钥算法的加密和解密

Figure 2-1Using the symmetric- key algorithm to encrypt and decrypt

对称密钥加密方案可以细分为两种类型[17]：通常分为分组密码（Block Cipher）和流密码（Stream Cipher）。在分组密码中，需要加密的明文被划分为固定长度的分组后再按分组来加密。在现代计算机密码系统中分组密码算法的典型分组长度为64位——但是随着计算硬件技术的进步，这长度已经不足以防止分析破译，已经逐步被128位分组所替代。现在最常用的分组密码算法有：AES,DES，3DES,IDEA, RC6和Safer++。

流密码也称序列密码，是一种非常重要的加密算法，它利用不断变化的加密变换对明文消息进行组逐字（通常为二进制数）的加密，从某种意义上说，他们是分组长度为1的简单分组密码算法。另外，使它很有用的事实是：在硬件实施上，流密码的速度一般要比分组密码快，而且不需要很复杂的硬件电路，加密变换可以随着被加密明文的每

一个符号而改变。在某些情况下（例如某些电信上的应用），当缓冲不足或必须对收到的字符进行逐一处理时，流密码就显得更加必要和恰当。由于流密码仅有有限的差错传播或没有差错传播，因此在一些容易发生传输误差的的环境中使用流密码就更有优势。

流密码发展到现在不仅仅积累了丰富的理论经验，还出现了众多的被广泛应用的密码算法标准。例如：GSM 通信系统中的 A5/1、A5/2 和 A5/3 流密码[18]；IEEE802.11 中 WEP[19]标准中 RC4[20]流密码；UMTS/3GPP 第三代移动通信中的 $f8$[20]流密码、蓝牙系统中的 E0 流密码以及 18033-4:2005 标准中的 SNOW2.0 流密码。

但是，进入 21 世纪以来，与分组密码相比，特别是随着分组密码 DES 到 AES 的更新替换，流密码的发展明显落后于分组密码。而且由于 A5/1 和 RC4 等流密码被发现存在严重的安全漏洞[21,22,23]，使得人们对流密码失去了信心，甚至有人认为 AES 可以全面代替流密码。但流密码固有的特点使得它拥有运行速度快、消耗资源少等优势使得人们逐渐加强了对流密码的关注。随着 2005 年 ISO 正式公布了一个标准——ISO/IEC FCD 18033-4: 2005，流密码的发展进入了复苏阶段。ISO 把基于线性反馈移位寄存器（LFSR）的纯流密码 SNOW2 编入该标准，表明了流密码同样可以达到 AES 的安全强度（2^{128}），而且具有更高的运行速度，占用更少的存储空间。

分组密码和流密码加密图解如下图2-2所示。

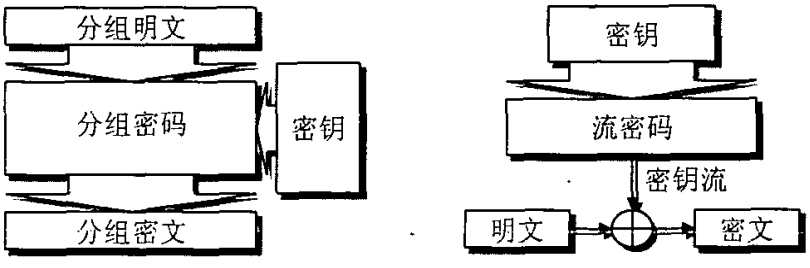


图 2-2　分组密码和流密码的示意图

Figure 2-2 Block cipher and stream cipher flow chart

对称算法的优点主要是保密强度高，计算开销小，处理速度快。其缺点主要是密钥管理困难。

2.2.2 非对称密钥算法（Asymmetric-key Algorithms）

非对称密钥算法也称为公钥密码算法（Public-key Algorithms）。对称密钥算法的一

个有两个缺点：（1）是密钥管理和分配困难，尚且缺乏安全有效的方法，即是任何密文传输之前，发送者和接收者必须使用一个安全信道预先传输密钥，在实际中，达到这一点是很困难；（2）传统的加密算法（对称算法）无法实现抗抵赖的需求。然而公钥密码算法就能很好的解决这两个问题。

1976 年 Diffie 和 Hellman 在美国国家计算机大会上提出了他们革命性的思想——公开密钥密码思想，并发表了“New directions in cryptography”[24]的著名论文，奠定了公钥密码的里程碑。1978 年 River、Shamir 和 Adleman[25]提出第一个实用的非对称密钥算法，也就是著名的 RSA 加密算法。

公开密钥算法的设计思想是基于单向陷门函数[26]（Troopdoor One-Way Function）：即公钥密码系统中的用户使用一对密钥，加密密钥 K_e 和解密密钥 K_d，用作加密的密钥 K_e 不同于用作解密的密钥 K_d，而且解密密钥 K_d 不能根据加密密钥 K_e 计算出来（至少在合理假定的时间内），其中加密密钥 K_e 公开，任何想与他通信的人都可以方便地查找到和使用，而解密密钥 K_d 保密。在公钥系统中，加密密钥叫做公开密钥（Public-key，简称公钥），解密密钥叫做私人密钥（Private-key，简称私钥）。这样每个用户都有两个密钥，即公钥和私钥。

公开密钥算法的安全性也是基于密钥的安全性，而不是基于算法的细节的安全性。如果他不知道你使用的具体密钥，他就不可能阅读你的消息。在公钥密码系统中，解密和加密是可分离的，通信双方无需事先交换密钥就可建立起保密通信。现在比较常用的公钥密码算法有：RSA密码系统、ElGamal公钥密码系统、椭圆曲线公钥密码系统等。

相对于对称密钥密码系统，它有两个缺点：（1）在吞吐率方面，大多流行的公钥加密方法要比已知最好的对称密钥方案慢几个数量级；（2）密钥长度明显比对称密钥加密所需的大得多。但它有以下优点：密钥分发简单；秘密保存的密钥量减少；在每个互不信任的双方之间，可以采用验证技术验证对方的身份；可以实现数字签名等。因此，公钥密码体制得到了越来越广泛的应用。

2.2.3 混合密码体制

在现在的密码系统中，公钥密码算法不会替代对称密钥密码算法，反之亦然，而应

该充分利用对称密钥算法与公钥算法具有互补性的优点这一特性来设计密码系统。混合密码系统（Hybrid Cryptosystem）随之而生，混合密码研究的核心就是如何取长补短，充分利用二者优势互补建立起更加有效的密码系统。

公钥密码的长处在于密钥管理方便，但是其加密效率低，反之，对称密码算法其密钥管理困难，但是其加密效率比公钥高好几个数量级。还有公钥密码学推动了有效的签名（特别是不可抵赖性），对称密钥密码学对一些数据的完整性应用比公钥算法更有效。所以在大多数实现的混合密码系统中，公钥算法用来保护和分发会话密钥（session key）（即密钥管理）以及数字签名；对称密钥算法采用会话密钥来加密会话数据和数据的完整性。密钥协商过程如下（假设Alice要和Bob进行安全通信）：

（1） 消息始发送者Alice获得消息接受者Bob的公开密钥 K_b；

（2） 消息始发送者Alice产生随机会话密钥 K_s，并用获得的Bob的公开密钥 K_b 加密会话密钥 K_s，并把加密后的会话密钥密文 $m = E_{K_b}(K_s)$ 传送给接收者Bob。

（3） 接收者Bob接收到密文 $m = E_{K_{be}}(K_s)$ 后，用自己的私钥 K_{bd} 解密密文 m，恢复会话密钥 K_s：　$D_{K_{bd}}(E_{K_{be}}(K_s)) = K_s$

（4） 通信双方采用会话密钥 K_s，通过对称密钥算法对他们的通信消息（数据）进行加/解密，以进行保密通信。

这样，把公开密钥密码用于密钥分配解决了很重要的密钥分配和管理问题，而对于对称密钥算法要而言，直到产生大量的通信数据需要对称算法加密时，才起作用。从而，当通信双方需要对通信数据进行加密时，才产生会话密钥，这极大的减小了会话密钥遭到损坏或破译的风险。当然，私人密钥在应对泄露问题上是脆弱的，但风险较小，因为只有需要对通信的会话密钥进行加密时采用到它。

2.2.4 杂凑函数（Hash 函数）和数据完整性

通信双方的通信数据仅仅通过加密来达到保密通信是远远不够的，在此，我们对开放通信网络的脆弱性做一个理想的、标准的假设：所有的通信都经受一个称为Malice的攻击，他可以随意地窃听、截取、重发、修改、伪造和插入消息等。当Malice插入了修

改过的或伪造的消息，他试图欺骗目标接收者，使其相信该消息来自某个其他合法的主体。这样我们就需要一种机制，使其得到消息的接收者可以验证该消息确实是来自声称的消息源，且在传输过程中未受到未授权的修改。完整性就是抗击对消息未授权修改的安全服务。

数据完整性保护定义[26]：设Data为任意消息，K_e为编码密钥，K_v为与该编码密钥相匹配的验证密钥。那么Data的数据完整性保护包括如下的密码变换：

篡改检测码的生成：$MDC \leftarrow f(K_e, Data)$。

篡改检测码的验证：

$$g(K_v, Data, MDC) = \begin{cases} True, \text{概率为 } 1 & \text{如果} MDC = f(K_e, Data) \\ False, \text{压倒性概率} & \text{如果} MDC \neq f(K_e, Data) \end{cases}$$

其中f和g都是有效的密码变换，MDC是*Manipulation Detection Code*（篡改检测码）的缩写。

类似于密码系统的情况，数据完整性保护技术也可以分为对称技术和非对称技术。但是由非对称技术实现的密码系统中，公钥和私钥有固定的用法：公钥用于消息编码（加密），私钥用于消息解码（解密）。而在由非对称技术实现的数据完整性系统中，公钥（或私钥）既可以用于编码，又可以用于验证。

在实现数据完整性的对称技术中，密码变换f和g都是对称算法，即$f = g$并且$K_e = K_v$。由于数据完整性和消息认证之间的密切关系，由对称密码技术生成的MDC常称为消息认证码（MAC）。MAC的生成和验证可以使用密钥杂凑函数技术，也可以使用分组密码加密算法。

实现MAC最常用的方法是使用所谓的带密钥的杂凑函数技术。从最高层次上说，杂凑函数可以分为两类：（1）不带密钥的杂凑函数，它规定只有一个输入参数（一个消息）；（2）带密钥的杂凑函数，它规定要有两个不同的输入，即一个消息和一个秘密密钥。杂凑函数是一个确定的函数h，它将任意的有限长比特串输入x映射为固定长度为n比特的杂凑值$h(x)$输出；并且在给定h和输入x，容易计算出$h(x)$。实用的h应具有以下几点性质：混合变换、抗碰撞攻击、抗原像攻击、实用有效。现在比较常用的带密钥的杂凑函数有：MDC-2（with DES）、MDC-4（with DES）等等；不带密钥杂凑函数有：MD5、SHA-1、SHA-256、SHA-384、SHA-512。

杂凑函授还有以下几个重要的用途：

- 在数字签名中，杂凑函数一般用来产生“消息摘要”或“消息指纹”。
- 在具有实用安全性的公钥密码系统中，杂凑函数被广泛地用来实现密文正确性验证机制。
- 知识证明：方便了用户数据的承诺或显示了数据的拥有，却不暴露数据本身。
- 密钥推导：从以前的密钥计算出新的密钥序列。
- 在需要随机数的密码学应用中，杂凑函数被广泛地用做实用的伪随机函数。

2.2.5 数子签名技术

数字签名技术（Digital Signature Technology）是建立在公开密钥加密体制基础上的。1976年Diffie和Hellman提出了数字签名的概念[27]，1978年RSA签名方案的提出，说明了签名的可行性和实用性，20多年来，数字签名技术在各行各业得到广泛的应用。消息的数字签名其实是一个数值，它依赖于只有签名者知道的某个秘密数以及代签名的消息的内容。

类似于手写签名，一个数字签名方案至少应该满足一下三个条件：

- 接受方能确认或证实发送方的签名，但任何其他人都不能仿造或伪造签名，简记为$R-$条件。
- 发方发出自己签过的消息后，就不能再否认他签过的消息，简记为$S-$条件。
- 当双方关于签名的真伪发生争执时，无偏见的第三方应当能够公正的解决问题，而且无须访问签名者的秘密信息（即私钥），简称为$T-$条件。

数字签名与手写签名的区别在于：手写签名是模拟的，且因人而异；数字签名是0和1的数字串，因消息而异。数字签名与消息认证的区别在于：消息认证使收方能验证消息发送者及所发消息内容是否被篡改过，但是当接受和发送者之间有利害冲突时，单纯用消息认证技术就无法解决他们之间的纠纷，此时必须借助满足前述要求的数字签名技术。

为了实现签名的目的，发送方必须向接受方提供足够的非保密消息，以便使接收方能验证其消息的签字；但又不能泄漏其用于签字的机密信息，以防止他人伪造签名。因此，签名者和证实者可公用的信息不能太多。任何一种产生签名的算法或函数都应当提供这两种信息，而且从公开的信息很难推测出用于产生签字的机密信息。再有，任何一

种数字签名系统的实现都有赖于仔细设计的通信协议。

数字签名方案一般可以分为两种[28]：

（1） 带有附录的数字签名方案：这些方案要把原始消息作为输入，以验证数字签名。通常情况下，这些方案以哈希（单向）函数为基础。也即是把消息转化为哈希结果（输出），然后再对哈希结果（输出）进行签名，把签名信息附加在消息之后或某一特定位置上。这些方案相当强大，不易受攻击，因而成为最常用的方案。常用的方案有：ANSI X.931、PKCS#1和Bellare-Rogaway PSS。

（2） 带有消息恢复的数字签名方案：在这些方案中，签署的原始消息能够从数字签名自身中恢复。因此，无需知道原始消息，因为数字签名承载了原始消息，但是这样造成了开销的急速增大，所以这些方案也只能适于较小的消息。常用的方案有ISO/IEC 9796-1、9796-2和Bellare-Rogaway PSS-R。

若按明、密文的对应关系划分，每一种中又可分为两个子类：一类是确定性(Deterministic)数字签名，其明文和密文一一对应，它对一特定消息的签字不变化，如RSA, Rabin等签字；另一类是随机化的(Randomized)或概率式数字签名，它对同一消息的签名是随机变化的，取决于签名算法中的随机参数的取值。一个明文可能有多个合法的数字签名，如ElGamal等。其中确定性签名机制还可以再分为一次签名方案和多次签名方案。

一个签名体制一般含有两个组成部分，即签名算法(Signature Algorithm)和验证算法(Verification Algorithm)。基于公钥密码算法的数字签名算法本质上是公钥加码算法的颠倒应用。也即是：签字算法或签名密钥是秘密的，只有签字人掌握；证实算法应当公开，以便于他人进行验证。

一个签名体制可由量(M, Ms, S, Key, V)组成。其中，M是明文空间，Ms是的代签空间，S签名空间，Key是密钥空间，V是证实函数的值域（真、伪）。这样，基于公钥密码算法的带有附录的数字签名方案步骤如下：

1. 发送方用一个Hash函数对消息m进行处理，形成一个固定长度的消息摘要m_s1：

   也即：$m_s1 = Hash(m)$　　其中$m_s1 \in Ms$，$m \in M$；

2. 发送方将自己的私人密钥$Ke(Ke \in Key)$和消息摘要m_s进行签名运算，产生数字签名s，即$s = Sig_{Ke}(m_s) \in S$；

3. 将数字签名和消息一起发送出去；

4. 接受方用同样的 *Hash* 函数对消息进行计算，产生消息摘要 m_s2。

5. 接受方用发送方的公开密钥 $Kd(Kd \in Key)$ 和接受到的数字签名 s 进行运算，产生消息摘要 m_s1。若 $m_s1=m_s2$，则数字签名得到验证，否则，数字签名验证失败。即：$Ver_{Kd}(m_s2,s)=v \in V$。

体制的安全性在于：从 M 和其签字 S 很难推出 Ke 或者很难伪造一个 M，使 M 和 S 可被证实为真。

消息签名与消息加密有所不同，消息加密和解密可能是一次性的，它要求在解密之前是安全的；而一个签字的消息可能作为一个法律上的文件，如合同等，很可能在对消息签署多年以后才验证其签字，且可能需要多次验证此签名。因此，签字的安全性和防伪造的要求更高些，且要求证实速度比签字速度还要快些，特别是联机在线实时验证。

数字签名在信息安全，包括身份认证、数据完整性、不可否认性以及匿名性等方面有重要应用，特别是在大型网络安全通信中的密钥分配、认证以及电子商务系统中具有重要作用。数字签名是认证的重要工具。

随着计算机网络的发展，过去依赖于手书的签字的各种业务都可用这种电子签名代替，它是实现电子贸易、电子支票、电子货币、电子购物、电子出版以及知识产权等系统安全的重要保证。

数字签名的算法很多，其中DSA[29]签名算法已经成为美国联邦信息处理标准（FIPS 186），称为数字签名标准（DSS）。其他的应用最为广泛的三种是:Hash签名、ElGamal签名、RSA签名。但是，当前的情况是：ANSI X.931已经得到了广泛的标准化，PKCS#1已经得到了非常广泛的部署，并且PSS被认为是最安全的；当今存在的基于PKI的基础设施均以目前的PKCS#1规范为基础。

2.2.6 身份识别和实体认证技术

在通信的时，有时出于安全的需要，必须设计允许一方（验证者）得到另一方（声称者）所声称的实体（或双方相互认证）的认证技术，以防止假冒。

认证这一概念可以分为三个子概念：数据源认证、实体认证和认证的密钥的建立。第一个概念主要涉及验证消息的某个声称属性；第二个概念则更多地涉及验证消息发送

者所称的身份；第三个概念则涉及进一步致力于产生一条安全信道，应用后继的应用层安全通信会话。

数据源认证（也称消息认证）与数据完整性密切相关，但是它们具有很大的差别，表现如下：

- 首先：数据源认证涉及通信。它是一种安全服务，消息接收者用它来验证消息是否来自源于所声称的消息源。数据完整性则不一定包含通信过程，该安全服务可以用于存储的数据。
- 其次：数据源认证必然涉及消息源的识别，而数据完整性则不一定涉及该过程。
- 再次：也是最重点的一点，数据源认证必然涉及确认的消息的新鲜性，而数据完整性却无此必要，一组老的数据可能有完善的数据完整性

根据迄今为止的讨论，可以将数据源认证这一概念的特征总结如下：

（1）包含从某个声称的源（发送者）到接收者的消息传送过程，该接收者在接收时会验证消息。

（2）接收方执行消息验证的目的在于确认消息发送者的身份。

（3）接收方执行消息验证的目的还在于确认在原消息离开消息发送者之后的数据完整性。

（4）验证的进一步的目的在于确认消息传输的“活现性”（也即是新鲜性）。

实体认证是这样的一个过程，即其中一方（通过获得证实的证据）确信参与协议的第二方的身份，并确信第二方真正参与了该过程（即在证据获得之时或之前是活动的）。通常这一过程也称为身份识别协议。

身份识别协议的目的：从验证者的角度，实体认证协议的结果或者是接受声称者的身份是可信赖的（可接受的），或者是终止接受（拒绝）。

身份识别协议的性质：

- 身份识别的交互性：一方或双方可对其他方确证他们的身份，分别出单向认证或相互认证提供。
- 计算效率：执行协议所需要的操作数。
- 通信效率：包括传输（消息交换）的步骤和需要的带宽（总传输的比特数）。
- 第三方（若有）的实时参与。
- 第三方（若有）需要的可信性。
- 安全保证性和秘密储存。

身份识别和数字签名方案之间的关系：

身份识别方案与数字签名方案是紧密联系在一起的，但比数字签名方案简单。数字签名方案包含有可变的消息摘要，并且通常提供不可抵赖性，以允许事后法官来解决争端。对于身份识别方案，消息的语义基本上是固定的——在当前时刻下及时声称身份。声称可以立即确证或拒绝，从而实时保证或解除相关的特权或访问。还有身份识别不像签名一样有生存期——解决争端时通常不必考虑以前的身份识别，且将来可行的攻击不会影响以前的身份的有效性。在某些情况下，使用标准技术可以将身份识别方案转换为数字签名方案。

认证密钥的建立：通常，通信双方运行实体认证协议的目的在于能够在高层或者应用层上进行安全通信。在现在密码学中，密钥是安全通信信道的基础。因此，为了进行高层或应用层安全通信而运行的实体认证协议通常都有一个子任务，即(认证的)密钥建立，或者密钥交换、密钥协商。

如实体认证的情况，可根据数据源认证获得有关宣称这的身份，在认证的密钥建立协议中，密钥建立素材也是重要的协议信息，因而它应该也是数据源认证的内容。

在文献中，（实体）认证协议、认证的密钥建立（密钥交换、密钥协商）协议、安全协议或有时甚至密码协议，常常都是指同一通信协议集。

常用的认证协议分为弱认证协议和强认证协议两类：

1. 弱认证协议有：口令认证包括：固定口令方案、PIN 和通行密钥、一次口令（向强认证过渡）。
2. 强认证协议有：挑战-响应身份识别包括对称密钥技术的挑战-响应和使用公钥技术的挑战-响应。
3. 此外还有自定义的和零知识的身份识别协议。

2.3 密码算法的安全性

密码算法的安全性：密码学的鼻祖可以说是信息论的创始人香农，他提出了一些概念和基本理论，并且论证了只有一种密码算法在理论上不可解的，即： One Time Padding，这种算法要求采用一个随机的二进制序列作为密钥，与待加密的二进制序列按

位异或，其中密钥的长度不小于待加密的二进制序列的长度，而且一个密钥只能使用一次（但因为此法代价过高，所以这种算法目前还很难广泛实用）。除此以外的其它算法都是理论上可解的。从历史来看，如DES算法，NIST的第一代数据加密标准(DES) 作为一个美国政府标准。其密钥实际长度是56比特，作2^{56}次穷举，就肯定能找到加密使用的密钥。但是直到被庞大的电脑并行网络攻击和特定的“DES2Cracking”硬件攻克时[30]（它采用的破译方法是强破译攻击法这种方法是针对特定的加密算法设计出相应的硬件来对密钥空间进行穷举搜索），它已连续使用了20 年。AES 比DES支持更长的密钥。假如能建造一个DES2Cracking机,能以每秒2^{55}个密钥进行DES 密钥搜索,则需149万亿年才能搜索完128bitAES 密钥。除非一些对AES的攻击速度比密钥穷尽搜索(Key exhaustion) 要快得多,否则AES 应能保持超过20 年之久的安全。

所以采用的密码算法做到事实上不可解就可以了，当一个密码算法已知的破解算法的时间复杂度是指数级时，称该算法为事实上不可解的

2.4 本章小结

本章简单的介绍了密码学的基本概念及相关术语。并且详细的介绍了对称算法和非对称算法的加解密原理，以及混合对称算法和非对称算法密码体制的优点，还对hash函数和数据完整性、数字签名技术和实体认证技术做了一一的介绍。最后给出了密码算法安全性的相关概念。为后续章节提供理论基础。

第三章　支持移动多媒体通信的安全中间件的系统构成

支持移动多媒体通信的安全中间件主要采用简化的PKI框架，同时兼顾无线移动通信的特点，利用中间件优点，完成对PKI基本功能{如加密解密、消息摘要、单向散列、数字签名、签名验证、证书认证和密钥生成、储存、销毁等实现与封装扩充}，形成一致的系统安全服务接口和通信安全服务接口。该中间件屏蔽了安全技术的复杂性，向用户提供统一的安全接口标准（API），满足各种级别的安全需求应用，能与其他无线应用中间件一起无缝地整合于无线应用平台。支持移动多媒体通信的安全中间件是一种无线安全中间件，其主要的应用平台有：（1）移动终端、（2）移动网络中的网络端平台、（3）第三方内容服务商平台。本文以通用模型进行叙述，应用于不同的平台可以根据其特性进行删简，比如应用于移动终端，就应该更简练高效。

3.1　安全中间件的总体结构

安全中间件不是一个简单的概念，而是实现安全策略、实现安全服务的基础框架。本文设计的支持移动多媒体通信的安全中间件包括六大部分、五层结构和四级接口[31,32,33]。

3.1.1 安全中间件的六大部分

这六大部分包括应用程序层、组件服务层、安全服务层、通用安全管理器、安全服务提供者模块和资源信息服务器组成，如图 3-1 所示。

- 安全服务提供者模块(SSPM: Security Service Provider Module)：是整个支持移动多媒体通信的安全中间件中的最基本部分。一般由基础安全提供商（如加密机厂商、智能卡厂商等）以基础安全服务接口的方式提供。根据现有的安全提供服务，我们将各种

安全服务分为加密服务、证书服务、可信策略服务、数据储存服务和其他可扩展的服务，相应的实现模块就是加密服务提供者模块、证书服务提供者模块、可信模型提供者模块、数据储存提供者模块和其他扩展安全服务提供者模块。由于此层为整个安全中间件提供最为基本的底层的安全服务，为此本项目编码实现了最基本的加密服务提供者模块、证书服务提供者和数据储存提供者模块。由于本安全中间件将可信模型提供者和其他资源提供者定义为可扩充模块，所以尚未实现。

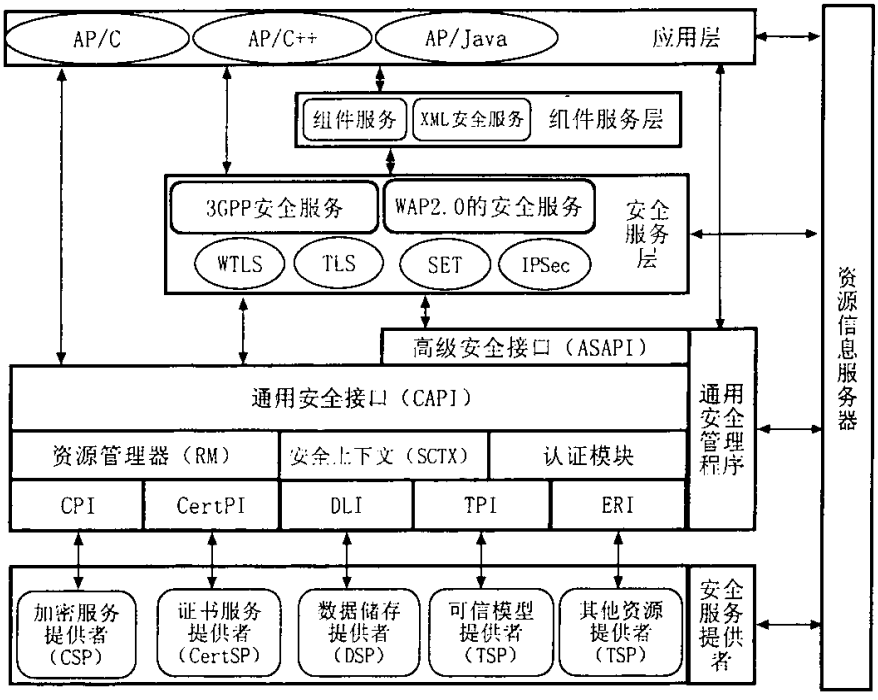


图 3-1 安全中间件的系统结构

Figure 3-1 Security middleware system structure

- 通用安全管理程序(CSM: Common Security Manager)：一般由安全中间件平台提供商提供，是安全中间件的核心。它通过管理安全服务提供者模块(SSPM)来屏蔽基础安全服务算法复杂性和实现的细节，提供平台无关性和开发/运行高效性。它提供的高级接口非常简单、易用、通用接口可以根据实际应用环境选择算法，使用应用与具体的实现无关。各高层模块利用它提供的接口，使用安全服务提供者提供的各种安全服务。另外，通用安全管理器也为各种基础安全提供者模块的灵活加载提供了动态装载机制。特别是应用于移动终端时，由于移动终端的局限性，模块的灵活加载尤其重要，否则必然会占据很大一部分移动终端的内存和计算能力，从而导致移动终端性能的严重下降。

- 安全服务层（SSL：Security Service Layer）：主要提供 3GPP 标准建议的安全服务和 WAP2.0 的 WTLS/TLS 层安全服务。基于移动网络的趋势是全 IP 网络和电子商务的无线化，所以还应该提供 SET（安全电子商务）和 IPSec 服务。该层延伸了现有的常规的安全中间件概念，从而可以使该层又可以成为一个中间件的形式存在。

- 组件服务层：一般应由组件提供商提供，并以组件级接口方式出现，如 XML 工具等。这种组件级开发，是软件开发发展中继 OOT 技术后的新里程碑。

- 应用层：应该由各具体应用用户（厂商）开发或提供。各种安全应用程序如果建立在组件级、安全服务级之上，它就可以不用关心加密模块的细节、网络协议的细节过程等；如果建立在基础安全接口或通用级接口之上，该用户可以是安全专家。这样就解决信息安全技术应用难的问题。

- 资源信息服务器：为五层安全模块的灵活组态提供实质性的支持，因为所有的安全服务、通用安全管理器和基础安全提供者这都必须进行注册，提供其各种能力表。特别是对各种动态加载和卸载进行管理。

在这六大部分中，通用安全管理器（CSM）处于核心位置，安全中间件中屏蔽安全实现的不同、提供统一接口、调度安全任务等基本功能都是在这个模块中实现的。因此，可以说 CSM 本身还可以作为一个原始的中间件。安全服务提供者（SSPM）处于最基础的位置，提供了最基本的密码学安全服务，而 CSM 则负责管理 SSPM；而安全服务层和组件服务层是直接或间接的建立在 CSM 的服务之上的，提供最为直接的服务。应用程序使用安全中间件时必然要使用 CSM 的服务；而资源信息服务器是为其它模块提供服务的支撑模块。可以说，在整个安全中间件所提供的各级接口中，CSM 所提供的通用安全编程接口也是最为核心的接口。

3.1.2 安全中间件的四级接口

从以上分析和图 3-1 可知，本安全中间件的四级接口可由图 3-2 所表示，分别为：组件级接口、体系级接口、通用级接口和基础级接口。

- 组件级：提供最为方便的安全接口工具，甚者透明的方案，真正屏蔽安全概念的复杂性，这是安全中间件得最终目标。例如，通过一个 Wizard（开发向导）、一个编辑器，就可以完成移动电子商务的开发，安全已隐含其中。

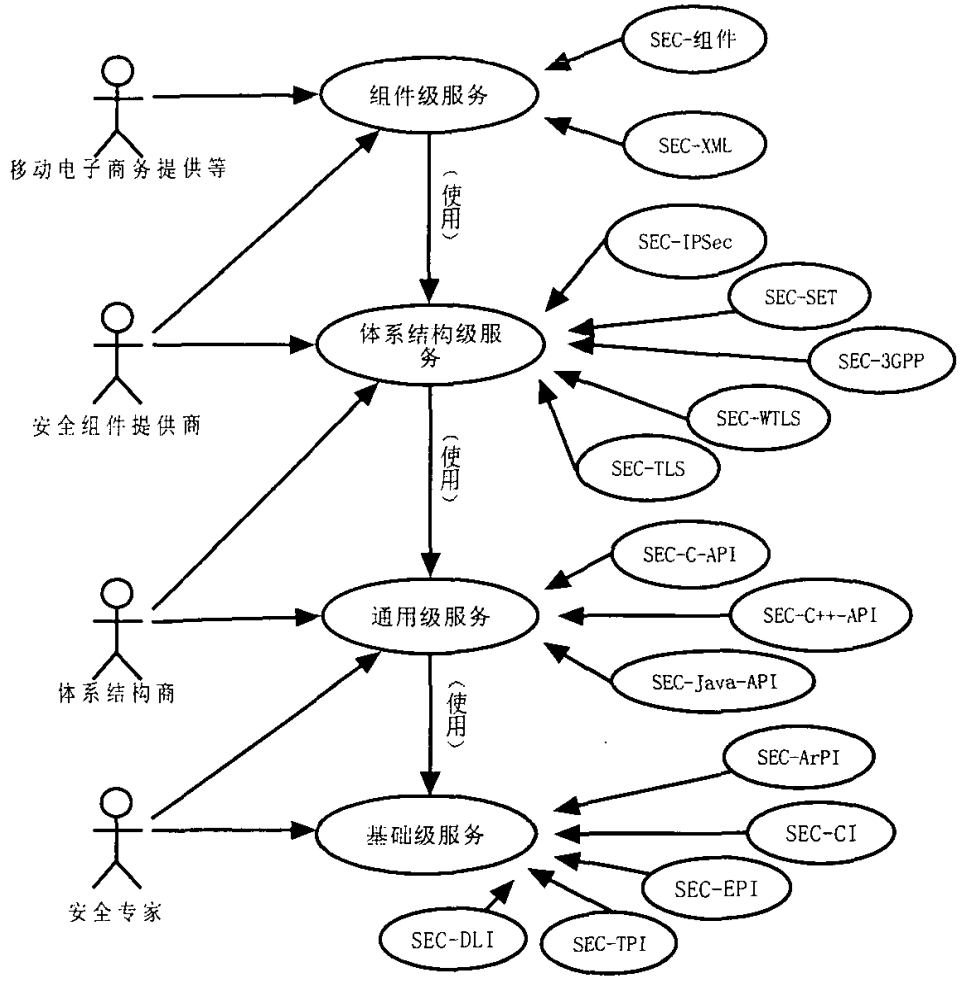


图 3-2 安全中间件中的四级接口

Figure 3-2 Security middleware four levels of interface

- 体系结构级：提供协议级、互操作级和通用中间件的安全服务接口，如WTLS/TLS、SET 和 IPSec 安全服务等（主要是 3GPP 和 WAP2.0 安全服务）。正是在体系结构级的基础上，我们才能提供面向对象、面向组件的、透明的安全组件服务，也是最具体最直接的安全服务。

- 通用级接口：这是安全中间件的核心接口。它承上启下，在多种平台上屏蔽了安全实现的细节，也是目前许多厂商所提供的“安全中间件”。根据分析，这一层涉及到大量的安全知识，因此即使多平台、多厂商的 API 统一了，开发者仍然要了解大量的安全背景才能选择使用的合适的算法。为此，安全中间件还应提供一个简单的高级接口，至少绝大部分常用的算法都是缺省的，使一般不了解密码学的用户也能使用。

- 基础级：这是为特定的安全专家提供的接口，理应由安全产品提供者直接提供，

当然是符合安全中间件一个统一规范的接口。为了方便和模块化，根据安全产品的功能，安全中间件可以将其细化为多个接口：加密服务提供者接口、证书接口、策略接口、存储控制接口、扩展接口。

3.2 通用管理器 CSM 模块

3.2.1 CSM 体系结构

CSM (Common Security Manger)即通用安全管理器[33]，它的任务包括:管理执行具体加密任务的安全服务提供者模块，负责这些模块的加载和卸载，确保模块的可信性；向上层提供统一的接口，屏蔽各种安全服务实现之间的差异，使应用与具体实现无关；接收任务请求，并根据请求的不同把请求分派到不同的模块，保证任务能并发高效地执行。应用于移动终端时，由于移动终端的性能限制（内存、CPU、和电池等不足），所以CSM必须尽可能高效地管理安全服务提供者模块，特别是加载和卸载功能。CSM的体系结构如图3-3所示。

狭义的通用安全管理器是指处于安全服务提供者模块和通用安全接口之间的一层软件结构，它通过请求/响应队列的同步机制、状态映射索引表和与队列有关的一些数据结构对请求/响应队列进行管理；利用模块配置文件、模块信息表对安全服务提供者模块进行管理；调度；在加载CSP(加码服务提供者)、CertsP(证书服务提供者)等安全服务提供者模块(SSPM)时对他们作签名检查，在调用SSPM功能时对SSPM作相互的身份认证，保证SSPM的安全可信；并在调用者和SSPM之间进行通信，根据用户进程的请求调用各种服务提供者模块的相应服务函数，并对用户进程作出响应。

广义的CSM还应该包括通用安全编程接口。它是安全中间件的核心接口，直接与CSM交互。

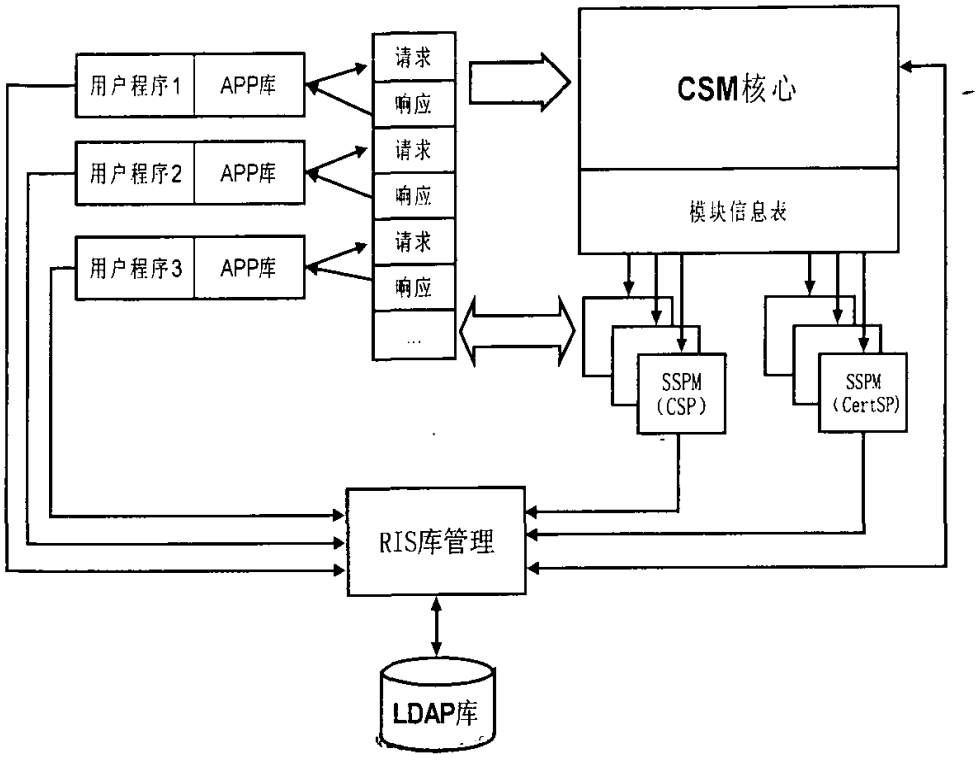


图3-3 CSM体系结构

Figure 3-3 CSM System Structure

3.2.2 CSM 运行流程

CSM在加载时，首先初始化管理SSPM模块，通过模块配置表文件获取需要在初始化时加载的安全服务提供者模块(SSPM)，从RIS中获取这些SSPM的所在路径等信息，加载这些模块并把他们填入模块信息表，并执行这些SSPM的初始化操作。然后初始化线程池。建立线程容器，按照配置文件的要求初始化一定数量的子线程，并填入线程容器，形成空闲线程列表。最后初始化请求/响应队列，建立队列单元列表，初始化队列状态索引表，启动队列，进入等待请求到达的状态。

用户的应用程序在开发时使用通用安全编程接口的函数库。用户进程在开始请求安全服务前，首先向RIS查询能够提供所需能力的安全服务提供者模块的信息。然后通过函数库的功能向CSM发出调用某一功能的请求。函数库为用户在请求/响应队列中找到一个空闲的单元用于存放用户请求的功能类型、请求服务的数据和参数以及运行结果。

请求的命令顺序包括:把模块加载到内存中的load命令，把用户进程和模块绑定的attach命令。具体的安全任务命令以及load和touch的逆操作unload和detach。

通过如信号量等同步机制，CSM 核心得知请求队列中有用户请求到达，再利用队列状态索引表找到用户请求使用的单元，获取用户请求的类型。CSM核心根据请求的类型查询模块信息表，如果没有查到该SSPM,就从RIS库中查询，然后把该SSPM调入到内存，并在模块信息表中添加相应表项。CSM核心线程从线程池中获取一个空闲的子线程，并把这个子线程和指定的SSPM绑定在一起。子线程调用安全服务提供者模块中的运行任务函数实现用户要求的功能。如有必要，CSM可与模块进行相互的可信性检验。

各种服务提供者模块的运行任务函数从请求队列中读取用户调用的数据、参数，完成用户要求的服务，把返回数据存放在请求/响应队列中的相应数据区，然后返回结果给CSM。 CSM通过同步机制通知用户进程取回结果数据。用户进程取回数据后，一次安全服务过程结束。

CSM也有把SSPM从内存中卸载的可能，比如用户程序发出unload命令。这时CSM首先检查模块信息表中该SSPM中的引用记数，判断是否还有在使用它的用户进程。如果该SSPM没有被正在处理，就先调用该SSPM的结束命令，让SSPM完成自己在被卸载前所需要完成的工作。然后CSM把SSPM从内存中卸载。

3.3 资源信息服务器 RIS 模块

3.3.1 RIS 的功能

安全中间件的体系结构中的各个实体，都被作为独立对等的模块看待，互相之间处于松散祸合的关系。应用程序和模块之间的关系在运行时才能确定下来。因此各模块如何了解其它模块的信息、功能和使用方法是安全中间件必须处理的一个问题。在本文所猫述的安全中间件中使用RIS来解决这个问题。

资源信息服务器RIS是一个为安全中间件提供对可执行对象的描述和定位以及这些对象的签名证书信息的接口。RIS在安全中间件或者开发者看来是一个数据库和数据库上提供的方法、能力集合，通过RIS的方法可以获取软件模块的相关信息。所有的系统的进程都可以对RIS进行访问。

RIS有四个不同的使用者，他们是应用程序，通用安全管理器CSM, 安全服务提供者模块SSPM和安全服务层。他们在安全中间件中分别属于不同的层次。CSM是RIS的最重要的使用者，他可以通过RIS了解各个组件的能力和属性，定位组件，得到组件的证书信息。每个SSPM在能被应用之前都必须在RIS中进行注册相应的信息。SSPM的功能不同注册的信息也不同，RIS详细规定了SSPM安装时需要注册的信息。

3.3.2 RIS 的使用

RIS是一个单独的系统级范围的服务。安全中间件中的任何模块在安装时都要确信RIS在系统中可用。RIS在安装时就要为各种模块创建信息模板，建立一系列的表结构。以后使用RIS的模块通过RIS提供的接口添加、更改和获得表的信息。其体系结构如图3-4所示。

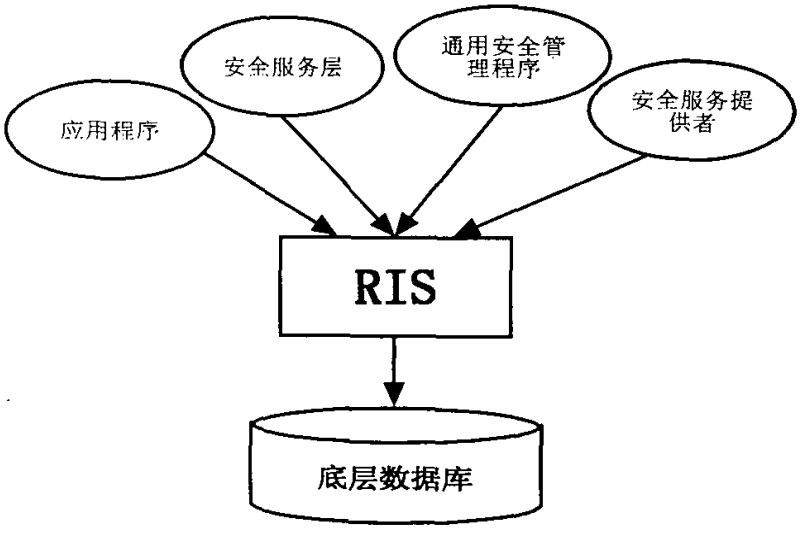


图3-4 RIS体系结构

Figure 3-4 RIS System Structure

RIS的获得可以通过程序库(动态或者静态)的加载。RIS定义了一组函数的接口。用户可以通过这些函数调用RIS的功能，在使用RIS前可以检查RIS的完整性，以确信RIS没有危险。安全中间件的配置中应指定RIS的位置。

RIS的功能被封装在一个名叫CRIS的类中，它的操作主要有:添加表项(叶节点)，删除表项，修改，查找表项。

A. 初始化和结束

在使用RIS之前必须对其进行初始化。与之对应的有RIS的结束。调用的成员函数是:CRIS::RIS_initialize; 和CKIS::RIS-Terminate。

B. 打开和关闭数据库

在完成初始化后接着就该进行打开数据库以进行具体的操作。对应的有关闭数据库。调用函数是CRIS::Dbopen和CRIS:: Dbclose.

C. 插入数据

在插入数据之前必须创建一个CRecord类的实例。CRecord类完成了对一个表项的封装。它提供了很多方法可以方便地构造一个表项。现在先介绍一下CRecord类的使用。

CRecord类方法CRecord::CreateRecord创建一个指定的表类型。RIS详细规定了表的类型和每种表的属性。

方法CRecord::SetAttributeInfo设置一个表的属性信息包括属性名字，属性的数据类型，属性值的最大个数。该函数的返回值是一个该属性的旬柄，以后对该属性的操作都以它作为参数。

CRecord::SetAttributeData方法有多态的几个实现，可以方便地加入数据。下面是一个简单实现创建一个表MOdobj的程序片段。

完成CRccord的构造后就可以使用CRIS类的方法把构造好的表项插入表结构中去。CRIS::Insert将完成这个操作，该方法有一个DB_RECORD_IDENTIFIER结构的引用参数，完成后将由它唯一表示插入的表项。

D. 删除数据

删除一个表项将要使用到在前面所获取的DB_RECORD_IDENTIFIER，以它作为参数指定要删除的表项调用CRIS::DataDelete，就可以把表结构中的该条记录删除掉。

E. 数据修改

修改数据前必须先构造一个CRecord的实例，和必须得到唯一表示一个数据的DB_RECORD_IDENTIFIER。使用它们作为参数调用CRecord::DataModify的方法，可以把DBes_RECORD_IDENTIFIER指定的记录修改为CRecord所指定的内容。

F. 查找数据

数据的查是最复杂的操作，将用到以下几个方法:CRIS::Search; CRIS::AbortSearch; CRIS::DataGefrst; CRIS::DataGetNext和一个数据结构QueryContex。该结构在一次查找中保存查找相关的上下文信息，比如查找的条件、当前查找到的位置等等，主要用于跟

踪查找操作。

调用CRIS::Search开始查找，把查找信息放入QueryContex中以供以后使用。CRIS::DataGetFirst和CRIS::DataGetNext是数据库操作中常见的命令，用于获取查找结果集中的第一条记录和遍历整个查找结果集。它们都使用QueryContex结构作为一个参数，他们返回一个指向CRecord类的指针，表示取得的数据表项。通过该指针可以用CRecord的方法得到更多的数据信息。CRIS:: AbortSearch将结束查找并释放查找过程中用到的内存。

3.4 本章小结

本章主要是叙述支持移动多媒体通信的安全中间件的体系结构。本文所叙述的安全中间件主要由六大部分、五层结构和四级接口组成，分别完成不同的功能。

本项目的重点和难点工作在于实现安全服务提供者模块和安全服务层。本项目现已按移动通信的特点要求完成了安全服务提供者模块（SSPM）中的CSP和CertSP部分，以及安全服务层的3GPP和WAP中必要的安全服务。

第四章 安全服务层（SSL）的设计与实现

支持移动多媒体通信的安全中间件的安全服务层（SSL:Security Service Layer）主要是提供3GPP标准建议的安全服务和WAP2.0的WTLS/TLS层的安全服务。但是，随着移动终端的发展和无线网络的进步，基于移动网络的趋势是全IP核心网络和无线电子商务和有线电子商务的融合，所以还应该留有提供SET（安全电子商务）和IPSec服务能力。该层延伸了现有的常规的安全中间件概念，从而可以使该层又可以成为一个中间件形式存在。 这样，不仅简化了体系结构级安全的开发和维护，还提供了基于对象的安全服务，对整个安全系统的设计、开发、集成和维护将产生革命性的影响，从而真正保证电子商务的顺利实现。下面详细述说3GPP标准建议的安全服务和WAP2.0的WTLS/TLS 层安全服务的设计与实现，以及简单介绍一下IPSec原理以及它的实现框图。

4.1 3GPP 标准建议的安全服务

4.1.1 3GPP 网络的基本结构

为了能更好的述说3GPP标准建议的安全服务，有必要先介绍一下3GPP网络的基本结构。为了能够与GSM网络兼容共存，3GPP网络中包括了一些GSM网络单元；其结构[34]如4-1图所示。

图 4-1 给出了 3GPP 网络的抽象结构图，它包括了 3GPP 网络中的主要功能实体。从功能上分，3GPP 网络结构主要包括 3 大部分：

一、用户设备 UE：它包括移动设备（ME: Mobile Equipment）和用户服务身份模块（USIM：User Service Identity Module)。它们分别执行如下功能:移动设备 ME 为移动用户访问网络建立通信连接，并执行与安全服务相关的数据加/解密和完整性

检测功能；用户服务身份模块（USIM）存有移动用户的国际身份号（IMSI）和认证密钥 K 等移动用户个人信息，并负责在网络连接建立阶段实时鉴别网络端身份，生成移动用户身份认证响应、生成保密密钥和完整性密钥。

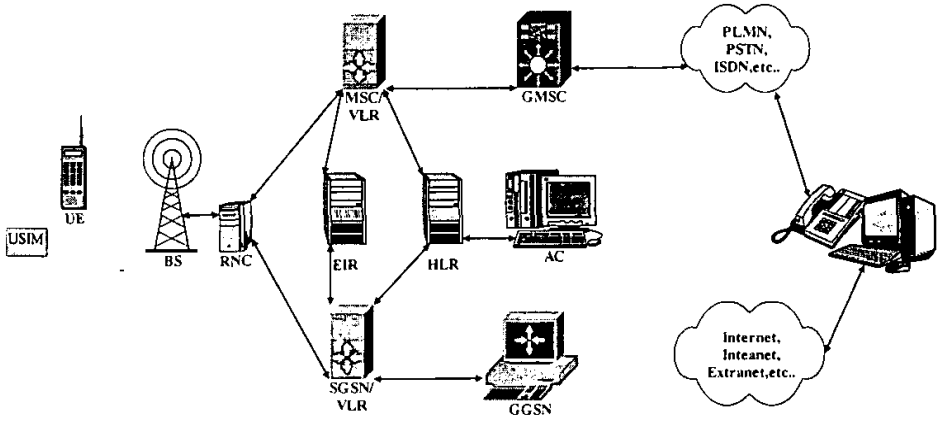


图 4-1 3GPP 网络基本结构

Figure 4-1 3GPP Network Basic architecture

二、无线访问网络（UTRAN）：UTRAN 包括基地站（BS：Base Station）和无线网络控制器（RNC：Radio Network Controller）。它们分别执行如下功能：（1）基地站 BS 在接口 IU 和 UU 之间转换数据流，也参与无线资源管理。（2）无线网络控制器 RNC 控制器管理区域内的无线资源。

三、核心网络 CN：3GPP 核心网络 CN 包括了 GSM 核心网络的所有功能单元，移动服务交换中心（MSC）、设备号登记处（EIR）、拜访局（VLR）、归属局（HLR）和认证中心（AC），此外还包括移动服务交换中心网关（GMSC）、通用分组无线服务支持节点（SGSN）和通用分组无线服务支持节点网关（GGSN），其中：移动服务交换中心网关（GMSC）是一个交换中心，与 MSC 一起完成电路交换服务中的通信连接。SGSN 在功能上类似于 MSC/VLR，但它则是用于分组交换服务。同样 GGSN 在功能上类似于 GMSC，也是用于分组交换服务。

4.1.2 3G 网络的安全结构

为了实现 3G 安全目标[35]，3GPP 给出了 3G 系统的安全结构。3G 系统的安全结构是安全特征和安全机制的组合。安全特征是满足一个或几个安全要求的业务能力；安全

机制就是用于实现安全特征的元素。3G 系统的安全特征可以归纳为网络的安全接入、网络安全、用户安全、应用安全、安全的可视性和可配置性。3GPP 定义的 3G 安全功能结构图如图 4-2 所示[36]。

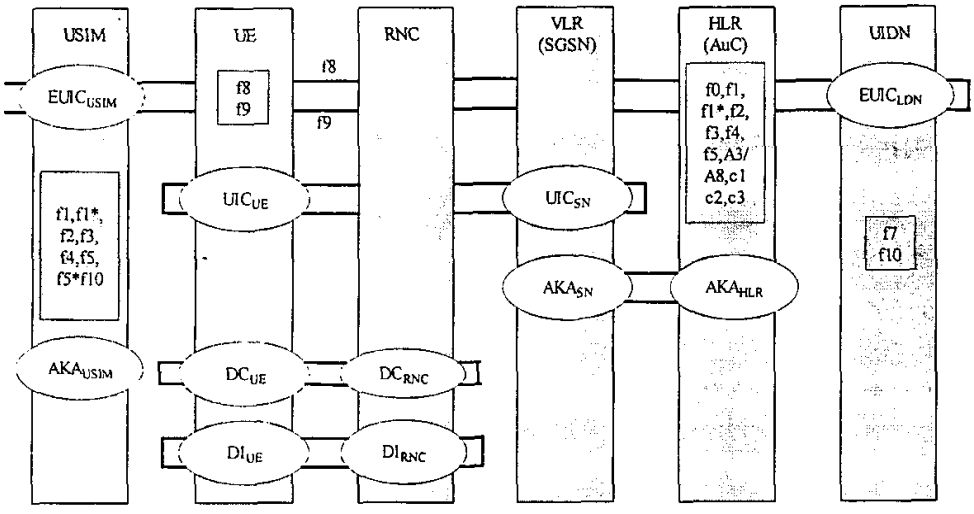


图 4-2　3G 安全功能结构

Figure 4-2 3G security function architecture

图中灰色竖条表示 3GPP 安全结构中包括的网络单元。而椭圆代表相应的安全措施，主要分为 5 类。

（1）　增强用户身份保密（Enhanced User Identity Confidentiality,EUIC）,通过归属网络内的用户标识解密结点（User Identity Decryption Confidentiality，UIDN）对移动用户智能卡身份信息进行认证。

（2）　用户身份保密（User Identity Confidentiality，UIC）。

（3）　认证和密钥协商（Authentication and Key Agreement，AKA），用于 USIM 卡、访问位置寄存器 VLR、归属位置寄存器 HLR 间的双向认证及密钥分发。

（4）　用户及信令数据保密（Data Cryptogram,DC），用于用户设备 UE 与无线网络控制器 RNC 间信息的加密。

（5）　消息认证（Data Identity,DI）,用于对交互消息的完整性、时效以及消息的来源与目的地进行认证。

3GPP 提出的安全结构中，定义了 10 种密码算法（见表 4-1）。其中，$f0 \sim f5$ 的作用有三个：

（1）　提供 USIM 与认证中心（Authentication Center，AUC）之间的相互认证。

（2） 导出会话密钥，用于加密用户数据和信令数据。

（3） 掩盖序号以保护用户身份的机密性。

表 4-1 定义了 10 种密码算法

算法代码	算法功能	算法代码	算法功能
$f0$	随机数生成函数	$f4$	完整性密钥生成函数
$f1$	正常情况下的网络认证函数	$f5$	正常情况下的匿名密钥生成函数
$f1*$	重同步消息认证函数	$f5*$	重同步情况下的匿名密钥生成函数
$f2$	用户认证函数	$f8$	数据加密算法
$f3$	加密密钥生成函数	$f9$	数据完整性算法

其中 $f0 \sim f5$，$f1*$，$f5*$ 是非标准化的[37,38]，可以由各国自主设计。$f8$，$f9$ 是通过分组密码算法 KASUMI 构造得到的标准化算法[20,39]，也是 3G 系统实现国际和国内漫游所必须的，用于用户设备 UE 和无线网络控制器 RNC 中，$f8$ 算法采用了 KASUMI 算法的输出反馈 OFB 模式，而 $f9$ 算法利用 KASUMI 算法的密码分组连接 CBC 模式，他们的输入和输出都是 64 比特，密钥是 128 比特。

4.1.3 3G 网络的认证和密钥协商与数据保密

3GPP 标准中的身份认证与密钥生成过程如下所述[40,41]：

（1） 当移动用户第一次访问网络时，传其国际移动身份号 IMSI 给 VLR。

（2） VLR 对该用户进行登记，并利用其 IMSI 从 HLR 调该用户的认证矢量 AV。

（3） HLR 利用用户的 IMSI 从 AC 数据库调用该用户的 AV（AV 的生成原理框图如图 4-4 所示）并传给 VLR。

（4） VLR 从 AV 中取出 RAND 和 AUTN,传给用户。

（5） USIM 先认证网络端的身份，如认证成功，USIM 生成移动用户身份认证响应 RES,并传给 VLR，同时生成保密密钥和完整性密钥。

（6） VLR 从 AUTN 中取出 AC 生成的 XRES，比较 RES 是否与 XRES 相同；如果相同，VLR 为移动用户分配一个临时的身份号 TMSI，并利用保密性算法

$f8$ 加密后传给移动用户；移动用户利用同样的算法和密钥解密出 TMSI，将去存入 USIM 卡，以便下一次认证之用。

上述协议第五步中 USIM 卡的其体操做过程：收到 RAND 和 ATUN 后，USIM 先利用认证密钥 K 和 RAND 通过匿名性密钥生成函数 $f5$ 生成匿名性密钥 AK，通过异或运算恢复序列值 SQN；然后，利用网络身份认证函数 $f1$ 计算出认证响应 XMAC，并与比较 AUTN 中的 MAC 进行比较。如果 XMAC=MAC,USIM 再查 SQN 是否在有效值范围之内；如果 SQN 有效，USIM 利用函数 $f2$,$f3$,和 $f4$ 计算出 RES,CK 和 IK,并将 RES 传给 VLR 进行移动用户身份认证，将 CK 和 IK 传给 UE 准备保密通信。上述描述图如图 4-3 所示

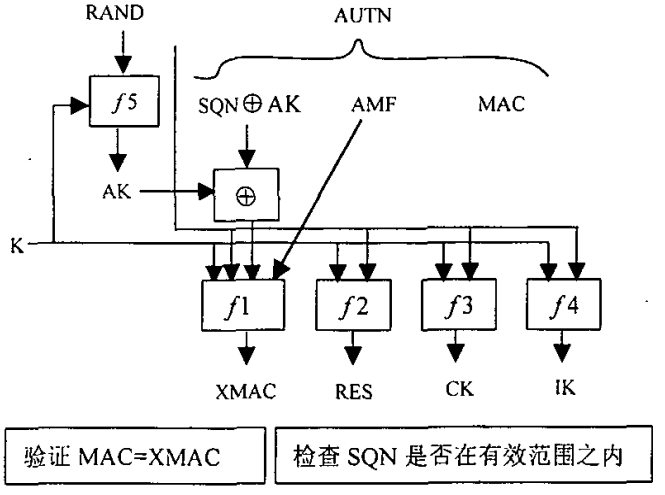


图 4-3　USIM 卡上的认证与密钥生成原理框图

Figure 4-3 User authentication function in the USIM

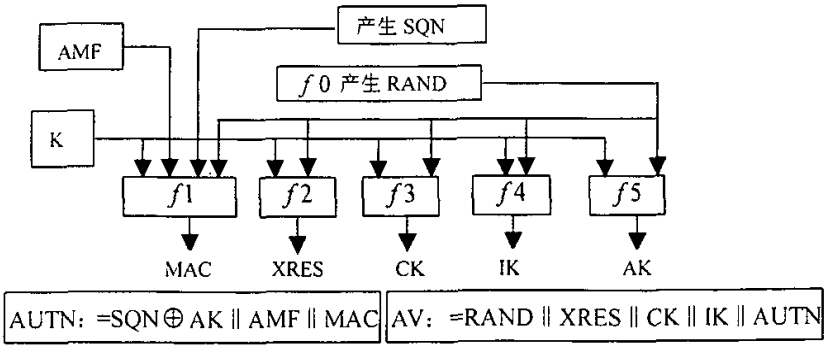


图 4-4　AC 中的认证参数生成原理框图

Figure 4-4 Generation of authentication vectors in the AC

说明：K：认证密钥，AMF：认证管理域，SQN：序列值，RAND：128 位长的良好的随机数，CK：128 位保密性密钥，IK：128 位完整性密钥。

- 重同步过程

在 USIM 认证网络端身份时，有可能出现如下情况：网络端的认证响应 MAC 是正确的，即 MAC=XMAC,但 SQN 却不在有效范围之内。这种情况说明 AC 中的累加器与 USIM 不同步，移动用户的 USIM 卡可以通过此方法来检查是否以前有人曾假冒其身份，同时归属局也可以通过此方法来防止拜访局滥用移动用户的认证参数。3GPP 标准规定的重同步是用 USIM 的累加器状态值来重新设置 AC 中的累加器。重同步认证参数生成过程如下图 4-5 所示：

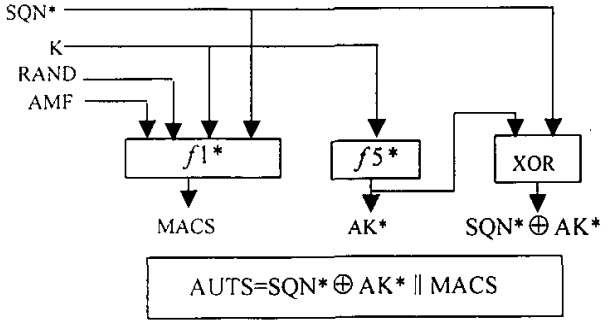


图 4-5 USIM 和 AC 重同步认证参数生成原理框图

Figure 4-5 Construction of the parameter AUTS in USIM and AC

而且，3GPP 标准规定了重同步协议，其协议流程如下：

（1）USIM 卡如上图 4-5 那样生成重同步认证响应 AUTS,传给 VLR.

（2）VLR 将 RAND 和 ATUS 传给 HLR/AC.

（3）AC 利用重同步匿名性密钥生成函数 $f5^*$ 和认证密钥 K 以及 MACS 计算出 AK,从而可以恢复出 SQN*.再利用重同步认证函数 $f1^*$ 和认证密钥 K,AMF,RAND,以及 SQN*计算出 MACS*,如果 MACS*与 AUTS 中的 MACS 相同，AC 用 SQN*导出的累加器状态值设置自己的累加器的状态，并生成一个新的 AV 传给 VLR。

（4）VLR 利用新的认证矢量 AV 可以与该移动用户重新进行身份认证。

- **UE 与 RNC 之间的数据加密模块**

GSM 网络在移动用户与基地收发站之间提供加密保护，而有些 GSM 网络中的基地收发器与基站控制器之间才用微波连接，这种微波连接很容易被窃听，因此，3GPP 网

络将加密保护延长至无线网络控制器。当 USIM 和 VLR 成功完成身份认证后，它们分别将保密性密钥 CK 传给 UE 和 RNC, 从而在 UE 和 RNC 之间建立一个保密性连接，保护它们之间传递的用户语音和数据信息以及信令信息。3GPP 标准加密方案框架如下图 4-6 所示：

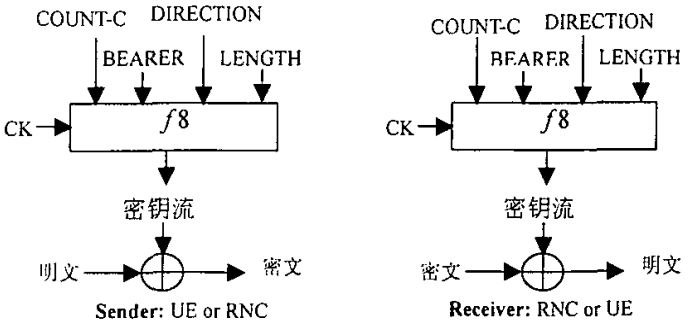


图 4-6　UE 与 RNC 之间的保密性算法的工作原理图

Figure 4-6 confidentiality algorithm work schematic diagram in the UE or RNC

$f8$ 算法的输入包括 128bit 保密性密钥 CK, 32bit 的加密序列号 COUNT-C,5bit 的载路身份 BEARER,1bit 的方向 DIRECTION(表示上行或下行链路)和 32bit 的密钥流长度 LENGTH,输出 LENGTH 指定长度的密钥流，该实现方案基于 KASUMI 分组密码算法。

- **UE 与 RNC 之间的完整性检测模块**

在 USIM 和 VLR 成功完成身份认证之后，它们分别将完整性密钥 IK 传给 UE 和 RNC，从而在 UE 与 RNC 之间提供了一个完整性认证连接。保护再它们之间传递的部分信令信息的完整性。完整性检测方案方框图如下图 4-7 所示。

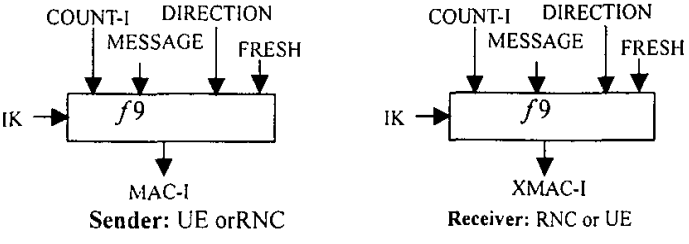


图 4-7 UE 与 RNC 之间的完整性检测算法的工作原理图

Figure 4-7 integrality algorithm work schematic diagram in the UE or RNC

UE 或 RNC 利用完整性密钥 IK，通过完整性认证算法 F9 对需要保护的信令信息 MESSAGE，生成一个称为消息认证码 MAC-I 的标签，并将它附在 MESSAGE 的后面一起传给 RNC 或 UE，收到的 MESSAGE 和 MAC-I 之后，UE 或 RNC 对 MESSAGE 利用

同样的算法和完整性密钥再次生成一个 XMAC-I，并比较 MAC-I 和 XMAC-I 是否相同。

4.1.4 3G 安全服务层的设计

基于上述安全原理，本文设计了 3G安全服务层，该层又进一步延伸了现有的常规的安全中间件的概念，可以作为一个更小的安全中间件存在。其层次结构如下图4-8所示。

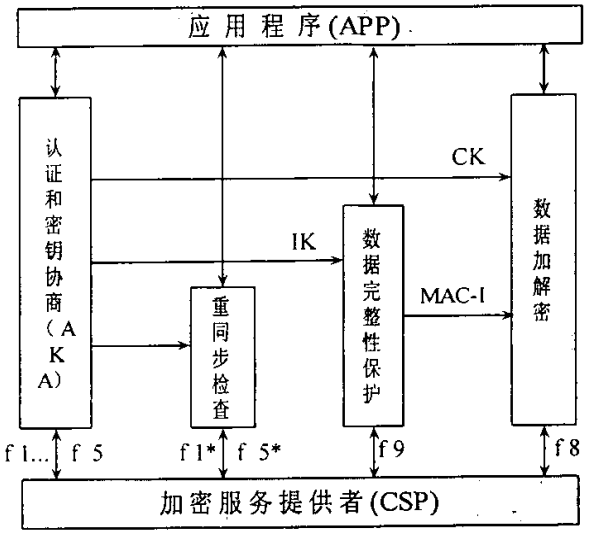


图 4-8 3G 安全服务层结构

Figure 4-8 3G SSL architecture

其中，应用程序（APP）包括上层应用程序和底层数据传输程序。

为了实现上述结构图，向应用程序提供统一的程序接口（API）。基于上述图结构，我们设计了一个 3G 安全服务类的基类 C3GSecurity。然后从基类 C3GSecurity 中派生出认证和密钥协商功能类 CAuGenKey 和重同步检测类 CReSynch。然后再从 CAuGenKey 类中派生出数据完整性保护类 CInteg 和数据加解密类 CEncrypt。他们中的相应功能的实现又是通过分别调用密码服务提供者（CSP）的相关密码函数来完成相应的功能计算，这样 3G 安全服务层进一步屏蔽底层安全算法的复杂性。在类的实现中必须提供调用底层算法的灵活性，还有就是要求底层函数的改变不会影响上层程序调用 3G 安全服务层程序的变化。这就要求在类方法的实现上不能直接调用 CSP 提供的底层函数，而是通过类封装实现底层函数功能向上层用户程序提供统一的接口（API），并且在调用 CSP 提供的底层函数时，可以根据用户的要求调用实现相同功能的不同函数，从而可以简化底

层函数的使用。3G 安全服务类结构图如图 4-9 所示。本文只提供最为基本的数据加解密、完整性、认证服务，未涉及密钥管理问题。

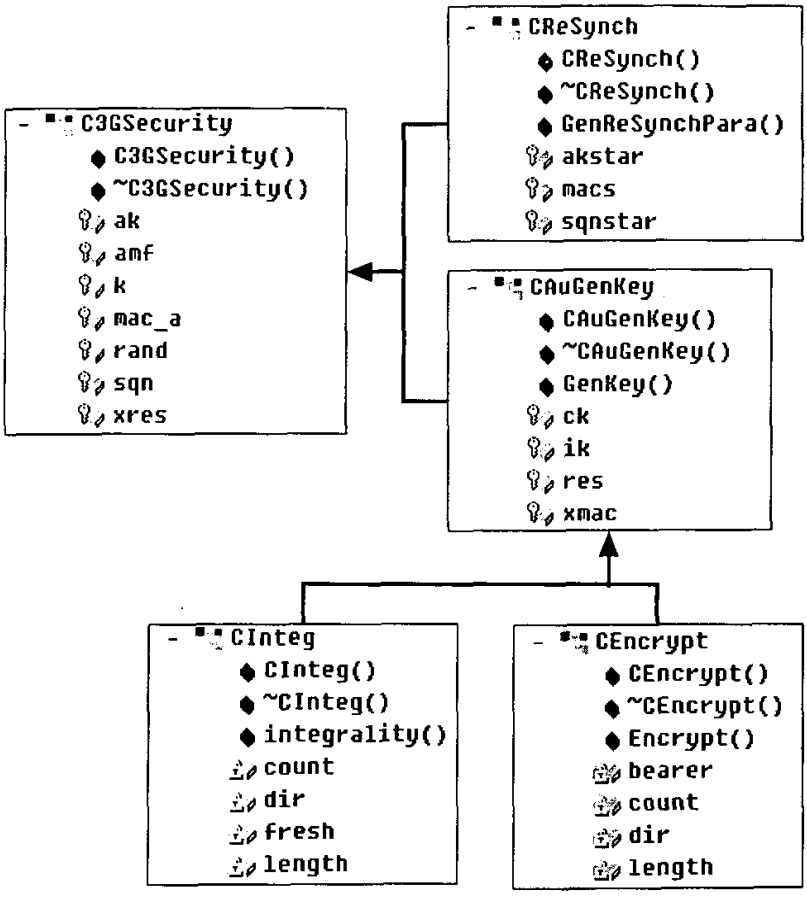


图 4-9　3G 安全服务层的 Class Diagram

Figure 4-9The Class Digram of the 3G SSL

4.2　WAP2.0 的安全服务

4.2.1　WAP 协议结构与网络模型

WAP 由一系列协议组成[42]，用来标准化移动通信设备，使新一代移动通信设备可靠地接入 Internet。在协议制定时采用了与因特网相似的分层结构。WAP 协议主要分为应用层、会话层、传输层、安全层和数据层，结构中的每层协议可以被上层的协议来访

问，这种分层结构为无线应用开发提供了可伸缩性的、扩展的环境。图 4-10 给出了 WAP 协议栈的结，同时给出了与有线网络协议的对比。可以看出 WAP 以 Internet 标准为基础，并对它的传输效率进行了优化，使 WAP 更适应于移动通信网络这一特殊环境。

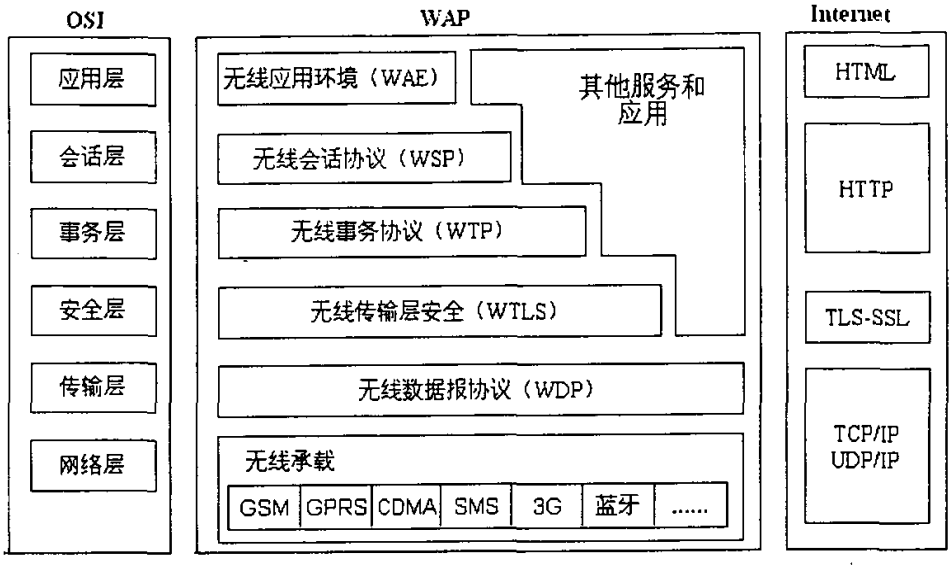


图 4-10 WAP 协议栈

Figure 4-10 WAP protocol stack

WAP 协议与因特网协议虽然结构相似，但具体协议不能直接通信。为了在 WAP 客户（支持 WAP 的无线设备如手机、PDA）与因特网服务器之间建立连接，WAP 规范采用了代理技术。WAP 代理可以实现如下功能：

（1） 协议网关：将源于 WAP1.X 版本中的无线协议如 WSP、WTP、WTLS 和 WDP 的请求转换成因特网 WWW 协议 HTTP 和 TCP/IP。此外，网关也负责查找客户请求中的服务器域名。

（2） 内容编码和译码器：内容编码器将 WAP 内容翻译成一种紧凑二进制格式，以便于充分利用窄带的无线信道。

（3） 管理代理客户档案：代理客户档案记录客户能力和个人嗜好，WAP 代理将这些资料组合起来，并将他们用于应用服务。

（4） 高效缓冲代理：通过一个有效维护一个经常被访问资源的缓冲器来改进网络访问性能，提高网络利用率。

基于代理技术的 WAP 网络包括 WAP 客户、WAP 代理和英特网服务器，其网络模型如图 4-11 所示。

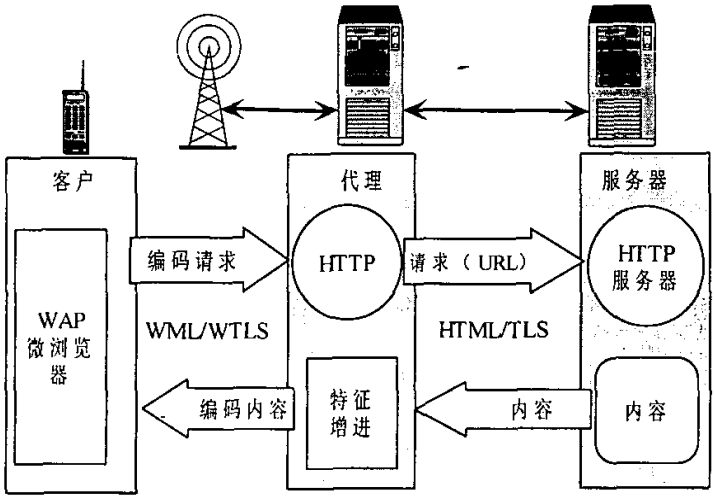


图 4-11　WAP1.X 网络模型

Figure 4-11 WAP1.X Network model

4.2.2　WAP 安全结构

WAP 中的所有安全问题均以 WAP 安全结构为基础。WAP 安全结构由 WTLS，WIM，WPKI 和 WMLScript 四部分组成[43]，每个部分在实现无线网络应用的安全中起着不同的作用，基于 WAP 的安全结构组成如图 4-12 所示。

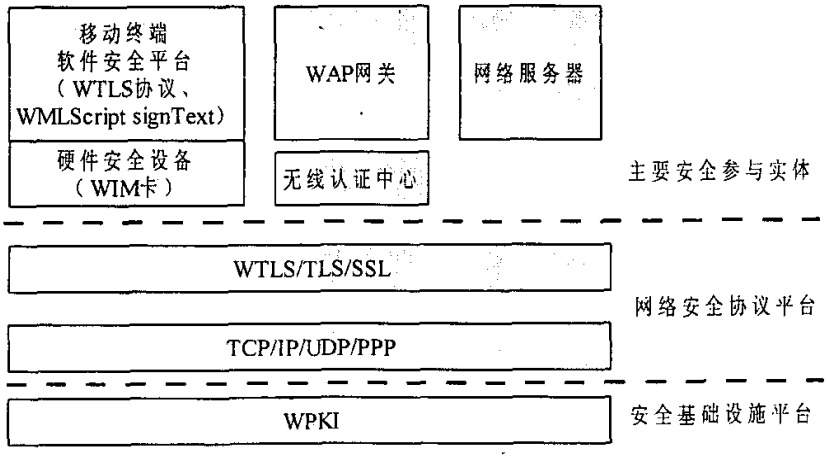


图 4-12　WAP 安全结构体系

Figure 4-12 WAP Security Structure System

其中，WPKI 作为安全基础设施平台，是安全协议能有效实行的基础。WPKI 和 WTLS，TCP/IP，WML 相互结合，实现身份认证、数字签名等功能。基于数字签名和私钥，WPKI 提供一个在分布式网络中的高度规模化、可管理的用户验证手段。网络安全协议平台包括 WTLS 协议以及有线环境下位于传输层上的安全协议 TLS，SSL 和 TCP/IP。安全参与实体作为底层安全协议的实际应用者，相互之间的关系也由底层的安全协议决定。

随着无线网络的发展和移动终端的进步到支持 IP 连接时，WAP 规范 2.0 版增加了对因特网协议（如 HTTP、TLS 和 TCP/IP 等）的支持，从而可以提供 TLS 隧道[44]来实现端到端安全。TLS 协议的无线改进版本允许安全交易操作，包括密码组、证书格式、签名算法和会话恢复的使用。它还规定了支持传输层端到端安全的 TLS 隧道方式。支持 TLS 隧道的 WAP 代理框架如图 4-13 所示。

在实施了承载层安全（例如 IPSec）的情况下，还支持直接访问 Web 服务器的方式（即直通方式）。图 4-14 给出了一个 WAP 设备通过 Internet 直接访问 Web 服务器的框图。无线 IP 路由器是 IP 网络的一部分，用来在一个链路层（如无线链路）和另一个链路层（如有线链路）之间转发 IP 分组。在直接访问的方案中，不需要使用无线优化的 TCP*协议。

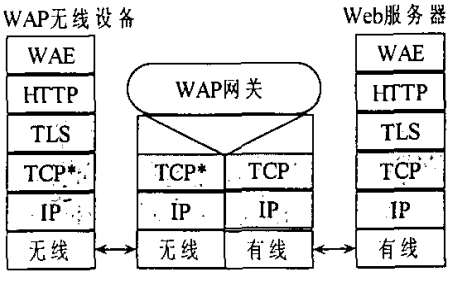


图 4-13 支持 TLS 隧道的 WAP 代理框架

Figure4-13 Supports TLS the tunnel the WAP proxy structure

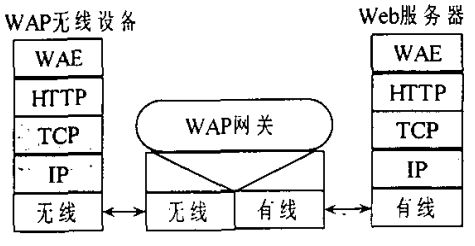


图 4-14 WAP 无线设备直接访问 Web 框图

Figure 4-14 User visits the Web diagram directly

在支持 WAP 规范 2.0 版的网络中，WAP 代理的工作大为简化，并且解决了 WAP1.X 中存在的“Security Gap”[45]，而且使得无线 Internet 和有线 Internet 的安全得到统一，更有利于实现电子商务所需的端到端安全服务，使得电子商务更加顺畅。

4.3 WAP2.0 安全服务层的设计

安全服务层中的 WAP2.0 安全服务层的设计主要是提供 WAP 安全框架中的签名函数、认证、数据保密性和数据完整性等服务。对于基于 TCP/IP 协议的移动通信网络，它们可以采取 IPSec 在网络层提供数据保密性，数据完整性和端到端保密性等安全性服务，采用 IKE 提供的身份认证服务。实现上述 WAP2.0 所需要的安全性服务，可以通过不同的密码技术在不同的协议层实现。图 4-15 给出实现不同的安全服务的 WAP 安全框架。

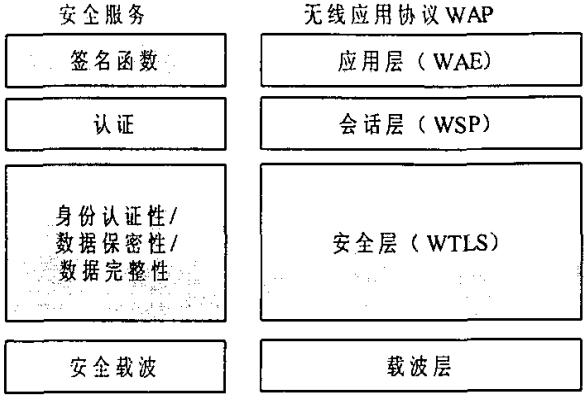


图 4-15 安全服务对应的 WAP 安全框架

Figure 4-15 Security service correspondence WAP security structure

签名函数：是采用 WML 脚本密码库[46]，它包含一组可以被 WML 脚本调用的安全函数，最主要的一个函数就是 SignText，提供签名服务。

WTLS 主要在传输层提供如下安全服务：

1. 数据保密性（Data Secrecy）：通过握手协议建立保密会话密钥，并采用加密算法加密数据报文。
2. 身份认证性（Identity Authentication）：握手协议中通过公开密钥证书来实现客户和服务器的相互身份认证。
3. 数据完整性（Data Integrity）：采用杂凑函数 MD5 或 SHA 来检测数据报文的完整性。

4.3.1 签名函数的设计

签名函数的设计主要是提供在应用层上的服务不可否认性服务。传输层的访问记录可以提供证据证明了用户登录了一个网页，然而访问了一个网页并不意味着用户进行了一次交易。签名函数在应用层提供数字签名来解决这个问题，当服务器要求用户对某一文本进行签名，它就调用 SignText，该函数将此文本和一个签名请求发给用户，用户利用其私钥对文本进行签名，并将签名返回服务器。

本项目支持RSA，DSA和ECCDSA签名，但由于移动终端的局限性，WAP规范建议采用ECCDSA签名。为了实现签名函数，我们设计了一个Crypto类，类中的签名函数的描述如下：

SignText 函数定义如下：*SignedString* = Crypto.SignText (*StringToSign, options, keyIdType, keyId).*

Parameters:	说明
stringToSign = String	StringToSign
options = Integer:	0x0001 – INCLUDE_CONTENT. 0x0002 – INCLUDE_KEY_HASH. 0x0004 – INCLUDE_CERTIFICATE.
keyIdType = Integer	Indicates the type of a key identifier: 0 – NONE. 1 – USER_KEY_HASH. 2 – TRUSTED_KEY_HASH.
keyId = String	Identifies the key in a way based on the previous parameter. the signature key
Return: string	SignedContent or Invalid.

Example: Var foo = Crypto.SignText ("Bill of Sale\n----------\n3 Bolognese $18.00\n1 Pepperoni $7.00\n4 Lemonade $6.00\n----------\nTotal Price $31.00", 0, 1,"\x37\x00\xB6\x96\x37\x75\xE3\x93\x48\x74\xD3\x98\x47\x53\x94\x34\x58\ x97\xB5\xD6"); // the application indicates the signature key。

Crypto类中的SignText 函数的实现的需要调用底层的CSP库的SHA-1哈希函数对待签名的文本产生摘要和调用CertSP证书库数字证书对摘要进行签名。签名的数字证书建议采用ECCDSA证书。

SignText 函数的实现中调用底层安全服务提供者模块(SSPM)时所用的格式定义如下：

Item:	Description:
null	No signature present.
rsa_sha_pkcs1	The signature is calculated according to [PKCS1], using octet string output
ecdsa_sha	The signature is calculated according to [X9.62], using octet string output.

- 签名算法枚举和签名算法的结构体如下：

```
enum {null (0), rsa_sha_pkcs1 (1), ecdsa_sha_p1363(2), (255)}
DataSignatureAlgorithm;
struct {
DataSignatureAlgorithm algorithm;
switch (algorithm)
{
case null: struct {};
default: opaque signature<0..2^161>;
};
}Signature;
```

- 签名证书枚举和签名证书的结构体如

```
enum { implicit(0), sha_key_hash(1), wtls_certificate(2),
x509_certificate(3), x968_certificate(4), certificate_url(5),
(255)}
SignerInfoType;
```

Item:	Description:
implicit	The signer is implied by the content.
sha_key_hash	The SHA-1 hash of the public key.
wtls_certificate	A WTLS certificate.
x509_certificate	An X.509v3 certificate.
x968_certificate	An X9.68 certificate.
certificate_url	A URL where the certificate is located.

```
struct {
SignerInfoType signer_info_type;
switch (signer_info_type)
{
case implicit: struct{};
case sha_key_hash: opaque hash[20];
case wtls_certificate: WTLSCertificate;
case x509_certificate: opaque x509_certificate<0..2^161>;
case x968_certificate: opaque x968_certificate<0..2^161;
case certificate_url:  opaque url<0..255>;
};
} SignerInfo;
```

4.3.2 WTLS层安全服务函数的设计

1. 握手协议和密钥生成函数

无线传输层安全协议WTLS协议[47]可以提供数据保密性、数据完整性和身份认证性等安全性服务，这些安全性服务的实现均与WTLS握手协议有着紧密地联系。一个完整的WTLS握手协议如图4-16所示。图中的“*”表明该项为可选项。

在通信开始时，握手协议主要实现如下功能：（1）协商密钥交换模式，选择密码算法；（2）根据应用需要，对通信一方或双方进行身份认证；（3）通信双方按照协商的密钥交换模式建立一个预先主密钥；（4）利用预先主密钥和随机数生成一个通信主密钥；（5）将所有密码安全参数传给协议记录层；（6）用户和服务器确认对方已经计算出相同的密码参数，且握手协议没有受到攻击者阻扰。

如果客户端和服务器端不想协商新的安全参数了，他们可以使用前面的安全参数，此时的握手过程变成图4-17 所示。

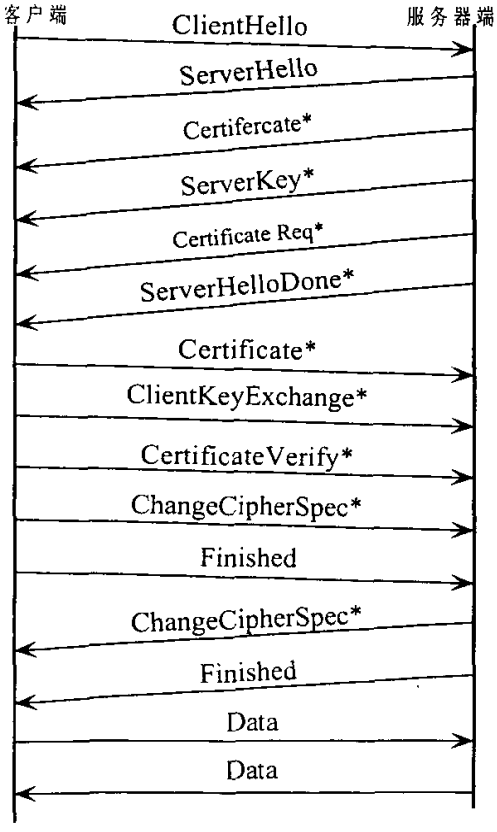


图4-16 WTLS握手流程

Figure 4-16 WTLS handshake flow

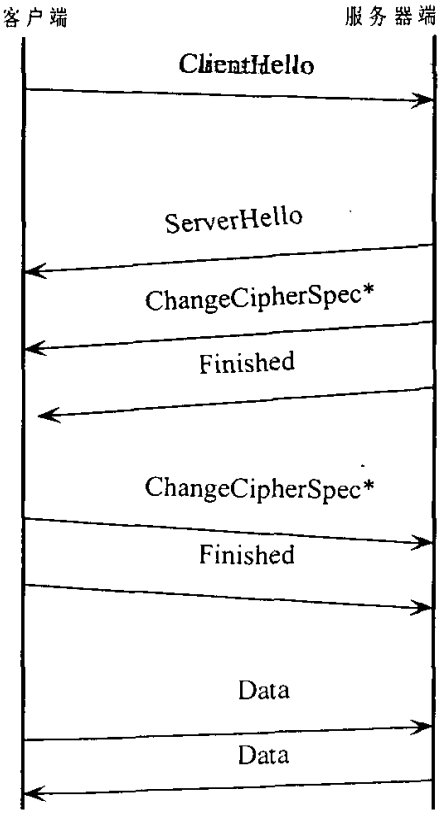


图4-17 WTLS简短握手流程图

Figure 4-17 WTLS short handshake flow

握手协议的消息格式如下所示：

```
enum{    /*消息类别*/
hello_request (0), client_hello (1), server_hello (2), certificate (11), server_key_exchange (12),
certificate_request (13), server_hello_done (14),
certificate_verify (15), client_key_exchange (16), finished (20), (255)
} HandshakeType;
struct{ /*消息内容*/
HandshakeType    msg_type;
Uint16 length;
switch (msg_type){
case  hello_request:    HelloRequest;
case   client_hello:     ClientHello;
case   server_hello:     ServerHello;
case   certificate:    Certificates;
case   server_key_exchange:   ServerKeyExchange;
case   certificate_requeset:   CertificateRequest;
case   server_hello_done:   ServerHelloDone;
case   certificate_verify:   CertificateVerify;
case   client_key_exchange: ClientKeyExchange;
case   finished:   Finished;
} body;
} Handshake;
```

密钥交换方法：

WAP规范了4种密钥交换方法：（1）共享秘密方法；（2）RSA加密传输方法；（3）DH密钥交换方法；（4）EC-DH密钥交换方法。

主密钥的生成：

在握手过程中通信双方交换的密钥被称之为预主密钥，用于生成保密会话所需的主密钥。在握手协议中预主密钥的生成方法各不相同，而将预主密钥转换成主密钥的方法则是相同的，即是调用底层的CSP伪随机函数生成。即：

master-secret = PRF(pre_master_secret, "master_secret", ClientHello.random+ ServerHello.random).

这里PRF代表底层CSP的伪随机函数。

主密钥的长度固定为20Bytes，而预密钥的长度则依密钥交换方法的不同有所不同，在主密钥计算出来后，预密钥立即被删除。

在一个恢复的会话中，主密钥与前一次会话中的主密钥相同。然而，由于握手协议中双方重新交换了随机数，这样新的会话将使用不同的密钥助。所谓密钥是指加密算法和MAC算法所需要的加密密钥，初始矢量和MAC密钥。

密钥组生成：当握手协议正式结束时，握手协议中生成的主密钥、选择的密码算法等信息均被传给WTLS层的记录协议。记录协议生成加密算法所需要的加密密钥和初始矢量，以及MAC算法所需要的MAC密钥。同样这些密码参数则是由主密钥、客户随机数和服务器随机数等参数经过调用底层的伪随机函数PRF产生而得，即：

Key_block = PRF (master_secret, expansion_label, seq_num+server_random+client_random)

将上述过程重复多次，直到生成了全部密码参数（加密密钥，初始矢量和MAC密钥）。其中，expansion_label值在客户写和服务器写中使用不同的值；seq_num则取决于握手协议中所选择的密钥更新频率。

伪随机函数PRF的构造如下：PRF (secret, label, seed) =P_hash (secret, label+seed)。其中：P_hash为数据扩展函数，它的实现可以利用底层CSP的HMAC_hash函数通过多次迭代将密钥和种子扩展为任意长度的输出，HMAC_hash函数的实现具体采用那个hash函数构造在握手协议中选择。P_hash (secret, data)函数构造如下：

P_hash (secret, data) = HMAC_hash (secret, A (1) +seed) +
HMAC_hash (secret, A (1) +seed) +
HMAC_hash (secret, A (1) +seed) +……

其中，A（0）=seed, A(i)=HAMC_hash(secret,A(i-1))。
其中HMAC_hash的实现如下所示：
HMAC_hash（Data，Key）=Hash（Key XOR opd + Hash（Key XOR ipad +Data））；
其中，opad和ipad是分别出0x36和0x5C重复64次而形成的两个固定字符串。

2. 身份认证

有些应用环境要求在之间进行安全传输，在建立安全传输之间必须进行移动终端和网络服务器之间的相互认证。 为了解决移动终端的存储能力和计算能力的不足，WAP协议的认证支持两个格式的证书，在移动终端认证网络服务器时，采用了一种新的证书格式——WTLS 证书[48,49]。WTLS 格式证书是对 X.509 证书的简化，证书尺寸本身就比 X.509 证书[50]小得多，更适合移动终端的有限储存资源，而且 WTLS 采用了椭圆曲线密码算法（Elliptic Curves Cryptography,ECC）代替了 PKI 中的 RSA 算法，使得密钥长度大大降低，其 160 位密钥的安全性相当于 1024 位的 RSA 算法。

当移动终端要验证网络服务器时，网络服务器将自己的 WTLS 证书发给移动终端来证明自己的身份，移动终端必须事先拥有对网络服务器证书签名的根 CA 的 WTLS 证书，用根 CA 证书验证服务器证书。

当网络服务器要验证移动终端时，可以有两种方式可以实现移动终端身份认证：

（1） WTLS 终端之间的 WTLS 客户端证书认证；

（2） 用 WMLScript 中的签名函数 SignText 实现终端和网络服务器的安全连接。

这两种方法都要求移动终端拥有一个私钥和一个用户证书 URL。私钥必须保存在终端中，通常保存在 WIM 中。由于移动终端证书是被 WAP 网关和网络服务器验证，也就是在有线环境中进行验证，因此用户证书可以是常用的 X.509 证书，但是由于 X.509 证书尺寸较大，不太可能存储在 WIM 中，为了解决这个问题，WAP 中规定 WIM 中存储的不是终端的证书，而是终端证书的 URL，而证书则集中存储在证书目录中。当 WAP 网关和网络服务器需要验证移动终端身份的时候，根据 WIM 中的证书 URL 到相应位置取证书。

3. 数据保密性和数据完整性函数的设计

为了实现 WTLS 层的数据保密性和数据完整性，继续在 Crypto 类上添加三个函数，分别完成上述功能。

数据的保密性、数据完整性和拒绝服务保护函数比较简单，分别调用底层 CSP 函数即可。

（1） 数据的保密性函数：Crypto.Encrypt (InputData,OutData，EncryptAlgorithm，Key，def)。

Parameters:	**Description:**
InputData = * unsigned char	Plaintext if def=0 ； Ciphertext if def=1
OutData = * unsigned char	Plaintext if def=1 ； Ciphertext if def=0
EncryptAlgorithm = int	0；1；2；3；4；....
Key = * unsigned char	128bit
def = int	Encrypt def=0；Decrypt, def=1
Return=int	0,true and 1, false

其具体实现同样还得调用底层的 CSP 函数。加密算法枚举和选用加密算法的结构体如下：

```
enum {3DES(0), AES(1), RC5(2), RC6(3), IDEA(4), (255)}
EncryptAlgorithm;

struct {
EncryptAlgorithm Algorithm;
```

```
switch (Algorithm)
{
case 3DES: opaque 3-DES;
case AES:  opaque AES;
case RC5:  opaque RC5;
case RC6:  opaque RC5;
case IDEA:  opaque IDEA;
};
} EncryptAlgorithmInfo;
```

（2） 数据的完整性函数：Crypto.Integrity (Data, Key,HashAlgorithm)。

Parameters:	**Description:**
InputData = * unsigned char	Plaintext
Key = * unsigned char	128bit
HashAlgorithm = int	0；1；2；3；4；.....

Crypto.Integrity (Data, Key,HashAlgorithm)的实现也是依靠调用的HMAC_hash函数实现，HMAC_hash的构造根据握手协议中选择的hash算法调用底层的CSP函数库中hash函数实现。方法如下公式所示：

HMAC_hash（Data，Key）=Hash（Key XOR opd + Hash（Key XOR ipad +Data））；

其中，opad和ipad是分别由0x36和0x5C重复64次而形成的两个固定字符串。Hash法枚举和选用Hash算法结构体如下：

```
enum {MD5(0), SHA-1 (1), SHA-256(2), SHA-384(3), SHA-512(4),
(255)}HashAlgorithm;
struct {
HashAlgorithm Algorithm;
switch (Algorithm)
{
case MD5:       opaque MD5hash[16];
case SHA-1:     opaque SHA1hash[20];
case SHA-256:   opaque SHA256hash[32];
case SHA-384:   opaque SHA384hash[48];
case SHA-512:   opaque SHA512hash[64];
};
} HashAlgorithmInfo;
```

4.4 IP 安全协议（IPSec） 原理

因特网工程任务组（IETF）制定了用于IP安全的系列标准，这些标准称为IPSec[52]。IPSec主要有三个协议，即认证头(Authentication Header: AH)[53]、封装安全负载(Encapsulating Security Payload: ESP)[54]和密钥交换协议(Internet Key Exchange :IKE)[55][56]，用于提供数据认证、数据完整性和加密性三种保护形式服务。AH和ESP都可以提供认证服务，但AH提供的认证服务要强于ESP，而IKE主要用于对密钥进行管理，以及对算法、协议和密钥三个方面进行协商。

4.4.1 验证头部 (Authentication Header)

1. AH 认证报头的功能有：

（1） 为 IP 数据报提供强大的身份验证，也就是使实体与数据报相关联。

（2） 为 IP 数据报提供强大的完整性验证，以防止重换攻击。

（3） 通过公开密钥数字签名算法，为 IP 数据包提供不可抵赖的服务。

2. AH 头部格式：

AH 在“IP 头”和“高层域”之间。AH 的长度是可变的，但必须是 32 比特数据报的倍数。AH 域细分为几个子域，其中包含为 IP 数据包提供密码保护所需要的数据。其格式如图 4-18 所示。

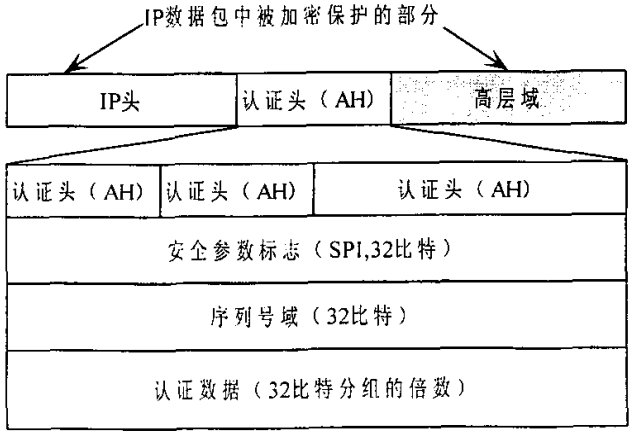


图 4-18 认证头（AH）的结构及其在 IP 数据包中的位置

Figure 4-18 Authentications (AH) structure and it's in IP data packet position

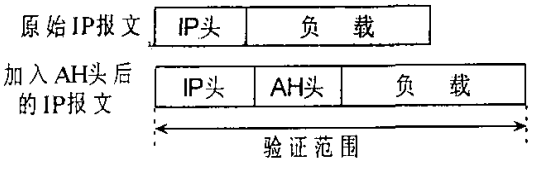
认证（实际上是指带有源身份信息的数据完整性）是IPSec的强制性服务，提供该保护所需的数据包含在AH的两个子域中。其中一个子域称为“安全参数标志”（SPI），该子域包含32比特的某个任意值，指出（唯一标志了）用于该IP数据包认证服务的密码算法。另一个子域称为“认证数据”，包含消息发送方为接受方生成的认证数据，用于接受方验证数据的完整性（因此这部分数据称为完整性校验值，ICV）。该IP数据包的接受方能够使用密钥和SPI惟一标志的算法重新生成“认证数据”，然后比较自己生成的认证数据和接受到的认证数据，完成ICV校验，其中使用的密钥采用IKE中协商好的。

子域“序列号”能够抗击IP数据包重放。其他子域没有安全方面的意义。

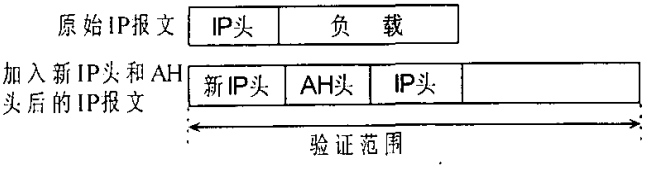
3、AH的使用模式

AH的使用模式有两种：

（1） 传输模式：只保护IP报文的不变部分



（2） 通道模式：保护整个IP报文



4.4.2 封装安全载荷（Encapsulating Security Payload）

1、ESP的功能有：

为IP报文提供数据完整性校验、身份认证和数据加密，还有可选择的重放攻击保护，即除提供AH提供的所有服务外，还提供数据加密服务。

2、ESP格式：

ESP主要是提供数据加密服务，它可以位于图4-19中AH之后的“高域区”。ESP的格式如4-19图所示。

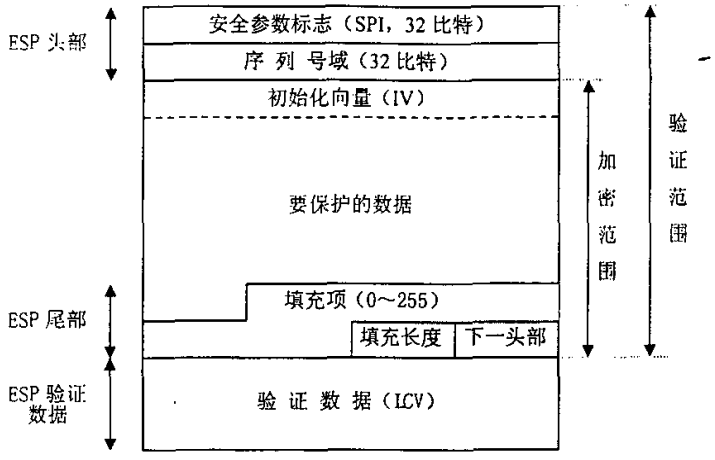


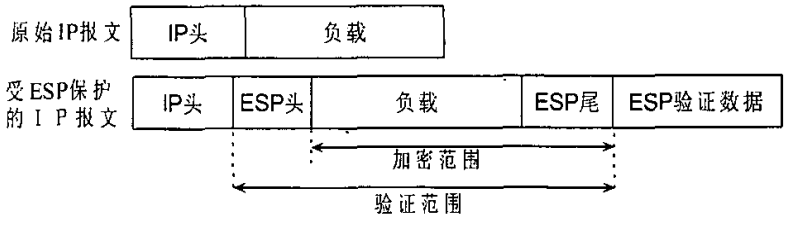
图 4-19 封装安全载荷（ESP）的结构

Figure 4-19 structure of the ESP

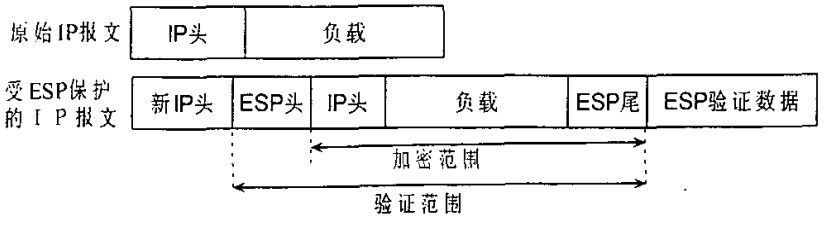
第一子域是“安全参数域”（SPI），在这里指出（惟一标志）所使用的加密算法。第二个子域“序列号”的含义与其在 AH 中的含义相同，第三个子域“要保护的数据”长度可变，其内容为机密数据的密文。因为 IPv6 数据包的长度是 32 比特的倍数，所以变长的“载荷数据”对应得明文需要填充，填充的数据在“填充数据”的子域中。最后，“认证数据”子域的含义与其在 AH 中的含义相同。

3、ESP 的使用模式

（1） 传输模式：只保护 IP 报文的不变部分



（2） 通道模式：保护整个 IP 报文



4、ESP 的处理：“先验证后解密”原则。

4.4.3 安全关联（SA）

在IPSec的中心概念中还有一个是“安全关联”（SA），SA用下面的数组唯一标志：（SPI，“IP目的地址”，“服务标志符”）。其中“服务标志符”之ESP或认证。

本质上，IPSec可以认为是AH+ESP。当两个网络节点在IPSec保护下的通信时，他们必须协商一个SA（用于认证）或者协商两个SA（用于认证和加密），并协商这两个节点间共享的会话秘密以便他们能够执行密码操作，具体参数有：（1）AH和ESP所用到的认证算法；（2）AH和ESP所用到的加密算法；（3）加密和认证时所使用的密钥；（4）加密密钥的生命周期；（5）SA自身的生命周期；（6）防止重放攻击所使用的序列号。该协商运用因特网密钥交换协议（IKE）来完成。

4.4.4 IPSec 安全服务层的设计

IPSec的设计方案由IPSec基本协议、密钥交换（IKE）、安全策略数据库、安全关联数据库等组成。IPSec的总体实现框架[57][58]如图4-19 所示。

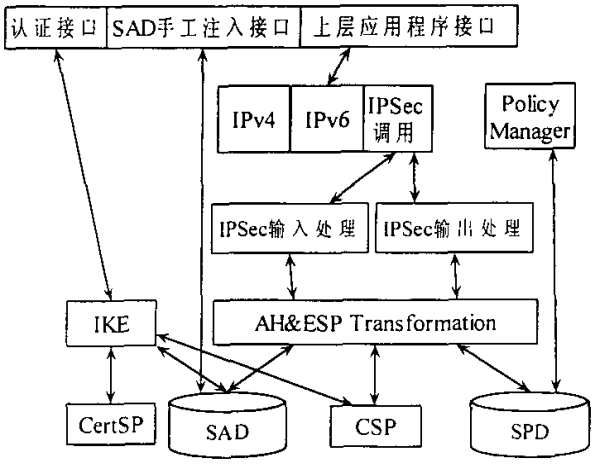


图 4-19 IPSec 的总体实现框架

Figure 4-19 Achieve overall framework ofthe IPSec

一般来说，在应用SA对通信数据进行保护时， IPSec会用到两个名义上的数据库：一个是SPD(Security Policy Database)，对于不同的数据，可能会用不同的安全策略来处理。有的数据必须加密，也有的数据不能加密，如的ICMP控制信息，或者有的数据应该被抛弃。SPD正是为不同通信数据制定不同的安全策略， IP层数据的类型、目标地址

以及不同的上层协议类型都有可能对应着不同的安全策略；还有一个是SAD(Security Association Database)，顾名思义，其内容就是SA，对“出去”的数据，其SAD的查询入口由SPD中的数据项来指向，对“进来”的数据，由上面所提到了唯一标识SA的三元组来定位的查询入口。

IPSec必须支持两类密钥和SA的管理方案：一类是手动管理；另一类是自动管理。手动的管理是指由人来配置一个系统和另一个系统进行安全通信时使用的密钥和SA，并手动维护这些密钥的协SA协商、更新以及删除。这种方案有其存在的理由，但更有价值的是自动的密钥管理方案。该方案需要管理密钥生命周期内的所有活动（包括生成、交换、更新和删除），需要维护SAD和SPD，还需要提供抗重放攻击的特性。密钥的自动管理可以使得协议的实现更具有普遍性，更有广阔的应用前景。所使用的自动密钥交换和SA的管理协议是IKE(Internet Key Exchange)，这是用Oakley密钥交换来实现ISAKMP(Internet Security Association and Key Management Protocol) 协议。后者是一个非常先进、全面和具有普遍意义的密钥管理框架。

IPSec安全服务的实现机制：

(1) 访问控制：IPSec使用身份认证机制进行访问控制，2个IPSec实体试图进行通信前，必须通过IKE(Internet Key Exchange)协商SA，协商过程中要进行身份认证，身份认证采用公钥签名机制，使用DSS算法或RSA算法，而公钥通常是从证书中取得。

(2) 数据源验证：IPSec使用消息鉴别机制实现数据源验证服务，在发送数据包前，要用消息鉴别算法HMAC计算MAC（消息鉴别码），HMAC将一部分消息和密钥作为输入，以MAC作为输出。MAC保存在IPSec头(AH、ESP头)的验证数据字段中，目的地收到IP包后，使用相同的验证算法和密钥计算验证数据，如果计算出的MAC与数据包中的MAC相同，则认为数据包是可信的，即通过验证。

(3) 无连接完整性和抗重播：IPSec使用数据源验证机制实现无连接完整性服务，IPSec根据IPSec头中的序号字段，使用滑动窗口原理，实现抗重播服务。IPSec数据包专门使用了一个单调递增的序列号及一个“滑动”的接收窗口，窗口的最左端对应于窗口起始位置的序列号，最右端对应于将来的第“窗口长度”个数据包，接收到的数据包必须是新的，而且必须落在窗口内部或靠窗口右侧，否则将其丢弃。

(4) 机密性和有限的业务流机密性：IPSec通过ESP协议中使用加密机制，保证数据的机密性，目前要求任何IPSec实现都必须支持DES。由于DES 已被证明在可计算性上是不安

全的，因此，未来的IPSec实现将支持3DES和AES(高级加密标准)，IPSec使用ESP隧道模式，对IP包进行封装，可达到一定程度的机密性，即有限的业务流机密性。

4.5 本章小结

本章主要是叙述支持移动多媒体的安全中间件中的安全服务层。该层延伸了现有的常规的安全中间件概念，从而可以使该层又可以成为一个中间件的形式存在。本层主要是提供 3GPP 标准和 WAP 协议提供的安全服务，基于移动网络的发展趋势是全 IP 核心网络，所以在安全中间的服务层中应该提供 IPSec 安全服务，现在本项目给出了一个 IPSec 的实现框架，具体尚未实现 IPSec 安全服务。

第五章 安全服务提供者模块的设计和实现

5.1 安全服务提供者概述

安全服务提供者模块(SSPM)应由基础安全服务提供商（把密码参数存储在USIM卡或智能卡上）以基础安全服务并且符合CSM接口规范的形式提供。CSM核心在需要的时候会把安全服务提供者模块中的相应子模块加载到内存中。CSM核心在收到用户的安全服务请求时，会按照用户的请求运行相应的安全服务提供者模块的子模块的对应任务函数，完成对用户的请求操作，将结果返回给用户。CSM核心在不需要该模块的时候，把它从内存中卸载，以释放该内存，提高内存利用率。

在SSPM和CSM之间，可以考虑为每种操作设立一个接口，例如算法服务提供者CSP和CSM之间的接口有加密接口、解密接口、签名接口、验证接口等等。我们为每种SSPM都设计了接口，在CSM中还只须判断对于特定的请求使用SSPM的哪个接口来产生服务。CSM核心在使用模块的服务功能时，只需要根据应用层的要求选择相应的接口并调用SSPM中合适的函数即可。这样使整个过程显得层次分明、简洁明了。

根据现有的无线网络所要求的安全服务的特点，安全服务提供者模块根据其提供的安全服务类型不同可分为以下几个种类，包括:密码服务提供者(CSP)、证书服务提供者(CertSP)、可信模型提供者(TSP)、数据存储提供者(DSP)以及其他服务提供者(ESP)。本项目目前提供了密码服务提供者(CSP)和证书服务提供者(CertSP)的示例，它们主要是作为安全服务提供者模块的开发模板使用，测试CSM核心的功能，同时也具有一定的安全服务能力。

CSP主要功能是用于生成密钥；对用户的数据作出摘要；对用户的数据进行加密、解密；对用户的数据进行签名、验证。CertSP主要用于证书的管理、证书信息的获取、证书公钥的获取、证书的验证。下面将详细介绍这两个模块的实现。

5.2 密码服务提供者（CSP）

5.2.1 CSP 体系结构

CSP即Cipher Service Provider,密码服务提供者，它是安全服务提供者模块的核心部件，同时也是整个安全中间件最为基础的部分。在功能上完成基于密码学的各类安全应用，如数据加密、解密、哈希生成、数字签名、验证、生成随机数、生成密钥等各种算法服务，采用分层结构，如5-1图所示。

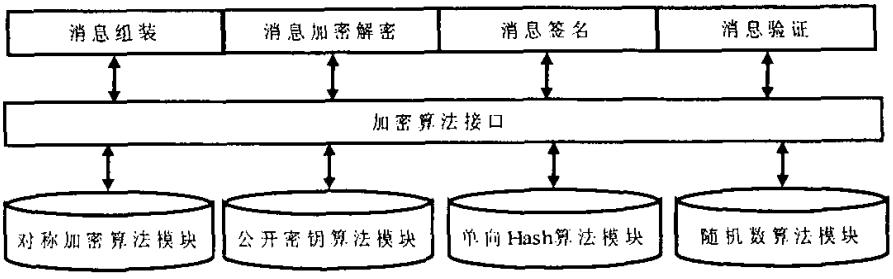


图 5-1　CSP 模块结构图

Figure 5-1 Modular structure of the CSP

（1）　对称加密算法模块：提供无线通信所要求的高强度的分组和流密码算法加密算法（如 AES、$f1...f9$，3-DES 等），且支持各种加密方式（如 ECB、CBC、CFB、OFB）。密钥长度的选择要同时兼顾算法安全与执行速度。

（2）　公开密钥算法模块：提供 Diffie-Hellman、RSA 和 DSA 与 ECC 四种可选的系统，公开密钥算法是安全认证、数字签名及证书应用的基础，同时也用作分配密钥之用。而 ECC 算法是计算资源有限的移动终端的最佳选择。在大数处理时应用中国余数定理来优化设计，签名方案采用 DSA 或 ECDSA（最佳）。

（3）　单向 Hash 算法模块：提供 MD5、SHA-1、SHA-256、SHA-384、SHA-512 等算法。提高了应用中 Hash 运算的正确性和安全性。Hash 算法是数据完整性的安全保障同时也是数据签名的基础，主要完成单向哈希功能计算。

（4）　随机数算法模块：采用伪随机数序列设计，在设计接口上考虑了对第三方密码服务的支持，随机数应用于密钥的生成，随机数的好坏直接影响到信息的安全，所以密钥生成的安全性关键就取决于随机数发生器的设计。

5.2.1 CSP 中算法接口和实现

CSP 部分提供各种各样的密码服务，不仅密码算法种类不同，有加密、解密、签名、验证、生成随机数、生成密钥等等，而且，同一算法又包含多种模式。作为一个功能完善的算法服务提供者，应该根据无线安全通信的要求提供各种各样的算法和各种算法的模式，而且不能排除在今后为 CSP 添加其它未知算法的可能。 因此对于 CSP 中的密码算法实现有以下两点要求：

第一:对于输入数据要能够灵活地选择不同的算法来处理得到输出数据；

第二:要能够很方便地扩展所支持的算法，加入新的算法支持。

对于这样的应用要求，加密库的设计采用了同一算法结构，具备良好的扩展性和通用性，算法信息采用通用数据结构定义。本人利用了C++语言的特性，使用了设计模式中的算法（Algorithm）模式。即设计一个所有算法的基类，提供一个处理数据的虚函数。所有实际的算法类都从这个算法的基类派生出来，对处理数据的虚函数作重载，实现自己对本算法的处理。

本项目中把密码学算法分为对称算法、非对称算法、Hash算法，随机数算法四类。每个类都是从抽象算法类CAlgorithm中派生出来，同时包含了自己本类算法中要使用的安全上下文。例如在对称算法中，成员变量含有密钥、IV、明文数据块的首址、密文数据块的首址，以及它们的长度。从这四种算法类中再派生具体的算法类，例如从对称算法类CSymAlg中又派生出*CAesEncrypt*、*CAesDecrypt*、*Cf8Encrypt*、*Cf8Decrypt*等几个类；分别实现具体的不同算法的加密和解密。如果要扩充新的算法类型可以从CAlgorithm中再次派生出一个新类。如果要在这四种算法类型中添加新的算法，可以从新算法所属的算法类型中派生出新的类。图5-2是CSP中算法类的类图，省略了其中部分的具体算法类。

在拥有了这样的类结构后，如何才能选择一个合适的算法类成了另一个问题。从本质上说，具体算法类的选择是按照调用者所发出的命令来决定的。也就是说根据队列单元中的命令、算法、算法模式等条件判断，选择一个合适的算法类。最简单的方法是采用switch...case结构，但是这样作会产生极长的switch... case结构，从而降低程序的可读性。所以在设计模式中的算法(Algorithm)模式常

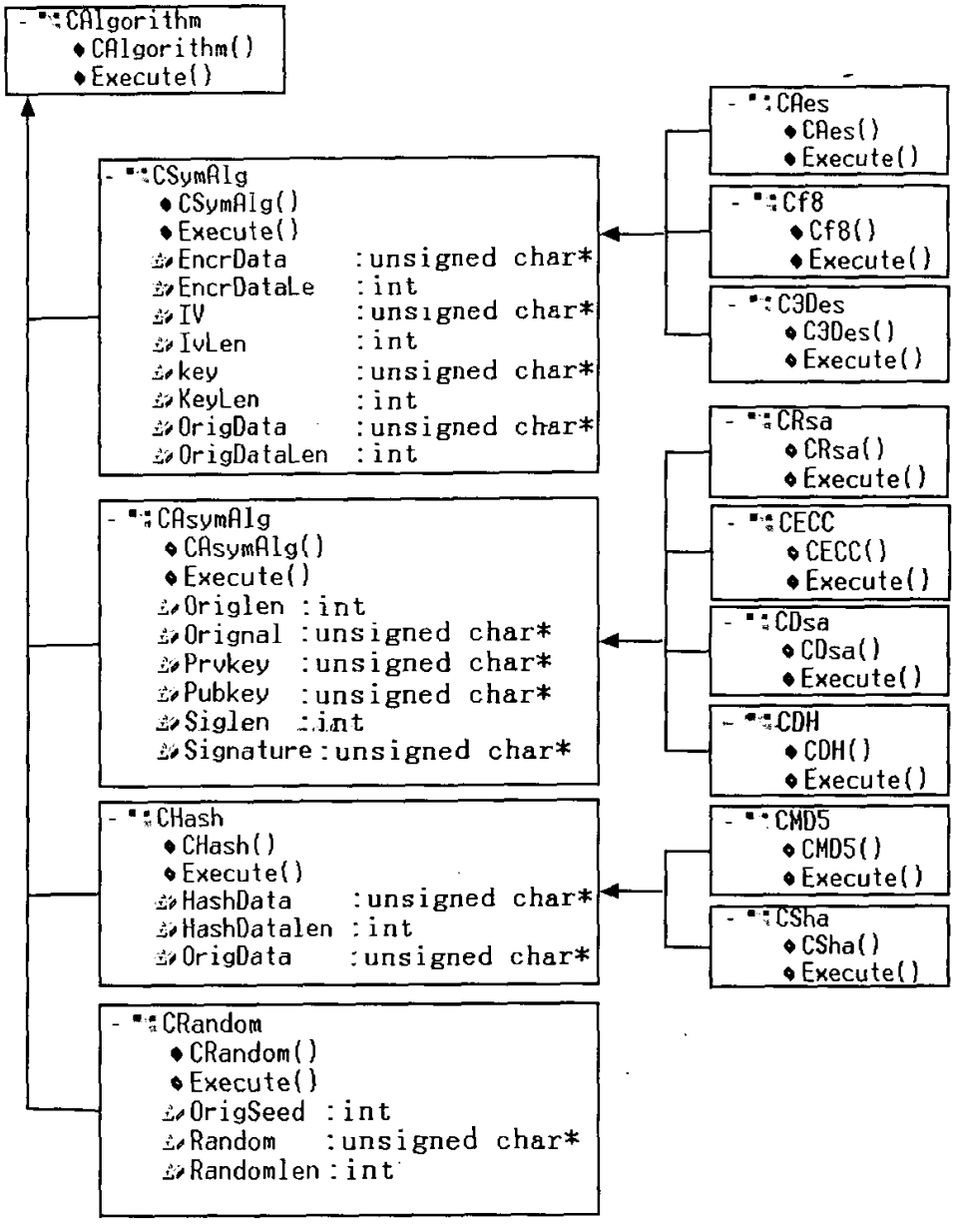


图5-2　CSP中的算法类的Class Diagram

Figure 5-2 the Algorithm Class Diagram of the CSP

常是伴随着工厂(Factory)模式一同使用，虽然这样做，可能多做一些工作，但会给你系统带来更大的可扩展性和尽量少的修改量。其实就是使用一个单独的类来检查上下文，判断应该使用的类，并生成类的实例。CSP中这部分的类图如图5-3所示。

其中CControlCipher是整个算法模块的控制类，也是算法模块和外部的接口。它的私有成员变量中包括输入数据首地址InData、输出数据地址OutData、密钥

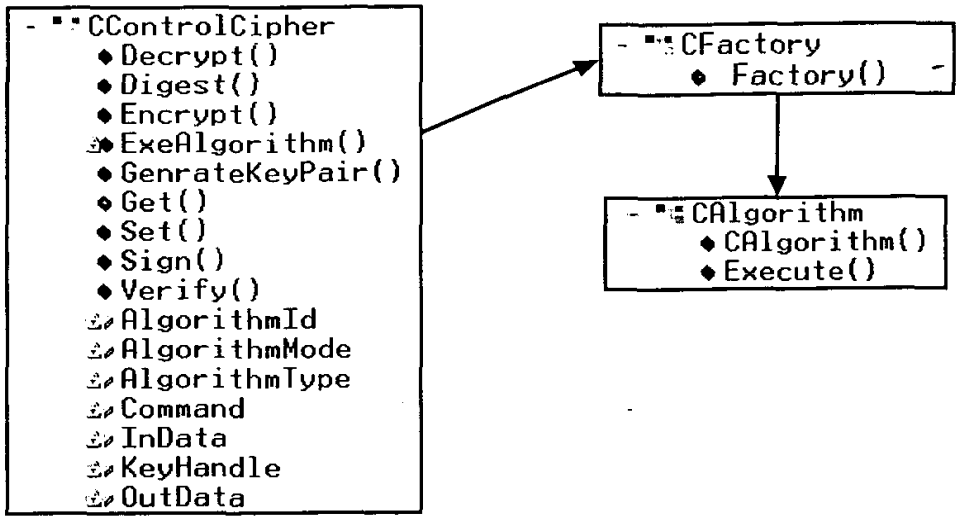


图5-3 CSP中控制类的Class Diagram

Figure 5-3 the Control Class Diagram of the CSP

句柄KeyHandle、请求命令Command、算法编号AlgorithmId、算法类型AlgorithmType、算法模式AlgorithmMode。基本上是对应着请求/响应队列单元中的各项数据。CControlCipher有一系列的Set和Get函数，用于读取和设置这些成员变量的值。CSP在使用了ParaParse分析了输入参数后，把得到的数据存放到CControlCipher类的各个成员变量中。

CControlCipher还有一个友元类CFactory类（即工厂类），它只有一个方法Factory，用于生成算法类的实例。CControlCipher可以直接察看CControlCipher中的各成员变量，因此当CFactory因此当CControlCipher以自己的引用为参数调用Factory方法时，CFactory类可以根据CControlCipher的成员变量来决定生成那一个具体的算法类。实际上CFactory类是把繁复的算法选择条件封装在了自己之内。如果要扩充CSP的算法类，不仅要添加CAlgorithm类的子类，也要在Factory方法中添加新的判断条件，使得CFactory类在条件符合时，生成新的算法类的实例。

CControlCipher中还有一系列的算法接口，例如Encrypt、 Decrypt等等，和一个私有方法ExeAlgorithm。ExeAlgorithm函数的参数是一个CAlgorithm类的指针，在它的内部用CControlCipher的输入数据地址和输出数据地址作参数，执行这个CAlgorithm类的Execute()函数，用该算法处理数据。

因此算法接口内需要先调用CFactory类的Factory方法，由CFactory类决定生成什么算法类。CFactory类把生成的算法类向上映射为CAlgorithm类，返回给CControlCipher，

CControlCipher再利用成员函数ExeAlgorithm执行得到的算法类的Execute()函数，从而完成算法服务任务。使用工厂(Factory)模式流程如图5-4所示。

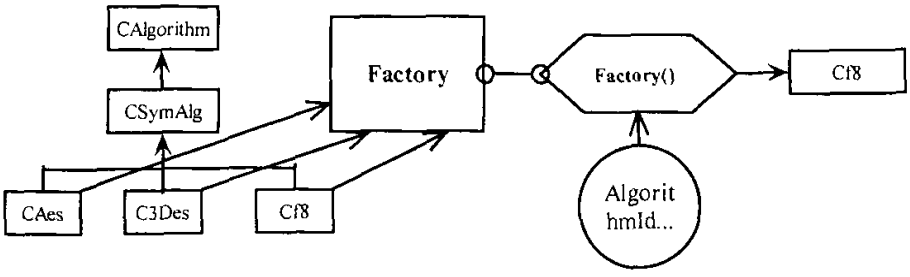


图5-4 工厂(Factory)模式流程

Figure 5-4 The Factory Pattern Flow

5.2.2 CSP 中密钥的生成和管理

CSP 中使用的密钥主要有对称密钥和非对称密钥两类。为了使用安全合方便，密钥和其他身份信息一般存贮在 WIM（无线个人身份模块）卡或 USIM 卡上或 USIM-WIM 组合卡上。对称密钥和 PrivateKey 的生成来自于随机数函数，即随机函数按相应的加密算法要求生成合适的随机数作为对称密钥或 PrivateKey。而 PublicKey 的生成源自于对应的非对称算法类。

在本项目中没有 WIM 或 USIM 卡存储密钥，取而代之的是把密钥放入密钥栈中（一个随机存储栈），把密钥在栈中的序号传递给用户；用户使用时传入的也只是堆栈中的序号而已。

由于密钥长度的关系(RSA一般为1024位或更长)，在C/C++中没有相应的数据类型来表示，所以在实现时专门定义了一个数据类型big表示大整数，big的运行规则基于Knuth's algorithms[59]。

为了方便密钥的产生和管理，我们专门设置了一个密钥基类 BaseKey，并从基类派生出对称密钥类 CSymKey，和非对称密钥类 CAsymKey。密钥栈中使用了指针数组，生成出来的对称密钥或非对称密钥的地址被放在密钥栈的指针数组中。在使用密钥时，由密钥的序号，在栈中得到密钥，然后由运行时类型识别来判断密钥的类型是否正确，密钥生成和管理的 Class Diagram 如图 5-5 所示。

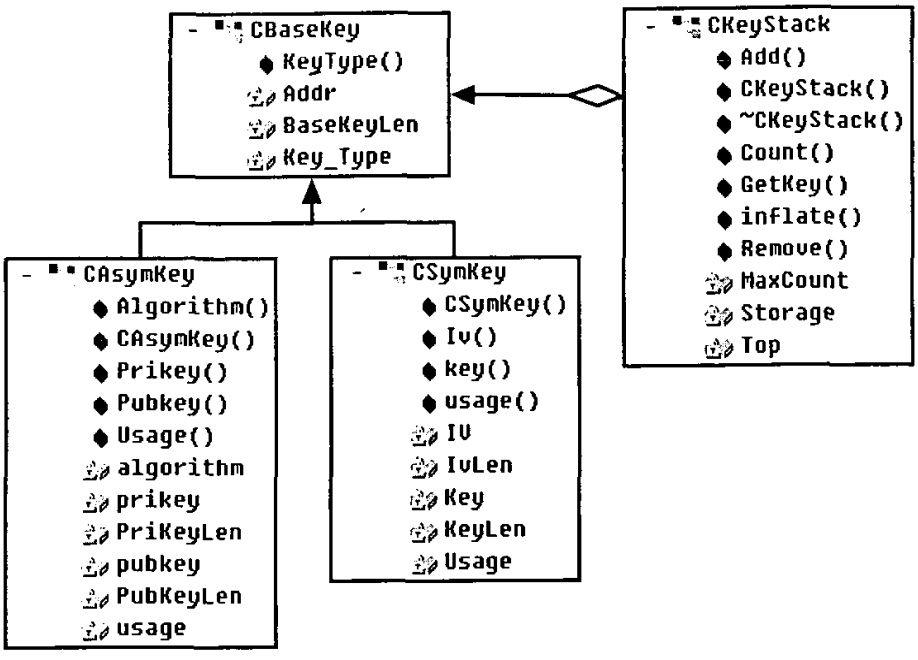


图 5-5 密钥生成和管理的 Class Diagram

Figure 5-5 Key Generation and Management Class Diagram

CBaseKey 是密钥的基类，主要用途是派生出对称密钥类和非对称密钥类。其中，KeyType 用来设置和获得 CBaseKey 对象的类型，用于密钥类的运行时类型识别。CSymKey 和 CAsymKey 两个类继承自 CBaseKey，分别封装了对称密钥和非对称密钥的相关函数。CSP 中编码或解码中用到的密钥实际就是用这两个类来存储。CKeyStack 是密钥堆栈类，其中的 Storage 是一个尺寸可以扩展的、指向 CBaseKey 类的指针数组。使用 Add 函数，可以把 CSymKey 和 CAsymKey 对象加到 Storage 数组中。GetKey 函数用于从 Storage 中获取密钥对象。

5.3 证书服务提供者（CertSP）

证书服务提供者（CertSP：Certificate Service Provider）用来完成证书的管理、验证、提取信息、提取密钥等工作，在整体构架上与CSP相似。证书管理功能类似于CSP的密钥管理，只是把密钥的句柄换成了对证书的句柄；任务的分发与CSP如出一辙；当然具体执行的任务不是加密而是证书功能。CertSP自己本身最大的特点是对CSP的调用。证书服务提供者的整体框架如5-6所示。

CertSP与CSP最大的区别是：CertSP在执行时要调用CSP的功能，即CertSP

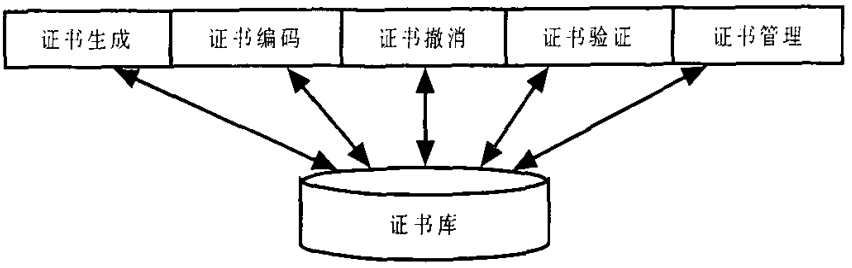


图5-6 证书服务提供者的整体框架

Figure 5-6Structure of Certificate Service Providers

一方面实现安全服务提供者模块的功能，另一方面也扮演用户进程的角色。CertSP的实现测试了CSM体系结构中，一个SSPM对另一个SSPM的调用能力。CSM体系结构一个重要的特点是各模块间的对等关系，即CSM对各模块的调用接口是相同的，不论是对CertSP或CSP都使用相同的接口调用；CSM对各进程也提供了相同的调用接口，不论是独立的用户进程，或是CSM进程空间中的安全服务提供者模块都可以使用相同的接口来调用SSPM的服务功能。因此，CertSP可以使用密码服务提供者模块中的函数，像一个普通的用户进程一样调用CSP的功能。

CertSP本身没有提供对密钥的管理功能，而它需要提供从证书中提取密钥的能力，而且提取出的密钥需要可以在CSP中使用(实际上,用证书中的公钥进行验证和解密提取密钥是证书的两个主要用途)，也需要能通过用户提供的密钥句柄对证书进行验证。因此，CertSP如同一个普通的用户进程一样，利用通用安全编程接口，使用CSP的密钥管理能力。

CertSP在加载的时候，会利用RIS检查系统中有没有安装CSP，如果没有安装，CertSP会跳过其后的操作并返回出错消息。如果系统中安装了CSP，CertSP会对CSM核心发出Load和Attach的命令，让CSM核心加载CSP模块。

当用户要提取证书中的密钥时，CertSP从证书中提取出公钥，然后利用CSM核心调用CSP的KeyDecode功能，把这个公钥存入CSP的密钥堆栈，并把从CSP中返回的密钥句柄返回给调用CertSP的用户进程。

当用户要用给定的公钥句柄验证证书时，CertSP利用CSM核心调用CSP的KeyEncode功能，得到公钥句柄指向的公钥，然后再用该公钥完成证书的验证工作。

在本项目中证书服务提供者支持X.509和WTLS两种证书。

5.4 算法测试

安全服务提供者模块(SSPM)提供了无线通信中常用的密码学算法，对称算法包括3-DES，AES，IDEA，RC6 和 3GPP 中的 $f1, f2, f3, f4, f5, f8, f9$，主要用于产生通信主密钥和数据加密；非对称算法包括：Diffie-Hellman、RSA、DSA、和 ECDSA,主要用于数据签名和验证以及密钥交换。下面给出部分算法的执行效率的比较。

由于ECDSA 算法只需要大约160 位的模运算就可以获得与RSA 和DSA 的1024位的模运算相同的安全强度,所以ECC 算法需要存储的密钥位数减少了,不但节省了密钥存储空间,还减少了密钥交换所需的时间。从表5-1中可以看出,在相同的安全等级上,ECC 数字签名和验证算法的执行时间加起来要比RSA 和DSA 短很多。

表 5-1 签名算法执行效率比较

	公钥位数	Hash算法	待签数据量	产生公钥时间	签名时间	验证时间
RSA	1024 bit	SHA-1	369.760 KB	172 (ms)	62 (ms)	63 (ms)
DSA	1024 bit	SHA-1	369.760 KB	203 (ms)	259 (ms)	259 (ms)
ECDSA	160 bit	SHA-1	369.760 KB	15 (ms)	16 (ms)	31 (ms)

表 5-2 对称加密算法执行效率比较

	加密数据量	数据分组长度	密钥长度	所花费的时间
3DES	4,141,560 KB	128	128 bit	13500 (ms)
AES	4,141,560 KB	128	128 bit	6750 (ms)
RC6	4,141,560 KB	128	128 bit	6859 (ms)
RC5	4,141,560 KB	64	128 bit	17031 (ms)
IDEA	4,141,560 KB	64	128 bit	14256 (ms)
F8	4,141,560 KB	64	128 bit	13453 (ms)

表 5-2 给出对称加密算法执行效率比较。从表中可以看出，AES 的执行效率最高。3G 安全标准中就是推荐使用 AES-128 来产生身份认证和密钥生成算法。在本项目中，推荐使用 AES-128 加密算法。

上述表格数据是在 CPU 为 Intel（R）4 3．00*2GHz，内存 512M 环境下的 HP7650 机上测试所得。

5.5 本章小结

SSPM是安全中间件中完成具体安全服务的软件模块，是安全中间件对外功能的具体体现的实际实施者，是安全中间件的最基本模块。它以动态链接库的形式提供，按照服务内容的不同，分为密码服务提供者(CSP), 证书服务提供者(CertSP)、可信模型提供者(TSP)、数据存储提供者(DSP)以及其他服务提供者(ESP)。本项目到现在为止暂且只能提供CSP和CertSP模块部分，其他部分在此不作分析。

算法服务提供者提供了在无线通信中可能遇到的各种算法，包括加密、解密、签名、验证、摘要等等。对于CSP中要使用的各种算法，在设计上使用了模式中的算法模式，便用一个公共的基类CAlgorithm来提供统一的接口，然后所有的具体算法都从这个类派生出来，再使用工厂模式，在运行时生成合适的具体算法类。以这样的方式掩盖了各种算法之间的差异，在程序的主体部分只关注对数据的处理过程，而不用关心具体处理的方法。使用这样的设计模式，使得CSP具有良好的可扩展性。

在CSP中还有密钥管理的功能，它把所有的密钥全部管理在自己的内部，用户请求获得密钥或指定密钥用于运算时，使用的都是CSP分配的密钥句柄。在CSP中使用了一个抽象类作为其它密钥类的基类，对称密钥类和非对称密钥类都是从这个基类中派生出来。使用一个密钥类的容器类来包含各种密钥类，完成对各种密钥的管理。

证书服务提供者用来完成证书的管理、验证、提取信息、提取密钥等工作。它的结构与CSP是很相似的。最特殊的地方在于它在使用中会调用CSP的功能完成自己的服务，即CertSP一方面完成SSPM的功能，另一方面也充当着用户程序的角色。CertSP的成功证明了在安全中间件体系中各模块的平等性，SSPM之间可以互相调用。

最后给出部分加密算法和签名算法的执行效率，从执行效率的角度推荐最佳的加密算法和签名算法。

第六章　混沌序列在 3G 安全通信中的应用

混沌现象是非线性动态系统中出现的一种类似随机的运动形式。一般而言，混沌现象隶属于确定性系统但又难以预测，它具有非线性、确定性、蝴蝶效应、不规则性、难以预测性等特性，由于它的这些特性特别适合应用于保密通信，所以在 80 年代末期开始受到密码学界的重视。尽管目前混沌保密通信技术的研究仍处于实验室阶段，但由于混沌保密通信具有实时性强、保密性高、运算速度快等明显优点[60]，已显示出其在保密通信领域中的强大生命力。随着研究的深入，出现了多种利用混沌进行保密通信的方法。迄今利用混沌进行保密通信大致可分为两大类：第一类是利用混沌同步的混沌进行保密通信；第二类是混沌数字保密通信。本文综合利用混沌特性和现代密码学的优点，基于“一次一密”的思想，在不显著增加计算时间与尽可能不改变 3G 原有的安全结构的情况下，提出了一种改进的安全通信方法，它利用混沌映射产生的混沌序列与 3G 通信安全中的保密密钥相结合，定时的变换初始密钥来提高通信安全的数据保密性。

6.1　Logistic 映射产生的混沌序列和性能分析

Logistic 映射是一个典型非线性混沌系统[60]，它虽然简单但却体现出混沌运动的基本性质。Logistic 映射系统方程式如式（6-1）：

$$X_{i+1} = f(X_i) = \lambda X_i(1 - X_i), \quad X_i \in [0,1], \ \lambda \in [0,4] \qquad (6\text{-}1)$$

其中 λ 是控制参数，X_i 是状态，f 把当前状态 X_i 映射到下一状态 X_{i+1}，当 λ 确定以后，可以由任意的 $X_0 \in [0,1]$ 迭代出一个确定的时间序列 X_1，X_2，…，X_i。对于不同的 λ 值，系统（6-1）呈现不同的特性，随着 λ 参数的增加，系统（6-1）不断地经历倍周期分叉，当 $\lambda \in (3.5699456,4]$ 时系统进入混沌状态。本文关心的是满映射 λ=4 时的情况，Logistic 映射的输入输出都分布在（0，1）上。当 $i \to \infty$ 时，Logistic 映射序列的概率分布密度函数[61]为（6-2）式所示：

$$\rho(x)=\begin{cases}\frac{1}{\pi\sqrt{x(1-x)}} \\ 0\end{cases} \tag{6-2}$$

式（6-2）表明式（6-1）所确定的序列具有遍历性，并且它产生的序列的概率密度分布函数与初始值无关。所以由系统（6-1）产生的序列$\{X_i\}$运动形式具有典型的下列混沌特征[62]：

1．**随机性**：当λ=4时，Logistic映射在有限次迭代内呈现不稳定运动，但其后长时间的动态行为将显示随机性质。从统计特性上可得，满映射λ=4时(初值为0．5000001，迭代1000次)产生的混沌序列均值为0．4970，自相关特性图如图6-1所示，互相关特性图如图6-2所示（每个迭代值减去均值0.4970），由此可得其概率统计特性与白噪声一致，是理想的密码序列。

2、**规律性**：尽管$\{X_i\}$体现出随机性质，但它是由系统方程（6-1）式迭代得到的，只要初值X_0确定后X_i便已确定，即其随机性是内在的。

3、**遍历性**：混沌运动的遍历性是指混沌变量能在一定范围内按其自身规律不重复地遍历所有状态。

4、**具有分维性**：混沌的奇异吸引子在微小尺度上具有与整体自相似的集合结构。

5、**对初值敏感性**：对初值的微小变化将导致序列$\{X_i\}$远期行为的毫无相关性。图6-3给出的是两个初值分别为：$X_{10}=0.5000001$，$X_{20}=0.5000002$以迭代50次得到的序列$\{X_{1i}\}$和$\{X_{2i}\}$。由图可以看出，初始值虽然相差了0．0000001，但是迭代了大约40次以后得到的序列便完全不一样了。

为了能用混沌序列进行数字加密，必须把离散的混沌序列转化为混沌的二值序列，定义如下：$Y_i=1$ ，$X_i>U_i$时：$Y_i=0$，$X_i\le U_i$时。其中：i=0, 1, 2, 3… n。U_i为量化阀值，它的选取对序列的性能有着极为重要的性影响。在本文中，提出了一种动态的阀值法。首先迭代N（比如N=10000）次进入混沌状态，取$U_0=\frac{1}{N}\sum_{n=1}^{N}X_n$，作为产生$Y_0$的量化阀值，然后$U_{i+1}=\frac{1}{2}(U_i+X_i)$作为产生$Y_{i+1}$的量化阀值。由于$X_i$的随机性，所以$U_i$也具有随机性。图6-4为选取的30个量化阀值的取值情况。

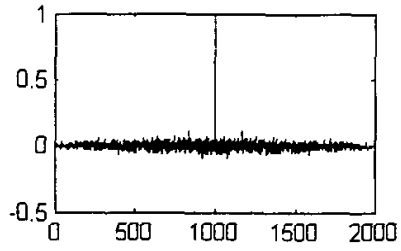


图 6-1 自相关特性图

Figure 6-1 Autocorrelation Characteristic Chart

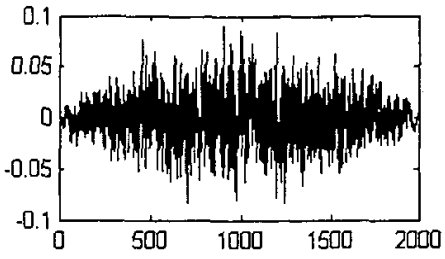


图 6-2 互相关特性图

Figure 6-2 Mutual Correlation Characteristic Chart

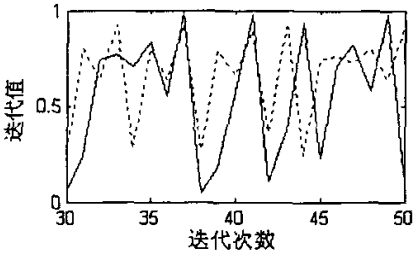


图 6- 3 对初值敏感测试图

Figure 6-3 To Starting Value Sensitive Test Chart

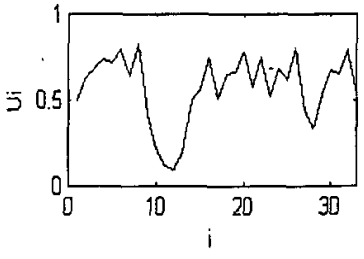


图 6-4 量化阈值 U_i

Figure 6-4 Quantification Valve

6.2 改进后的 3G 中的安全认证与密钥协商

USIM 卡中改进后的认证与密钥生成框图[36]如图 6-5 所示。

改进后的 3GPP 的身份认证与密钥生成过程[37]可以描述如下：

（1） 当移动用户第一次访问网络时，传送其国际移动身份号 IMSI 给 VLR。

（2） VLR 对该用户进行登记，并利用其 IMSI 从 HLR 调该用户的认证矢量 AV（AV 中新增加一个安全参数 K*以供产生混沌序列之用）。

（3） HLR 利用用户的 IMSI 从 AC 数据库调用该用户的 AV 并传给 VLR。

（4） VLR 从 AV 中取出 RAND 和 AUTN,传送给用户。

（5） USIM 先认证网络端的身份，如认证成功，USIM 生成移动用户身份认证响应 RES,并传给 VLR，同时生成完整性密钥 IK 和保密密钥 CK。完成 IK 和 CK 的计算后进行混沌的迭代产生 K_Chaos，以供改变 CK*之用。

（6） VLR 从 AUTN 中取出 AC 生成的 XRES，比较 RES 是否与 XRES 相同；如果相同，VLR 为移动用户分配一个临时的身份号 TMSI，并利用保密性算法 F8 加密后

传给移动用户；移动用户利用同样的算法和密钥解密出 TMSI，将去存入 USIM 卡，以便下一次认证之用。

上述协议第五步中 USIM 卡的具体操做过程：收到 RAND 和 ATUN 后，USIM 先利用认证密钥 K 和 RAND 通过匿名性密钥生成函数 $f5$ 生成匿名性密钥 AK，通过异或运算恢复序列值 SQN；然后，利用网络身份认证函数 $f1$ 计算出认证响应 XMAC，并与比较 AUTN 中的 MAC 进行比较。如果 XMAC=MAC,USIM 再查 SQN 是否在有效值范围之内：如果 SQN 有效，USIM 利用函数 $f2$，$f3$,和 $f4$ 计算出 RES,CK 和 IK,并将 RES 传给 VLR 进行移动用户身份认证，同时利用安全参数 K*和收到的 RAND 一起生成 K_Chaos,并将 CK 与 K_Chaos 异或得到的 CK*,然后将 CK*和 IK 传给 UE 准备保密通信。

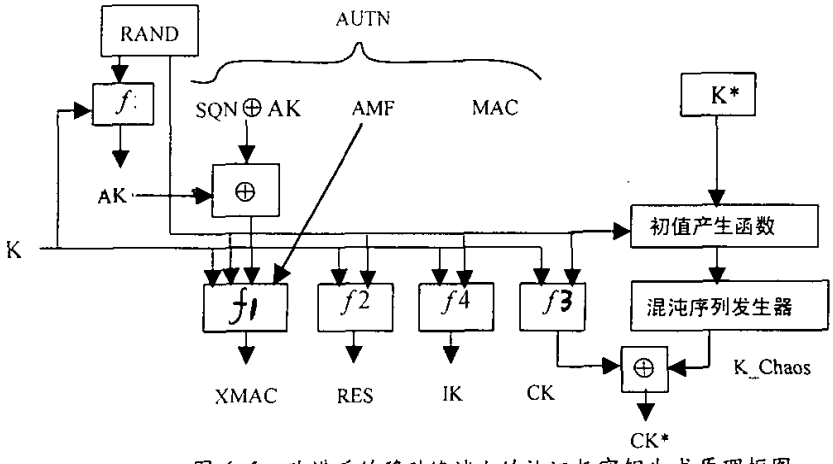


图 6-5　改进后的移动终端上的认证与密钥生成原理框图

Figure 6-5 User Authentication Function in the USIM after Improvement

产生初值函数：首先把 128 位的 RAND 与 128 位的 K*相异或，得到一个新的 128 位数 K1，然后把 K1[0],K1[1],...,K1[9]作为一组相加转换成一位十进制数作为初值的第一位小数。依次类推，可以得到一个有 13 位有效值的在（0，1）之间的小数初值，作为混沌序列发生器的输入。由于每次通信的 RAND 都不一样，所以每次通信的混沌迭代初值都是不一样的。

6.3　安全性和复杂性分析

安全性分析：

在原来的 3G 安全算法中，每一次通信的数据都是采用固定的 128 位初始密钥 CK 来

加密。改进后的安全算法结构中，基于 Shanon 的“一次一密”思想，通过混沌映射迭代产生不同的 K_Chaos,从而可以定时地改变加密初始密钥 CK*，来提高加密数据的抗破译能力。在改进后的安全结构中，通信双方的保密参数由原来的一个认证密钥 K，增加多一个安全参数 K*，如果通过穷举破译的话，初始密钥空间由 2^{128} 变为 2^{256}，再加上数据是通过混沌迭代后产生的序列 K_Chaos 与 CK 异或产生的 CK*作为初始密钥加密，K_Chaos 在定时的改变，所以 CK*也在定时的改变，具有“一次一密”的优点，所以使得通过穷举破译在时间上变得不可能。由于产生的初始密钥 CK*是由现代密码算法函数 F3 和混沌映射迭代所得，它同时具有两种加密算法体制的优点，所以针对某一单独的密码体制分析方法都没法破译被加密过的数据。

结构复杂性分析：

本文是在尽可能不改变 3G 安全通信中原有的认证与密钥生成过程的前提下进行的强保密数据安全研究。在图 6-5 中可以看出，产生 K_Chaos 模块是添加上去的，如果通信的某一方不存在这一模块，则存在的一方 K_Chaos 自动为{0}，保密性密钥 CK*=CK，将按标准方式进行安全通信。在算法实现上也很容易实现。

时间复杂性分析：

由于改进后的加密算法结构要生成新的初始密钥 CK*要经过混沌映射迭代所得的 K_Chaos 与 CK 异或，然后才能把 CK*送到数据加密函数去产生加密密钥后才能加密下一段明文，相对原来的采用固定的初始密钥加密所有通信数据相比必然有一定的时间延迟。在实际应用中可以按实时性和安全性灵活地设定加密多少位明文后才产生新的初始密钥 CK*。表 6-1 给出改进前后的安全算法结构加密数据时间的比较。

表 6-1 改进前后的安全算法结构加密数据时间

数据大小（类型）	固定密钥所需时间（秒）	每 64 位变化密钥所需的时间（秒）	每 640 位变化密钥所需的时间（秒）	每 6400 位变化密钥所需的时间（秒）
49564 字节　（图片）	0. 1090	0. 2970	0. 1250	0. 1100
2290432 字节（图片）	4. 7960	13. 7960	5. 7500	4. 9380
4141558 字节（音频）	8. 7050	24. 953	10. 422	8. 9060
11369455 字节（视频）	23. 5310	68. 4540	28. 406	24. 453
17585013 字节（视频）	36. 4850	105. 7810	44. 015	37. 750

上述表格数据是在 CPU 为 Intel（R）4 2. 66GHz，内存 512M 环境下的 IBM PC 机上测试所得。所采用的加密算法为 3GPP 标准建议中的 $f8$ 算法[20]。从表格数据中可是得出变

密钥比固定密钥有一定的延迟，如果是每加密 64 位比特明文就变换密钥，则延迟时间在 2 倍左右，而当每加密 6400 比特才变换一次密钥时，时间延迟小于 5%。所以我们可以根据安全性和实时性的要求做出合理的选择。

6.4　本章小结

本文针对 3GPP 安全通信中采用的同一初始密钥加密一次通信数据的特点，结合 Shanon 的“一次一密”思想，提出了一种改进的安全通信方法，它把混沌映射生成密钥流代替原分组密码密钥的思想引入到 3G 安全通信中来，对 3G 安全通信中的认证与密钥生成框图进行了改进，以增强通信数据保密性。该方法具有易于同步实现，实时性强、保密性高、运算速度快等明显优点。同理，同样的方法可以用来增强其他方式的通信安全，比如端到端的网络视频电话，视频会议系统等通信，该思想还可以用来提供高保密性的文件数据加密等。

第七章　全文总结与展望

在信息化快速发展的今天，如何确保信息的安全，在理论上已经得到很好的解决，但是由于信息的广泛应用，所传输的信息也已各种各样，所涉及的行业也是各行各业，所传输的信息有机密的也有不可否认的，信息安全的问题就越来越受到人们的关注；但是，由于安全技术的复杂性不是一般人所能掌握的，信息安全问题（特别是电子商务安全）已成为信息化进一步发展的主要障碍。如何让复杂的安全技术简单易用，已成为当前谈论的热点。中间件技术具有屏蔽复杂性等优点，如何将安全技术与中间件技术结合起来，让安全技术真正简单易用、易普及，已成为当前研究的热点问题。由于无线网络传输信息的多样化（如简单的语音信息、复杂的多媒体信息等），网络带宽的有限性和移动终端的局限性，使得无线网络的安全问题比有线网络更加复杂，其安全技术的应用更加困难。所以研究支持移动多媒体通信的安全中间件更具有现实意义。

7.1　工作总结

本文系统地论述了安全中间件技术的研究与进展情况，特别是无线安全中间技术的研究与进展。首先对信息安全所必需的密码学概念和相关技术进行介绍，然后对支持移动多媒体通信的安全中间的体系结构进行设计和分析，其中重点叙述了安全服务层（SSL）和安全服务提供者模块（SSPM）的设计与实现。最后提出了一种混沌序列在3G 安全通信的应用方法。下面对已完成的具体工作进行总结。

1．　完成了对支持移动多媒体通信的安全中间的体系结构设计；

2．　对安全中间件中的通用管理器（CSM）模块和资源信息服务器 RIS 模块大概设计进行了陈述；

3．　完成了安全中间件中的安全服务层中的 3GPP 和 WAP 安全服务的设计与实现，并对给出了 IPSec 的实现框架；

4．　完成了安全服务提供者模块(SSPM)中的密码服务提供者（CSP）和证书服务提供者模块（CertSP）；

5.　提出了一种混沌序列在 3G 安全通信中的应用方法，并给出了他的优点和复杂度。

总的来说，基本完成了支持移动多媒体通信的安全中间中的数据加密解密，数字数字签名、和证书服务等功能，并提出了一种混沌序列在 3G 安全通信中应用方法。

7.2　工作展望

信息安全问题，一直是密码学、信息论等学科领域研究的一个热门问题。如何把复杂的信息安全技术简单易用，将会是今后研究的一个重点问题。研究如何借助中间件技术去简化复杂的信息安全技术，特别是在无线网络中的安全技术，将会更有现实意义。为此还要对今后的工作进行进一步的改进，主要是从一下几个方面进行。

1、 整个安全中间的实现框架应该根据以后的实际应用情况进行进一步的改进；

2、基于移动核心网络的全 IP 化，应继续完成 IPSec 的相关工作，这也是一个重点方向；

参考文献

[1] Patel S.Weakness of North American Wireless Authentication Protocol .IEEE Personal Communication, June 1997.40~44

[2] GSM Security http://jya.com/crack-a5.htm

[3] Biryukow A, Shamir A, Wagner D.Real-time cryptanalysis of GSM's A5/1 on a PC, Proceedings of 7th International Workshop on Fast Software Encryption 2000.New York, April 2000.1~18

[4] Xu S B.Analysis and design Of Authentication Protocol for Portable Communication Systems, Ph.D thesis, Xidian University, March 1998

[5] Xu S B, Jansen C, van Tilborg H. A New Authentication Protocol for Portable Communication Systems. Proceedings CM'99-IFIP Conference on Communications&Multimedia Security, Leuven, Belgium, September 1999.216~230

[6] 3GPP TS 35.202 V6.0.0 (2004-12) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification.

[7] Daemen J, Rijmen V. The design of Rijdael: AES- the Advanced Encryption Standard, Springer, 2002

[8] Advanced Encryption Standard (AES) website, http://csrc.nist.gov/encryption/aes

[9] The SESAME homepage, https://www.cosic.esat.kuleuven.ac.be/sesame

[10] JALAL AL-MUHTADI, DENNIS MICKUNAS, AND ROY CAMPBELL. A lightweight reconfigurable security mechanism for 3G/4G mobile devices, IEEE Wireless Communications, April 2002.

[11] Parimala Rajan, Prabhakar Manchikatla, Suman Srinivasan. Recent Advances in 3G systems. EEL6591 Project Report. Spring 2003

[12] Burkhard Stiller, Christina Class, Marcel Waldvogel, Germano Caronni, Daniel Bauer. A Flexible Middleware for Multimedia Communication: Design, Implementation, and Experience. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 17, NO. 9, SEPTEMBER 1999

[13] 蒋融融，江颉，陈铁明.无线安全中间件的研究与设计.微电子学与计算机.2003, No.11:27-29

[14] 常煜芬，张育平. 中间件技术研究 .计算机应用研究.2001，10：21-23

[15] 冯登国.国内外密码学研究现状及发展趋势.通信学报.Vol.23 No.5, May 2002

[16] 胡磊，王鹏等译.应用密码学手册.电子工业出版社 北京.2005 年 6 月

[17] 刘云毅. 流密码的密码分析设计与应用.博士学位论文.南京大学 2006，01

[18] M. Walker and T. Wright. Security. In F. Hillebrand, editor, GSM and UMTS: The creation of global mobile communication, pages 385-406. John Wiley & Sons, 2002.

[19]LAN/MAN Standard Comitee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999 edition. IEEEE standard 802.11, 1999.

[20] B. Schneier. Appled cryptography: Protocols, algorithms and source code in C. John Wiley & Sons Inc., New York. 2nd edition, 1996.

[20] 3GPP TS 35.201 V6.0.0 (2004-12) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specification

[21] Thomas Johansson. Analysis and Design of Modern Stream Ciphers (Invited Paper). IMA 2003, Lecture Notes in Computer Science. Vo. 2898, page: 66, 2003.

[22] C. J. Mitchell and A. W. Dent, International standards for stream ciphers: A progress report. SASC-The State of the Art of Stream Ciphers, Novotel Brugge Centrum, Brugge, Belgium, 14th-15th October 2004.

[23] Yi Lu, Willi Meier, Serge Vaudenay. The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption. CRYPTO 2005, Lecture Notes in Computer Science, vol. 3621, 2005, 97-117.

[24] W. Diffie and M.E. Hellman. New directions in cryptography. IEEE Trans. Info.Theory, IT-22(6):644-654, 1976

[25] R.L.Rivest, A.Shamir, and L.Adeleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM,21(2):120-126,1978

[26] 王继林，伍前红等译.现代密码学理论与实践. 电子工业出版社：北京.2004,7

[27] W. Diffie and M.E. Hellman. Multiuser cryptographic techniques. In Proceedings of AFIPS 1976 NCC, pages 109-112.AFIPS Press, Montvale

[28] 贺军等译.数字签名.清华大学出版社：北京.2003, 1

[29] National Institute of Standards and Technology. FIPS PUB 186-2. DIGITAL SIGNATURE STANDARD (DSS). 2000 January 27

[30] 何明星,范平志. 新一代私钥加密标准 AES 进展与评述.计算机应用研究.2001 年第 10 期

[31] 余堃，周明天.公共安全中间件核心——公共安全服务.小型微型计算机系统，2003，Vol.24, No.7:1191-1193

[32] 许立. 安全中间件的研究及相关模块的设计和实现.硕士学位论文，电子科技大学 2001.1

[33] Anand Rajan，Matt Wood，David Bowler.Mechanics of the Common SecurityServices Manager (CSSM). Intel Architecture Labs.

[34] 徐胜波，马文平，王新梅.无线通信网中的安全技术.人民邮电出版社，2003

[35] 3GPP TS 33.120. Security Objectives and Principles.1999

[36] 姜楠，王健.移动网络安全技术与应用.电子工业出版社 北京，2004

[37] 3GPP TS 35.205 V6.0.0 (2004-12)3rd Generation Partnership Project; Technical Specification Group Services and System Aspects;3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*.

[38] 3G TS 33. 105 V3.4.0 (2000-07) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Cryptographic Algorithm Requirements. (Release 1999)

[39] 3GPP TS 35.202， V6.0.0 (2004-12) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification.

[40] 3G TS 33.102， V6.3.0 (2004-12) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security Architecture.

[41] 3GPP TS 33.220 V7.0.0 (2005-06) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Generic Authentication Architecture (GAA); Generic bootstrapping architecture.

[42] WAP Forum: WAP Wireless Application Protocol WAP2.0 Technical White Paper. January 2002.

[43] 姜楠，王健.移动网络安全技术与应用.电子工业出版社 北京，2004

[44] WAP Forum: WAP TLS Profile and Tunneling Specification. 11-April-2001

[45] WAP Forum: WAP Transport Layer End-to-End Security Specification.Version 2.0 June 28, 2001

[46] WAP Forum: WAP WMLScript Crypto API Library Specification. 20-June-2001.

[47] WAP Forum: Wireless Transport Layer Security Specification. Version2.0.April 6, 2001

[48] WAP Forum: WAP Public Key Infrastructure Specification. 24-Apr-2001

[49] WAP Forum: WAP Certificate and CRL Profiles Specification. 22-May-2001

[50] RSA Laboratories, PKCS #1: RSA Cryptography Specifications Version 2.0, Sep 1998

[51] IETF RFC 2459: Internet X509 Public Key Infrastructure Certificate and CRL Profile

[52] S.Kent and R.Atkinson.Security Architecture for the Internet Protocol. The Internet Engineering Task Force Request For Comments (IETF RFC) 2401, November 1998. Available at www.ietf.org/rfc/rfc2401.txt

[53] S.Kent and R.Atkinson.IP Authentication Header (AH). The Internet Engineering Task Force Request For Comments (IETF RFC) 2042, November 1998.Availaber at www.ietf.org/rfc/rfc2402.txt

[54] S.Kent and R.Atkinson. IP Encapsulating Security Payload (ESP), The Internet Engineering Task Force Request For Comments (IETF RFC) 2406, November 1998. Available at www.ietf.org/rfc/rfc2406.txt

[55] D.Harkins and D.Carrel. The Internet key exchange protocol (IKE).The Internet Engineering Task Force Request for Comments (IETF RFC) 2409, November 1998

[56] C.Kaufman. Internet Key Exchange Protocol (IKEv2). The Internet Engineering Task Force: INTERNET-DRAFT, draft-ietf-ipsec-ikev2-03.txt, October 2002

[57] 徐佳,荆继武.实现 IPSec 的一种方案.计算机工程.2002 年 1 月,Vol.28, No.1

[58] R.Thayer, N.Doraswamy and R.Glenn. IP Security Document Roadmap. The Internet Engineering Task Force Request For Comments (IETF RFC) 2411, November 1998. Available at www.ietf.org/rfc/rfc2411.txt

[59] Knuth D E. The art of computer programming.Zeroth printing (revision 15), 15 February 2004

[60] 关新平，范正平等．混沌控制及其在保密通信中的应用．北京：国防工业出版社．2002，10.

[61] 孙枫，秦红磊，徐耀群，郝燕玲．基于混沌的分组密码置换网络的设计．中国工程科学．2000，9

[62] 邓绍江,肖迪，涂凤华． 基于 Logistic 映射混沌加密算法的设计与实现．重庆大学学报．2004,4． 61-63

[63] 左涛， 蒋国平．基于离散混沌序列的加密算法研究．南京邮电学院硕士论文．2004，3．40-57

致 谢

特别感谢我的导师覃团发教授，感谢导师在我攻读硕士期间，从课程学习到论文选题，从平时撰写论文到最后的学位论文，在各方面导师都给予了我耐心细致的指导和帮助。导师严谨的治学态度、求真务实的科研作风、勤奋工作的敬业精神和渊博的知识使我终身受益。覃老师在生活上对我的关心，使我终身难忘。

特别感谢师兄刘运毅博士对本论文的修改提出的宝贵意见。

同时，对陈海强老师、姚海涛老师、唐秋玲老师、唐天兵老师、黎英云老师、陈俊江老师、黄旭方老师、崔嵩老师表示感谢。感谢他们在我读研期间给予我学业、生活上的关怀和帮助。

在此，也要感谢同时也要感谢在通信实验室一起学习过的师兄师弟、师姐师妹们，他们是：王海霞、胡永乐、陈俊江、区骋、陈琳、唐振华、李淑明、黄海星、陈健玲、王雨生、秦德兴、金莲芳、黄增峰、王国忠、帅勤、刘淑华、常沛、俞静、陈语中、沙丽娜，田卉、梁琳、翟玉兰、苏敏、陈义钦、朱朝阳、谭俊光、蔡奕奕、黄志、黎相成、黄宣钧，感谢他们对我的关怀和帮助，让我感到了集体的温暖。

感谢我的家人，他们支持和鼓舞使我能安心于学习，顺利完成论文。

感谢我的研究生同学，特别是董明刚、朱正平、陈望明、冯凌凌，他们陪伴我度过了愉快的三年时光。

特别感谢许业军、牟礼林、梁奕艳、梁开挺、许业文、梁飞、薛忠、杨业挺、罗善超、莫鹏、田晓冰、陈燕丽、庞家惠、王福广等同学，在我读研期间对我的关心和帮助。

在论文完成之际，谨向所有关心、支持、帮助过我的人们表示深深的谢意。

攻读硕士学位期间发表的学术论文

1. 万海斌，覃团发，常沛，陈语中. 支持移动多媒体通信的安全中间件. 无线电工程. 2006年，第05-06期
2. 万海斌，覃团发，常沛，陈语中. 混沌序列在3G安全通信中的应用研究. Journal of Communication and Computer. Oct, 2005. P67~P71
3. 常沛，覃团发，万海斌，陈语中. 移动多媒体中间件中的位置管理模块. 第十届全国青年通信学术会议论文集. 2005年8月. P53~P56
4. 刘淑华，胡强，覃团发，万海斌. 语音增强算法的研究. 第十届全国青年通信学术会议论文集. 2005年8月. P282~P287
5. 覃团发，陈语中，常沛，万海斌. 支持移动多媒体通信中间件. 广西大学学报. 2005, 3
6. 陈语中，覃团发，常沛，万海斌. 支持移动多媒体通信中间件的WAP网关. 无线电工程，第36卷，总第201期，2006.2