代　号　10701　　学　号　0820421277
分类号　TP309.7　　密　级　公开

西安电子科技大学

硕士学位论文

题（中、英文）目　基于 AES 和 ECC 的加密体制研究及硬件实现

Research and Hardware Implementation of Cryptosystem Based on the AES and ECC

作者姓名　周杰　　指导教师姓名、职务　牛海军　教授

学科门类　工学　　学科、专业　计算机系统结构

提交论文日期　二〇一一年一月

西安电子科技大学
学位论文创新性声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：周杰　　日期 2011.2.20

西安电子科技大学
关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名单位为西安电子科技大学。

本人签名：周杰　　日期 2011.2.20

导师签名：牛海军　　日期 2011.2.20

摘要

安全问题在计算机网络中扮演了非常重要的角色。随着互联网越来越被大众所接受与使用，各种商业和社会业务开始转化为电子形式在网上进行，使用各种各样的加密算法，可以保证这些活动的安全性。而目前加密算法的实现正逐渐由软件实现向硬件转变，硬件实现加密的产品越来越受到重视。所以研究分析当前流行的加密体制及其硬件实现对我国的信息安全有很重要的意义。

本文通过对现有数据加密体制的分析，给出了一种将 AES(Advanced Encryption Standard)和 ECC(Elliptic Curve Cryptography)加密体制相结合的混合数据加密体制，从而更为高效地实现了网络通信系统中的信息加密、数字签名和身份验证，解决了密码体制中速度和安全性不能兼顾的问题。文章首先在对目前信息安全的现状和数据加密技术的基本概念及数学模型作了简单综述的基础上，研究分析了各种数据加密技术的思想、加解密流程，给出了对称密码、公钥密码体系的数学背景及其特点；其次，针对数据加密方法的特点，提出了基于 AES 与 ECC 的混合密码体制，并就这两种算法的数学原理、算法实现、安全性能等方面进行了详细的讨论；同时还给出基于 AES 与 ECC 的混合密码体制的工作原理。接着，针对 AES 和 ECC 加密算法硬件实现的设计做出了详细的分析，采用高速的 DSP 芯片 TMS320C6713 为平台，时下流行的 USB2.0 接口芯片 EZ-USB FX2 CY7C68013 与主机进行通信。着重阐述了平台各个部分的设计，给出了设计结果。最后对 AES 和 ECC 的混合密码体制运用于某研究所的保密通信服务系统中的实现方案进行了论述。从而使 ECC 密码体制密钥短，计算开销少，带宽要求低、运算速度快的优点和 AES 算法的安全性能高的优势充分的发挥，有效的解决了消息的机密性、身份鉴别、完整性和不可否认性。

关键词：ECC　AES　混合加密体制　DSP　保密通信

Abstract

Security issues are playing an important role in computer networks. As the Internet becomes more and more accessible to the public, many kinds of commercial and social transactions can be performed electronically through the network, various cryptographic algorithms are needed in order to insure the security of these activities. In addition, now the implementation of cryptographic algorithm is on the transition from software to hardware and the cryptoproduct is paid more and more attention. Therefore, it is very important to study and analyze the popular cryptographic algorithm and it is hardware implementation for the sake of nation's information security.

The present data encryption systems are analyzed in the dissertation to propose the mixed cryptosystem based on the AES (Advanced Encryption Standard) and ECC (Elliptic Curve Cryptography), which will actualize encryption of data, digital signature and identification with high efficiency in Internet communication, and solve the problem of antinomy of speed and security in cryptosystem. First, this dissertation researches and analyzes on the thought, the method and ecology and field of application of many kinds of data encryption technology show the mathematical description to symmetry cryptogram and public key cryptogram and its characteristics based on briefly on the current situation of the in formation security and the basic notion as well as mathematical model of data encryption technology; Secondly, the mixed cryptosystem based on AES and ECC is build forward in according with the characteristic of the encryption method of the data the detailed discussion is hold on two mathematical principles, algorithm realization safe performance, etc. At the same time its work flow is also given; thirdly, AES and ECC encryption algorithm for hardware implementation of the design to make a detailed analysis, using high-speed DSP chip TMS320C6713 platform, popular USB2.0 interface chip EZ-USB FX2 CY7C68013 to communicate with the host. Focuses on the design of various parts of the platform, gives the design results; finally, the plan which is applied to the secure communication service system of a research institute is elaborated. The advantages of the ECC cryptosystem such as short-key, low expenses, low request of the band, the quick speed of calculation and the high superiority of AES algorithm effectively solute the problem of the confidentiality, the personal identification, the integrity and the Non-repudiation of information.

Keyword: ECC AES Mixed Cryptosystem DSP Secure Communication

目录

第一章 绪论

1.1 研究背景

当今世界计算机和通讯技术迅猛发展，信息产业已经成为现代社会的支柱产业。单机网络化、局域网络互联化已成必然之势。随着 Internet 网在国内的普及和发展，越来越多的企业用户和个人用户进入 Internet 网。因此 Internet 网上的数据安全问题已经成为各界广泛关注的焦点。伴随着网络的普及，信息安全日益成为影响网络的重要问题，而网络所具有的开放性、自由性在增加应用自由度的同时，对安全提出了更高的要求。如何使信息网络系统不受黑客和工业间谍的入侵，已成为企事业单位信息化健康发展所要考虑的重要问题之一。

由于政府、商业部门及个人的数据以电子的形式存储，信息安全便成为非常重要的方面。电子媒体更有利于数据的使用：存储方便，传输迅速，访问数据库便利。商业领域已经认识到信息是他们最有价值的资本。但是，随着电子的革命，电子信息面临着更大更新的、潜在的安全威胁。相对于纸介质上的信息，电子信息比纸介质上的信息更容易被远程截取或篡改。

简单地说，信息安全即阻止未经授权使用电子数据——不论这个未经授权的使用是否泄露、改变或替换了有关数据。信息安全应包含下面三个内容：

保密性——对未经授权的人或团体保密。

完整性——确保数据的真实性。

有效性——经过安全处理后，信息仍然有效[1]。

许多措施已经用于提供安全服务，同时也没有一种措施可以保证绝对安全。在各种各样的措施中，加密系统提供的安全级别较高，适应性较好。广泛地说，一个加密系统即将电子数据转换为一种改变了的形式。信息的拥有者要确保信息在改变了的形式下的安全。信息安全就是要确保不能被未察觉地修改数据，或者被未授权组织加密。加密系统的关键在于使用一个密钥来完成转化过程。这个密钥本身就是一个电子字符串。当然，信息安全不仅要使用加密系统转化数据，而且密钥的拥有者必须保护好密钥本身。可以毫不怀疑地说，一个加密系统的正确使用可以为我们今天的电子信息提供较高级别的安全性。加密系统提供的服务有三种目标的信息安全：保密性、完整性及有效性。为了进一步说明加密系统，保密性和完整性又被细分为下面五类，它们能够用于建造一个加密系统块：

保密性——对未经授权的组织或个人保密。

使用者鉴定——确保实时传送数据的团体正是要传送数据的团体。

数据来源鉴定——确保消息的来源合法。

数据完整性——确保数据未被未授权使用者篡改。

不可否认性——参与传送的实体含有一个绑定，使得实体不能被否认。既实体的接收者能够向一个中立的第三方证明传送者为真正的传送者。

目前我国对计算机网络安全产品的认证研究刚开始起步，尚没有对计算机安全发布权威性的标准方案。新刑法中关于计算机网络犯罪的条款也没有解决法律上的问题。不管在设计网络安全方案时，还是在确认事故、攻击、入侵等级时都显得无所适从。对企业来说，等待国家去研究发布相应的标准是不恰当的。企业应当尽快组织自己的 IT 部门或购买其它公司的相应服务来确立自己网络的安全策略，并且定期地检查实际情况与安全策略之间的差距。

随着信息产业现代化进程的加快，网上的各种应用也随之发展起来。如:电子银行、网上定票、各种费用查询与支付、人才中介、电子报税、工商信息、网上书店、预定饭店、网上展览会、展销会、招商洽谈会、实时交通信息、网上图书馆、网上博物馆、远程教学、远程医疗、电视会议等等。众多的网上应用带来了网络安全的问题。网络安全问题一般可分为网络系统安全和数据安全两类。网络系统安全问题是指网络系统遭到未经授权的非法攻击、存取或破坏；数据安全问题则指机要、敏感数据被窃取并非法复制、使用等。认证就是其中的主要问题。

在认证中应用最广泛的是口令，口令具有共享秘密的属性。例如，要使服务器操作系统识别要求服务的用户，那么用户必须把他的用户名和口令发送至服务器。服务器就将它与数据库里的用户名和口令进行比较，如果相符，就通过了认证，可以对其服务。这个口令其实就是由服务器和用户共享。更保密的认证可以是多种方法组合而成，在安全方面最薄弱的一环是规程分析仪的窃听，如果口令以明码传输，接入到网上的规程分析仪就会在用户输入帐户和口令时将它记录下来，任何人只要获得这些信息就可以登录服务器工作。“黑客”往往采取多种方法来获取用户口令。如登录界面(Shell Scripts)攻击法。它是在被攻击主机上启动一个可执行程序，显示一个伪造的登录界面。当用户在这个伪装的界面上键入登录信息(用户名、密码等)后，该程序将用户输入的信息传送到攻击者主机，然后关闭界面给出提示信息“系统故障”，要求用户重新登录。此后，才会出现真正的登录界面。再比方说诱入法：“黑客”编写一种看起来像合法的程序，放到商家的主页，诱导用户下载。当一个用户下载软件时，“黑客”的这个软件与用户的软件一起下载到用户的机器上。该软件会监视用户的电脑操作，它记录着用户输入的每个口令，然后把它们发送给“黑客”的 Internet 信箱。因此对口令和传输数据进行加密是非常必要的，这一切也加速了加密、解密、数字签名技术的产生和迅速发展。

网络安全是计算机安全在网络环境下的扩展和延伸，主要包括用户身份验证、

访问控制、数据完整性、数据加密、防抵赖和审计追踪等安全要求。

数据加密技术是对信息进行重新编码，从而达到隐藏信息内容，使非法用户无法获得信息真实内容的一种技术手段。网络中的数据加密则是通过对网络中传输的信息进行数据加密，满足网络安全中数据加密、数据完整性等要求。可见，数据加密技术是实现网络安全的关键技术。

1.2 算法发展现状

1.2.1 对称算法现状

对称算法(Symmetric Algorithm)，有时又称为传统密码算法。也就是加密密钥能够从解密密钥中推算出来，同时解密密钥也可以从加密密钥中推算出来。而在大多数的对称算法中，加密密钥和解密密钥是相同的。所以也称这种加密算法为秘密密钥算法或单密钥算法。它要求发送方和接收方在安全通信之前，约定一个密钥。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可以对发送或接收的消息解密，所以密钥的保密性对通信的安全性至关重要。对称加密的优点在于算法实现的效率高、速度快。

对称加密的缺点在于：第一，密钥量问题。在单钥密码系统中，每对通信就需要一对密钥，当用户增加时，必然会带来密钥量的成倍增长，因此在网络通信中，大量密钥的产生、存放和分配将是一个难以解决的问题。第二，密钥分发问题。单钥密码系统中，加密的安全性完全依赖于对密钥的保护，但是由于通信双方使用的是相同的密钥，人们又不得不相互交流密钥，所以为了保证安全，人们必须使用一些另外的安全信道来分发密钥，例如用专门的信使来传送密钥。这种做法的代价是非常大的，甚至可以说是很不现实的，尤其在计算机网络环境下，人们使用网络传送加密的文件，却需要另外的安全信道来分发密钥，很明显，这需要新的解决方法[2]。常用的对称加密算法有数据加密标准(DES,Data Encryption Standard)、数据加密算法(DEA,Data Encrytion Algorithm)和高级加密标准(AES,Advanced Encryption Standard)等。DES 算法由 IBM 公司开发，并被美国国家标准局于 1977 年 2 月采纳作为“非密级”应用的一个标准，此后，DES 成为全世界使用最广泛的加密标准。DES 算法加密时把明文以 64bit 为单位分成块，采用美国国家安全局精心设计的 8 个 S 盒(S:Substitution)和 P 置换(P:Permutation)，经过 16 轮迭代，最终产生 64 比特密文，每轮迭代使用的 48 比特子密钥由原始的 56 比特产生。DES 的加密与解密的密钥和流程完全相同，区别仅仅是加密与解密使用的子密钥序列的施加顺序正好相反。DES 算法在历史上曾发挥重要作用，但也存在以下问题：

1．DES 密钥空间的规模 256 对实际安全而言太小；

2．DES 的密钥存在弱密钥、半弱密钥和互补密钥；

3．DES 里的所有计算，除去 S 盒，全是线性的。

S 盒的设计对密码算法的安全性至关重要。然而，美国国家安全局并没有公布 S 盒的设计原则，因此，有人怀疑 S 盒里隐藏了“陷门(trapdoors)”，如果是这样，美国国家安全局就能轻易的解密消息。此外，由于 DES 的密钥空间小，针对 DES 算法进行穷举攻击就可以取得成功。在 1998 年 7 月，电子前沿基金会(EFF)使用一台 25 万美元的电脑在 56 小时内破译了 DES 密钥。1999 年 l 月 RSA 数据安全会议期间，EFF 通过遍布全世界的 10 万台计算机的协同工作，用 22 小时巧分钟就宣告破解了一个 DES 的密钥。为了增强 DES 算法的安全性，密码设计者又提出了基于 DES 的 Triple2DES、独立子密钥方法和推广的 GDES 算法等。这些改进最终的作用不大，有些还削弱了 DES 的安全性。总之，DES 需要新的有效的加密标准来代替。

美国国家标准和技术研究所(NIST, National Institute of Standards and Technology)于 1997 年 1 月开始了遴选 DES 替代者——高级加密标准 AES(Advanced Encryption Standard)的工作。其目的是为了确定一个非保密的、全球免费使用的分组密码算法，用于保护下一世纪政府的敏感信息，并希望成为秘密和公开部门的数据加密标准。

AES 的确立过程简介如下：

1．1997 年 9 月 12 日，NIST 在联邦登记处公布了征集 AES 候选算法的通告。并对候选者提出以下基本要求：

(1)比 Triple2DES 快，且至少和 Triple2DES 一样安全；

(2)应当具有 128 比特分组长度和 128/192/256 比特密钥长度；

(3)具有较大的灵活性。

2．1998 年 8 月 20 日，NIST 召开了第一次候选大会并公布了 12 个国家的 15 个候选算法。

3．1999 年 3 月 22 日，NIST 召开了第二次 AES 候选会议，从中选出 5 个候选算法:MARS(IBM)、RC6(MIT)、Serpent(英、以、美)、Twofish(美)和 Rijndael(比利时)。

4． 2000 年 10 月 2 日，NIST 宣布比利时的密码学家 Joan Daemen 和 Vincent Rijimen 设计的“Rijndael 算法”最终获胜。

1.2.2 公钥密码算法现状

1949 年，美国数学家香农(Shannon)发表了“Communication Theory of Secrecy

Systems(保密系统的通信理论)”一文。该文采用信息论的观点对通信保密问题作出全面的论述，阐明了密码系统、完善保密性、纯密码、理论保密性和实际保密性等概念，建立了保密通信的数学理论，大大深化了人们对密码学的理解[3]。该文章的发表宣告了科学的密码学理论时代的到来，将密码学的研究纳入到科学的轨道。1973 年 5 月 15 日，NIST 在联邦记录中公开征集密码体制，该举措导致了 DES 的出现，它曾成为世界上最广泛使用的密码体制。随着分布式计算和并行处理技术的发展，56bits 的 DES 已难以胜任公用数据标准算法。为此，1997 年，NIST 开始征集 AES 算法，以此作为 DES 的替代品。

1976 年，美国密码学专家 Whitfield Diffie 和 Martin Hellman 发表了“New Directions in Cryptography(密码学的新方向)”一文，提出了非对称密码体制，冲破了长久以来一直沿用的对称密码体制，开拓了密码学的新方向，使密码学发生了一场变革[4]。这篇划时代的文章，奠定了非对称密码体制的基础，被认为“密码学上的一个里程碑” [5]。自从非对称密码体制的概念被提出以来，相继提出了许多公钥密码方案。

目前，只有 3 类公钥密码体制被认为是安全有效的：

1．基于大整数因子分解问题(IFP, Integer Factorization Problem)。典型代表：RSA(Rivest, Shamir, Adleman)；

2．2 基于有限域离散对数问题(DLP, Discrete Logarithm Problem)。典型代表：数字签名算法(DSA, Digital Signature Algorithm)；

3．基于有限域椭圆离散对数问题(ECDLP, Elliptic Curve Discrete Logarithm Problem)。典型代表：椭圆曲线密码体制(ECC, Elliptic curve cryptosystem)。

随着计算机速度的不断提升，分解大整数的能力也日益增强，为保证 RSA 体制的安全性总是要增加密钥位数。目前 768bits 的密钥 RSA 体制已不安全，一般建议使用 1024bits 的密钥，预计要保证 20 年的安全性就要选择 2048bits 的密钥，但密钥长度的增大加大了实现的难度，势必会造成效率的降低和存储空间的浪费[6]。为解决上述问题而 ECC 具有密钥短、计算量小、速度快和灵活性好等优势，有取之不尽的椭圆曲线可用于构造椭圆曲线有理点群，且不存在计算 ECDLP 的亚指数算法。目前技术下只需要 160bits 的密钥即可保证其安全性，因而倍受国际关注。国际上已制定椭圆曲线公钥密码标准 IEEEPl363。

相对于 RSA 等密码体制来说，椭圆曲线密码体制是比较新的技术，在椭圆曲线密码体制中，首要问题就是安全椭圆曲线的选取问题，如果选取的椭圆曲线本身是不安全的，那么基于该椭圆曲线的任何方案都是不安全的。如何选取基点也是构造椭圆曲线密码体制的关键要素。所以在本章介绍椭圆曲线密码体制的常用攻击方法，在大素数域上如何选取安全的椭圆曲线和基点，并仿真椭圆曲线部分运算，给出椭圆曲线密码体制的优点。有限域上的椭圆曲线密码体制的安全性依

赖于有限域上的椭圆曲线点群中的离散对数问题的难解性。目前，解决这个数学困难问题的最有效的算法仍然需要完全指数时间。同时由于 RSA 密码体制中所要求的密钥长度越来越大，导致工程实现变得越来越困难，人们发现椭圆曲线密码体制是克服此困难的一个有效的方案，因此椭圆曲线密码体制成为了一个研究热点。

Schoof 首先于 1985 年提出计算椭圆曲线有理点个数的算法，Atikin 和 Elkies 于 1989 年到 1992 年之间，对其作出了重大改进，随后经过 Couvergnes、Morain、Lercier 等人完善，到 1995 年人们已经能较容易地计算出满足密码要求的椭圆曲线有理点的个数，从而解决了椭圆曲线的选取问题和对任意椭圆曲线上有理点个数的计算问题，为椭圆曲线密码系统的实现铺平了道路。现在已有许多的厂商已经或正在开发基于椭圆曲线加解密和数字签名的产品。加拿大 CertiCom 公司是国际上最著名的ECC密码技术公司，已授权多家企业使用该公司的ECC密码技术产品。

2000 年武汉已经开发成功首套椭圆曲线加密软件，采取与目前国际上通行的 RSA 加密法迥异的一种数论计算方法，其核心技术指标达到国际网络安全性认证标准。2003 年 5 月 12 日中国颁布的无线局域网国家标准 GB15629.n 中，涉及到证书的签名采用的就是椭圆曲线 ECC 算法。2005 年，清华大学微电子研究所的白国强等人设计完成椭圆曲线密码芯片 THECC/233-100，是国内第一块 ECC 芯片。该芯片具有完全自主知识产权。数字签名算法采用了 IEEE1363 标准中的算法。经电路板验证，在 100MHZ 的工作频率下芯片工作稳定，每秒可以连续完成数字签名 4000 次。椭圆曲线密码体制和其它公钥密码体制相比具有如下优点：所需的计算负载小、存储要求低、所占带宽窄，这些问题正是网络传输系统所要考虑的。随着网络的日益普及、椭圆曲线密码理论的日益成熟，椭圆曲线密码体系的应用日益成为各国学者研究的热点问题，引起了世界标准组织及密码学界越来越广泛的关注。

密码技术，特别是加密技术，是信息安全技术中的核心技术，国家关键部门中不可能引进或采用别国的加密技术，只能靠自主开发。因此，我国必须要有自己的算法、标准和体系，来应对未来的挑战。

1.3 硬件发展现状

由于这两种加密体制要进行大量的运算，所以选用数字信号处理 (DSP,Digital Signal Processing)芯片，也称为数字信号处理器，是一种特别适合于进行数字信号处理运算的微处理器，其主要应用是实时快速地实现各种数字信号处理算法。与通用微处理器相比，DSP芯片的其他通用功能相对较弱一些。但是，近年来新推出的DSP芯片已经将通用微处理器的一些功能集成在芯片中，DSP芯片已经可以实现

普通微处理器的功能。

1．普遍采用了数据总线和程序总线分离的改进哈佛结构，比传统处理器的冯·诺依曼结构有更高的指令执行速度；

2．DSP大多采用了流水线技术。计算机在执行一条指令时，总要经过取指令、译码、访问数据、执行等几个步骤，需要若十个指令周期完成。流水线技术是将各指令的执行时间重叠起来。综合起来看，使得每条指令的最终执行时间是在单个指令周期内完成的；

3．片内有多条总线可以同时进行取指令和多个数据存取操作，并且有辅助寄存器用于寻址，它们可以在寻址访问前或后自动修改内容，以指向下一个要访问的地址；

4．针对滤波、相关、矩阵运算等需要大量乘累加运算的特点，DSP硬件的乘累加结构使得它可以在一个指令周期完成一次乘法和一次加法运算；

5．具有中断处理器和定时控制器，可以方便地构成一个单芯片系统；

6．多数DSP带有直接内存访问(DMA, Direct Memory Access)通道控制器和同步串行接口，配合片内多总线结构，可以大大提高数据块传输速度；

7．具有软、硬件等待功能，能与各种高低速存储器接口。

最成功的DSP芯片当数美国德州仪器(TI)公司的一系列产品。TI公司于1982年推出其第一代产品TMS32010。公司发展至今，已经成为全球领先的数字信号处理与模拟技术半导体供应商，主要推出了2000系列、5000系列、6000系列等产品，不同系列的DSP具有不同的适用场合：2000系列DSP主要用于数字电机控制、工业自动化、电力转换系统等工业控制领域；5000系列主要用于有线和无线通信领域；6000系列DSP主要应用于数字语言处理、数字图像处理等高速数字信号处理领域[7]。现今DSP器件在高速度、可编程、小型化、低功耗等方面都有了长足的发展，单片DSP芯片最快每秒可完成16亿次(1600MIPS，每秒1600兆次指令)的运算，生产DSP器件的公司也不断壮大。

目前，市场占有率前四名依次为：Texas Instruments、Lucent、AnalogDevice、Motorola涉足这一领域的公司还有AT&T、Fujitsu、Harris、IDT、NEC、INMOS、OKI、Samsung。

由于各DSP厂家的竞争及生产工艺的不断提高，使得DSP器件的价格不断下降，且性能不断提高，这些年来基本上按照这样一种规律发展：约每18个月性能提高一倍，而价格下降一半，这就是著名的摩尔定律。DSP器件应用面从起初的局限于军工、航空航天等领域，扩展到今天的诸多电子行业及消费类电子产品中。并且随着DSP芯片性能价格比的不断提高，DSP芯片将会在更多的领域内得到更为广泛的应用。

1.4 本文的主要工作及结构安排

本文在广泛研究了国内外公钥密码技术基础之上，提出了基于 ECC 和 AES 加密技术的加密板的研究。本文首先在对目前信息安全的现状和数据加密技术的基本概念及数学模型作了简单综述的基础上，研究分析了各种数据加密技术的思想、方法和应用领域，给出了对称密码、公钥密码体系的数学描述及其特点；其次，针对数据加密方法的特点，提出了基于 AES 与 ECC 的密码体制，并就这两种算法的数学原理、算法实现、安全性能等方面进行了详细的讨论；并给出基于 AES 与 ECC 的密码体制的工作原理；随后将 AES 和 ECC 的密码体制得到了可行性验证；最后提出了基于 AES 与 ECC 混合密码体制。从而使公钥算法密钥易管理，密钥短，计算开销少，带宽要求低，运算速度快的优点和对称算法的加、解密速度快的优势充分的发挥，构造出安全、高效的密码系统，解决了电子邮件的机密性、身份鉴别、完整性和不可否认性。

第二章 USB通信协议

2.1 USB基本原理

通用串行总线(USB,Universal Serial Bus)通过总线支持计算机与外设之间的数据交互，外设之间共享 USB 带宽。

2.1.1 USB的产生

USB是一种应用在计算机领域的新型接口技术，最早是由Compaq、Intel、Microsoft等多家公司于1994年11月共同提出的，其目的是用USB来取代PC现有的各种外围接，使外围设备(简称外设)的连接具有单一化、即插即用、热插拔等特点。它的出现大大简化了PC机和外设的连接过程，使PC机接口的扩展变得更加容易。可以说，USB是计算机外设连接技术的重大变化。

2.1.2 USB系统结构

对开发人员来说，这种连接可被分为三个逻辑层：功能层、USB设备层和USB总线接口层，如图2.1所示。不同层次的实现者对于USB的有不同要求，这使得我们得从不同层次来研究USB系统。这种分层结构简化了USB通信机制，有利于理解主机的软硬件和USB设备之间的通信关系，同时也使不同层次的实现者只关心USB的相关层次，不必掌握全部细节。

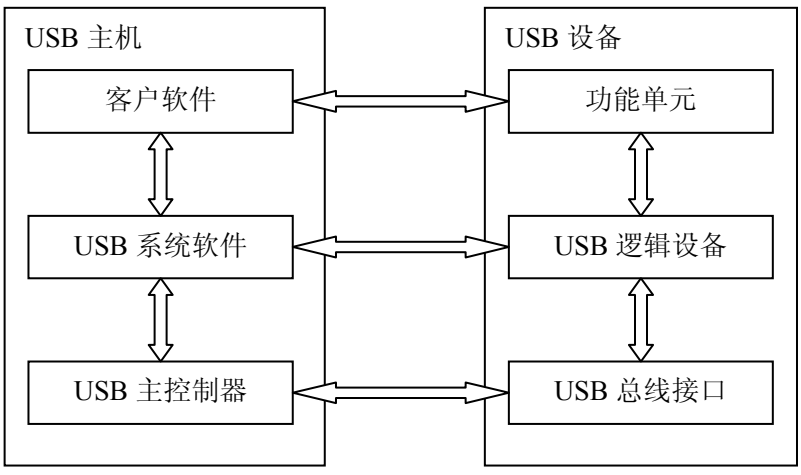

图 2.1 USB的通信模型

USB串口总线连接设备时必须使用集线器(Hub)，类似于在局域网中的连接。USB总线接口包括USB主控制器和根集线器，其中USB主控制器负责处理主机与设备之间的电气和协议层的互联，根集线器提供USB设备的连接点。USB系统使用USB主控制器来管理主机和USB设备之间的数据传输，而从USB的拓扑结构来看有四个部分：

主机和USB设备是USB系统的主要部分；

物理拓扑描述了USB各设备在物理上通过星型结构连接到主机上，表明了各部分之间的相互连接关系，如图2.2所示；

逻辑拓扑为各部分的作用和功能；

客户软件与功能软件的关系说明客户软件与其相关的功能接口。

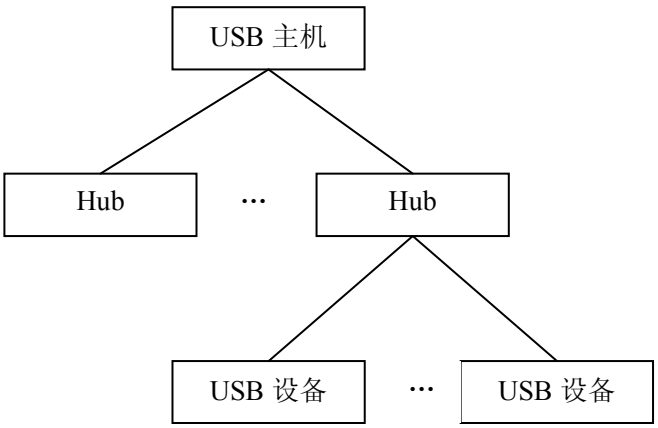

图2.2 USB物理拓扑模型

从终端用户看来，USB系统就是一台主机和若干附属的USB设备之间的通信功能，如图2.3的三个部分：

图2.3 USB简单模型

1． USB主机

在USB系统中只允许有一个主机，它可被分为三个不同的功能模块：客户软件、USB系统软件和USB总线接口。

(1)客户软件：客户软件负责和USB设备的功能单元进行通信，以实现其特定功能。它不能直接访问USB设备，其与功能单元间的通信必须经过USB系统软件和USB总线接口模块才能够实现。它一般包括USB设备驱动程序和界面应用程序两部分。

(2)USB系统软件：负责和USB逻辑设备进行配置通信，并管理客户软件启动的数据传输。它一般包括USB总线驱动程序、USB主控制器驱动程序和非USB主机

软件三部分。

(3)USB总线接口：包括主控制器和根集线器两部分。根集线器为USB系统提供连接起点，主控制器负责完成主机和USB设备之间数据的实际传输。

2．USB设备

在终端用户看来，USB设备为主机提供了多种多样的附加功能，如扩展USB端口、传输文件等，但对USB主机来说，它与所有USB设备的接口都是一致的。一个USB设备由三个功能模块组成：USB总线接口、USB逻辑设备和功能单元。

USB总线接口是USB设备中的串行接口引擎(SIE,Serial Interface Engine)，USB逻辑设备被USB系统软件看作是一个端点的集合，功能单元被客户软件看作是一个接口的集合。其中，S匝、端点和接口都是USB设备的组成单元。

3．USB互连

(1)设备架构

为了正确描述USB设备的特性，USB提出了设备构架的概念，如图2.4所示。设备构架认为USB设备是由一些配置、接口和端点组成的，即一个USB设备可以含有一个或多个配置，在每个配置中可含有一个或多个接口，在每个接口中可含有若干个端点。其中，配置和接口是对USB设备功能的抽象，实际的数据传输由端点来完成。在使用USB设备前，必须指明其采用哪个配置和接口。USB设备使用各种描述符来说明其设备构架，包括设备描述符、配置描述符、接口描述符、端点描述符和字符串描述符，它们通常被保存在USB接口芯片的固件中。

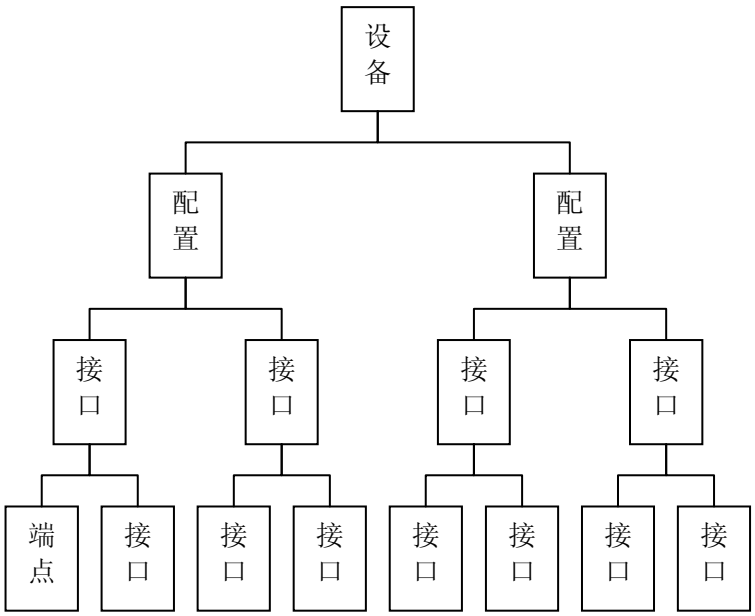

图2.4 USB设备架构

(2)管道

在USB系统的分层结构中，可以认为数据传输是在主机软件(USB系统软件或客户软件)和USB设备的各个端点之间直接进行的，它们之间的连接被称为管道。管道用于客户软件和USB设备的功能单元之间进行通信，它们是在USB设备的配置过程中建立的，在配置完成后，客户软件就可以使用它们了。管道是对主机和USB设备间通信流的抽象，它表示主机的数据缓冲区和USB设备的端点之间存在着逻辑数据传输，而实际的数据传输是由USB总线接口层来完成的。

管道和USB设备中的端点一一对应。一个USB设备含有多少个端点，其和主机进行通信时就可使用多少条管道，且端点的类型决定了管道中数据的传输类型，如中断端点对应中断管道，且该管道只能进行中断传输。不论存在着多少条管道，在各个管道中进行的数据传输都是相互独立的。

由0号端点构成的管道称为缺省管道，它是双向的，既支持IN传输也支持OUT传输。在USB设备连接、上电或复位后，USB系统软件就可以使用该管道来和USB逻辑设备进行配置通信了。在USB设备被正确配置后，客户软件也会拥有对该管道的“使用权”，但其“所有权”仍属于USB系统软件，并由它来协调客户软件对该管道的访问。

不论是对USB系统软件还是对客户软件而言，其他非0号端点所对应的管道都只有在USB设备被正确配置后才可使用。按管道中所传输数据结构的不同，可将其分为如下两种类型：

流管道：在流管道中传输的数据不具有USB定义的结构，它可用作块传输、同步传输和中断传输，且通常是单向的。即如果客户软件要和USB设备完成玳传输和OUT传输，则必须使用两条流管道。

消息管道：在消息管道中传输的数据具有USB定义的结构，它只能用作控制传输，且通常是双向的，既支持IN传输又支持OUT传输。USB设备0号端点所实现的缺省控制管道就是一条消息管道。

2.2 USB传输管理

在USB总线上，所有与USB设备功能单元的数据传输都是由客户软件启动的，而且必须经过主机的USB系统软件和USB总线接口模块才能将数据发送到USB设备。其传输过程涉及四个软硬件实体：客户软件、USB总线驱动程序、USB主控制器驱动程序和USB主控制器。图2.1中的向下箭头表示了客户软件向USB设备发送数据的情况，其传输步骤如下：

1．客户软件把要传输的数据放入数据缓冲区，并向USB总线驱动程序发出IRP(I/O请求包)，以请求数据传输。

2．USB总线驱动程序响应客户软件发出的IRP，并将其中的数据转化为一个

个具有USB格式的事务处理，然后将其向下传递。

3．USB主控制器驱动程序负责为这些事务处理，建立一系列以帧/小帧为单位的事务处理列表，并保证其不会超过USB的带宽。

4．USB主控制器负责读取事务处理列表，并将其中的事务处理以信息包为单位发送到USB总线上，且传输速率可以是低速、全速或高速。

5．USB设备接收到这些信息包后，SIE自动将其解包并将数据放入指定端点的接收缓冲区内，由芯片固件对其进行处理。

同样，客户软件在接收USB设备发来的数据时也必须经过上述四个软件实体。

2.3 USB传输类型

不同的USB设备对数据传输提出了不同的要求，如传输数据量的大小、传输速率的高低、需同步传输或突发传输等。根据这些要求，USB定义了4种传输类型：控制传输、块传输、中断传输和同步传输，归纳了它们各自的特点。

2.3.1 块传输

块传输适用于传输大量的、且对传输时间和传输速率均无要求的数据。当USB总线带宽紧张时，它会为其他传输类型让出自己所占用的帧/小帧时间，而其本身将被延迟，这时块传输的传输速率很低、占用的传输时间也很长；当USB总线空闲时，它会以很快的速率传输，其传输时间也很短。所以块传输可以发送大量的数据而不会堵塞USB总线，但其传输时间和传输速率得不到保证。另外，它还采用差错控制和重试机制来确保数据传输的正确性。块事务处理一般包括令牌、数据和握手三个阶段。但在某些情况下，如端点被停止、传输出错等，其也可能用握手阶段来代替数据阶段、或彻底丢弃数据阶段和握手阶段，而产生只包含两个阶段或一个阶段的块事务。块事务处理是单向的——IN或者OUT，如果USB设备需要双向块传输的话，则必须使用两个块端点，且每个方向对应一个。

2.3.2 中断传输

中断传输适用于传输少量或中量的、且对服务周期有要求的数据。USB为中断传输保留了总线带宽，以保证其能在规定的周期内得到服务，但其并不是一直使用准确的传输速率。另外，USB还采用差错控制和重试机制来确保中断传输的正确性。中断事务处理包括烈传输和OUT传输，可具有令牌、数据和握手三个阶段。当主机准备接收中断数据时，它将发出IN令牌包，而USB设备将向其返回

DATAx数据包、NAK或STALL握手包；但如果USB设备接收到的IN令牌包有错误，则它将丢弃该信息包，并不作任何响应。当主机准备发送中断数据时，它将会发送出OUT令牌包和DATAx数据包，而USB设备将向主机返回ACK、NAK或STALL握手包；但如果USB设备接收到的OUT令牌包或数据包有错误，则它将丢弃这些信息包，并不作任何响应。即在某些情况下，中断事务可只包含两个阶段或一个阶段(令牌阶段)。

2.3.3 同步传输

同步传输适用于传输大量的、速率恒定的、且对服务周期有要求的数据。USB为同步传输保留了总线宽带，以保证其能在每帧/小帧中都能得到服务。即同步传输将一直使用准确的传输速率，因此其传输时间是可以预测的。另外，为确保数据传输的及时性，同步传输没有采用差错控制和重试机制，即不能保证每次传输都是成功的。同步事务处理是单向的，它包括IN传输和OUT传输，但其只具有令牌和数据阶段，而没有握手阶段。同步事物不使用任何握手包，对于发送方，不管数据接收是否成功，它总是在每一帧/小帧中连续发送数据，且不会对前一帧/小帧中出错的数据进行重传；对于接收方，它可以判断出数据传输是否发生了错误，但不会向发送方返回任何握手包。USB总线传输的误码率很低，大多数情况下数据都能被成功地发送和接收。

2.3.4 控制传输

控制传输适用于传输少量的、且对传输时间和传输速率均无要求、但必须保证传输的数据。USB为控制传输保留了总线带宽，且主机USB系统软件可以为它动态地调整其所需的帧/小帧时间，以确保其能够被尽快传输。另外，USB还采用差错控制和重试机制来保证控制数据传输的正确性。控制传输主要用于发送和接收与USB设备的配置信息有关的数据，如设备地址、配置描述符等，但它也可用于传输其他用途的数据。控制传输可用于低速、全速或高速设备，且所有的USB设备都必须支持控制传输。具体的说，任何USB设备都必须在其0号端点的缺省管道中支持控制传输，USB系统软件会使用该管道来访问USB设备的状态，并对其进行配置。除0号端点外，USB设备还可以拥有其他的控制端点嘲。控制事务处理包含建立、数据和状态三个阶段，每个阶段都由特定的事务组成。其中，建立阶段负责完成主机向USB设备发送控制请求，它们具有USB定义的格式，该阶段由一个SETUP事务组成；数据阶段是可选的，如果有，它将根据建立阶段指明的传输方向传输具有USB定义格式或设备类、供应商自定义格式的数据，该阶段包含一个

或多个IN/OUT事务；状态阶段用于USB设备向主机报告建立阶段和数据阶段的传输结果。其也具有USB定义的格式，它由一个OUT事务或一个IN事务组成。可以看出，在一次控制传输中，既要完成IN事务又要完成OUT事务，且传输的数据具有USB定义格式，所以需使用双向的消息管道。

2.4 本章小节

本章根据数据传输的基本理沦，针对实时保密通信系统中存在的数掘传输问题进行了研究，并采用USB来完成DSP与主机的快速通信。

第三章AES和ECC理论

有限域在密码编码学中的地位越来越重要，许多密码算法都依赖于有限域的性质。要了解 AES 及 ECC 加密体制就要首先了解有限域的性质。

3.1 群、环和域

群、环和域都是数学理论中的一个分支，即抽象代数或者称为近世代数的基本元素。

3.1.1 群

群 G，有时记做$\{G,\square\}$，是定义了一个二元运算的集合，这个二元运算可以表示为$\square$，G 中每个序列(a,b)通过运算生成 G 中的元素$(a\square b)$，并满足一下公理：

(1) 封闭性：如果a和b都属于 G，则$a\square b$也属于 G。

(2) 结合律：对于 G 中任意元素a、b、c，$a\square(b\square c)=(a\square b)\square c$都成立。

(3) 单位元：G 中存在一个元素e，对于 G 中任意元素a，都有$a\square e=e\square a=a$成立。

(4) 逆元：对于 G 中任意元素a，G 中都存在一个元素a'，使得式$a\square a'=a'\square a=e$成立。

如果一个群的元素是有限的，则该群称为有限群。群的阶等于群中元素的个数。否则，该群称为无限群。

一个群如果还满足条件(5)，则称为交换群。

(5) 交换律：对于 G 中任意的元素a、b都有$a\square b=b\square a$成立。

当群中的运算符是加法时，其单位元是 0；a的逆元是$-a$，减法用一下的规则定义：$a-b=a+(-b)$。

如果群中的每个元素都是一个固定的元素$a$$(a\in G)$的幂$a^k$($k$为整数)，则称群$G$是循环群[8]。我们认为元素$a$生成了群$G$，或者说$a$是群$G$的生成元。循环群总是交换群，它可能是有限群或无线群。

3.1.2 环

环 R，有时记做$\{R,+,\times\}$，是一个有两个二元运算的集合，这两个二元运算分

别称为加法和乘法，对于 R 中的任意元素 a、b、c，满足以下公理：

R 关于加法是一个交换群。对于此种情况下的加法群，我们用 0 表示其单位元，$-a$ 表示 a 的逆元。

(1) 乘法的封闭性：如果 a 和 b 都属于 R，则 ab 也属于 R。

(2) 乘法的结合律：对于 R 中的任意元素 a、b、c，有 $a(bc)=(ab)c$ 成立。

(3) 分配率：对于 R 中的任意元素 a、b、c，式 $a(b+c)=ab+ac$ 和式 $(a+b)c=ac+bc$ 总成立。

本质上说，环就是一个集合，我们可以在其上进行加法、减法$[a-b=a+(-b)]$和乘法，而不脱离该集合。

环如果还满足条件(4)，则称为交换环。

(4) 乘法的交换律：对于 R 中的任意元素 a、b，有 $ab=ba$ 成立。

而整环就是满足条件(5)的交换环。

(5) 乘法的单位元：在 R 中存在元素 1，使得对于 R 中的任意元素 a，有 $a1=1a=a$ 成立。

(6) 无零因子：如果有 R 中元素 a、b，且 $ab=0$，则必有 $a=0$ 或者 $b=0$。

3.1.3 域

域 F，有时记做 $\{F,+,\times\}$，是有两个二元运算的集合，这两个二元运算分别称为加法和乘法，且对于 F 中的任意元素 a、b、c，满足一下公理：

F 是一个整环。

乘法逆元：对于 F 中的任意元素 a(除 0 以外)，F 中都存在一个元素 a^{-1}，使得式 $aa^{-1}=(a^{-1})a=1$ 成立。

本质上来说，域就是一个集合，我们可以在其上进行加法、减法、乘法和除法儿不脱离该集合。除法又按以下规则来定义：$a/b=a(b^{-1})$。

对称加密，也称传统加密或者单钥加密，是公钥密码产生之前额日益的加密技术。直到今天，它仍然是使用最为广泛的一种加密类型。

首先，来定义一些术语。原始的消息称为明文，而加密后的消息称为密文。从明文到密文的变化过程称为加密；从密文到明文的变换过程称为解密；加密方案则称为密码体制。

3.2 AES密码体制概述

3.2.1 AES算法数学基础

1．有限域

无限域在密码编码学中没有特别的意义，然而有限域却在许多密码编码学算法中扮演者重要的角色，可以看出，有限域的元素个数必须是一个素数的幂 p^n，n 为正整数[9]。有限域通俗的来讲其实就是有限个元素的域，域中元素个数 p^n 称为域的阶。q 阶域存在当且仅当 $q=p^n$，p 为素数并称为有限域的特征，有限域记为 $GF(p^n)$[10]。GF 代表 Galois field，以第一位研究有限域的数学家的名字命名。

AES 算法中涉及两类如公式(3-1)和(3-2)形式的特征域 2 上的多项式

$$b(x)=b_7x^7+b_6x^6+b_5x^5+b_4x^4+b_3x^3+b_2x^2+b_1x+b_0, b_i\in GF(2^8) \quad (3\text{-}1)$$

$$a(x)=a_3x^3+a_2x^2+a_1x, a_i\in GF(2^8) \quad (3\text{-}2)$$

具有相同阶数的有限域是同构的，即它们具有相似的代数结构，其区别仅在于元素的表示。对于每一个素数幂，恰好存在一个有限域，我们用 $GF(p^n)$ 表示。

2．有限域上的多项式

在有限域 GF 上定义形如多项式(3-3)：

$$b(x)=b_{n-1}x^{n-1}+b_{n-2}x^{n-2}+\cdots+b_2x^2+b_1x+b_0 \quad (3\text{-}3)$$

x 称为有限域上多项式的变元，$b_i\in GF$ 是多项式的系数。我们仅将有限域上的多项式看作为抽象的实体而不参与计算，由于不对多项式中的各项之和经行计算，因此即使多项式定义在 2 的特征域上，我们仍然使用符号"+"。

3．有限域上的运算

有限域上的基本运算包括模加、模乘、求逆等[11]。模加比较简单，模乘运算一般分成乘运算和模 p 规约来实现。虽然运算不复杂，但是在实际应用中有限域的阶都很大，计算起来还是很费时间的。采用常见的 32 位体系结构实现。假设数组为

$a=(a_t,a_{t-1},\cdots,a_1)$，整数 n 表示为(3-4)：

$$n=\sum_{j=1}^{t}a_j\square 2^{32(j-1)} \text{(其中 t=6)} \quad (3\text{-}4)$$

(1)有限域上的加法

有限域上的加法就是异或运算。先将两个操作数按低位对齐，然后从右到左对数组里的每一个元素进行的加减法运算。每一次相加，产生一个临时的 64 位的和数，将和数低 32 位复制到新数组对应元素中，和数的的高位为 0 进/借位，该进

位被加到左边相邻的数组元素的运算中。当加法运算的结果大于 p-1，需要将它们分别减去 p，以使结果落在有限域内。

算法：模 p 上的加法

输入：模 p，整数 a,b∈[0,p-1]

输出：c= (a+b)mod p

```
1. r=0; temp=0;
2. For(i=1; i≤t; i++)
{temp=temp+a[i];
temp=temp+b[i]+r;
c[i]=temp&0xFFFFFFFF;
r=temp>>32;
};
3. c=c mod p;
4. Return(c).
```

(2)有限域上的乘法运算

简单的异或不能完成乘法运算，但是可以使用容易实现的技巧。思想就是按位相乘，然后将每一位相乘的结果移位相加。

算法：模 p 上的整数乘法

输入：整数 a,b∈[0,p-1]

输出：c= a·b

```
1.temp=1;
2.for(i=1; i≤t; i++)
{r=0; t=l;
for(j=1;j≤t;i++)
        {temp=temp×a[i];
        temp=temp×b[j];
        temp=temp+c[i+j-1]+r;
        c[i+j-1]=temp&0xFFFFFFFF;
        r=temp>>32;
        }
        c[i+j-1]=r;
}
3.c=c mod p;
4.Return(c).
```

(3)有限域上求乘法逆元的运算

当 p 值较小时，求$GF(p)$中元素的乘法逆元很容易。只需构造一个乘法表，所要的结构就可以直接得到。但是当 p 值比较大的时候，这种方法就不切实际了。

如果 gcd(m，b)=1,那么 b 有模 m 的乘法逆元。已知 b 和 m(b，m 均为非零元素)，且 gcd(b，m)=1(b 和 n 的最大公约数为 1)，对于 b<m，存在 b^{-1}<m 使 bb^{-1}=1 mod m。

算法：模 p 上的乘法逆元

输入：整数 b,m∈[0,p-1]

输出：$c=b^{-1}$

1.(A1, A2, A3)←(1, 0, m)；(B1, B2, B3)←(0, 1, b);

2.IF B3=0 Return A3=gcd(m,b); 不可逆

3.IF B3=1 Return B3=gcd(m,b); $B2=b^{-1}$ mod m

4. $Q=\left\lfloor \frac{A3}{B3} \right\rfloor$

5.(A1, A2, A3)←(A1-QB1, A2-QB2, A3-QB3);

6.(A1, A2, A3)←(B1, B2, B3);

7.(B1, B2, B3)←(T1, T2, T3);

8.转到 2.

3.2.2 AES算法结构描述

在 Rijndael 算法是一个可变数据块长和可变密钥长的分组迭代加密算法，分组长度和密钥长度均可分别为 128，192 或 256 位，但在 AES 标准规范中，分组长度只能为 128 位，密钥长度可为 128，192 或 256 位[12]。AES 标准中很多参数都与密钥长度有关，见表 3-1。

表 3-1 AES 轮密钥信息

密钥长度 (位)	轮数 (次)	每轮的密钥长度 (位)	扩展密钥长度 (位)
128	10	128	176
192	12	128	208
256	14	128	240

AES 每一轮都是用代换和混淆并行的处理整个数据分组，输入或输出数据的所有比特组成一个“状态”(State)，相邻的 32 个数据比特或密钥比特称为一个

“字”(Word)，Nb、Nk 分别表示数据分组和加密密钥的字长度，对于 128，192，256 比特，Nb、Nk 分别取值 4，6，8。则可以将数据分组的 State 和密钥都表示为字节矩阵的形式。例如当 Nb=6，Nk=4 时的矩阵如表 3-2 所示。

表 3-2 当 Nb=6,Nk=4 矩阵形式

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$	$b_{0,4}$	$b_{0,5}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,4}$	$b_{1,5}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	$b_{2,4}$	$b_{2,5}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,4}$	$b_{3,5}$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

AES 采用的是替代、置换(SP)网络结构。

每一轮由 3 层组成：

(1) 非线性层：进行 SubByte 变换(即 S 盒替换)，起到混淆的作用；

(2) 线性混合层：进行 ShiftRow 行变换运算和 MixColumn 列变换运算以确保多轮之上的高度扩散；

(3) 密钥加层：子密钥简单的异或到中间状态上。

3.2.3 AES算法轮变换

轮变换由 4 个不同的阶段组成，包括一个混淆和三个代换：

字节代换(SubByte)：用一个 S 盒完成分组中的按字节的代换。

行移位(ShiftRow)：一个简单的置换。

列混淆(MixColumn)：一个利用在域$GF(2^8)$上的算术特性的代换。

轮密钥加(AddRoundKey)：利用当前分组和扩展密钥的一部分进行按位异或。

用 C 语言伪代码描述为：

```
Round(State，RoundKey)
{
SubByte(State);
ShiftRow(State);
MixColumn(State);
AddRoundKey(State,RoundKey);
}
```

最后一轮运算稍有不同，少了列混淆变换也就是 MixColumn(State)函数。

```
FinalRound(State,RoundKey)
{
```

SubByte(State);

ShiftRow(State);

AddRoundKey(State,RoundKey);

}

接下来详细介绍轮变换中的各个阶段

1．字节代换(SubByte)

字节代换的正向变换是一个简单的查表操作，代替了基于矩阵乘法的复杂仿射变化。在中 AES 定义了一个 S 盒，它由 16×16 个字节组成的矩阵包含了 8 位所能表达的 256 中可能的变换。在 State 中把每个字节的高 4 位作为行值，低 4 位作为列值，然后取出 S 盒中对应行列的元素作为输出。也就是说将每个字节通过 S 盒做非线性运算。

S 盒按如下的方式构造：

(1) 把 S 盒中的每个字节映射为它在有限域$GF(2^8)$中的乘法逆；{00}被映射为它自身。

(2) 把 S 盒中的每个字节记为$(b_7,b_6,b_5,b_4,b_3,b_2,b_1,b_0)$。对 S 盒中的每个字节的每位做(3-5)变换：

$$b_i' = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i \qquad (3\text{-}5)$$

这里的c_i是指值为{63}的字节 c 的第 i 位。

即$(c_7,c_6,c_5,c_4,c_3,c_2,c_1,c_0)=(01100011)$。符号 ´ 表示更新后的变量的值。AES 是以公式(3-6)的方式用矩阵描述这个仿射变换：

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad (3\text{-}6)$$

这两个变化构造出 S 盒。S 盒在状态的所有字节上的运算用 SubByte(State)来表示。

逆字节代换也就是所谓的逆 S 盒变换，它则是利用查逆 S 盒来进行字节替换。首先进行逆仿射运算，然后求出输入字节的乘法逆。该逆变换是由$GF(2^8)$上的逆变换得来的,也可以理解为逆字节代换是字节代换的逆。如公式(3-7)所示：

$$b_i' = b_{(i+2)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus d_i \qquad (3\text{-}7)$$

这里，字节 d={05}，或者 00000101。也可以用公式(3-8)来描述这个转换：

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3-8}$$

2．行移位变换(ShiftRow)

正向行移位变换，也就是行左平移运算。即 State 的行以不同的位移向左循环平移，State 第一行保持不变。State 第二行循环左移一个字节，State 第三行循环左移两个字节，State 第四行循环左移三个字节。逆向行移位变换就是将 State 中的正向行移位操作进行相反方向的移位操作。

其实，行移位就是将某个字节从一列移到另一列中，它的线性距离是 4 字节的倍数。

这一变换用 ShiftRow(State)来表示。

3．列混淆变换(MixColumn)

列混淆变换的正向列混淆变换是对每列独立的进行操作，我们把 State 每一列中的每个字节映射为一个新值，它是由该列中的四个字节通过函数变换得到的。如公式(3-9)所示：

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \tag{3-9}$$

矩阵的系数基于码字见有最大距离的线性编码，这使得在每列的所有字节中有良好的混淆性。乘法矩阵中的每个元素都是一行和一列中所对应元素的乘积之和。在这里的乘法和加法都是定义在 $GF(2^8)$ 上的。State 中的第 j 列(0≤j≤3)的列混淆变换可表示为公式(3-10)：

$$\begin{aligned} s'_{0,j} &= (2\square s_{0,j}) \oplus (3\square s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ s'_{1,j} &= s_{0,j} \oplus (2\square s_{1,j}) \oplus (3\square s_{2,j}) \oplus s_{3,j} \\ s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2\square s_{2,j}) \oplus (3\square s_{3,j}) \\ s'_{3,j} &= (3\square s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2\square s_{3,j}) \end{aligned} \tag{3-10}$$

逆向列混淆变换可由矩阵乘法定义(3-11)：

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \quad (3\text{-}11)$$

列混淆变换和行移位变换使得在经过几轮变换后，所有的输出为均与所有的输入为相关。列混淆变换的系数受到算法执行效率的影响，这些系数的乘法涉及到之多一次移位和一次异或。

4．轮密钥加变换(AddRoundKey)

在正向轮密钥加变换中，将 State 矩阵和轮密钥进行简单的异或。我们能把这个操作看成是基于 State 列的操作，也就是把 State 的一列中的四个字节与轮密钥的一个字进行异或；将其视为字节级别的操作。

逆向轮密钥加变换与正向轮密钥加变换相同，异或操作是其本身的逆。

轮密钥加变换非常简单，却能影响 State 中的每一位。密钥扩展的复杂性和 AES 的其他阶段运算的复杂性，确保了该算法的安全性。

3.2.4 AES算法加、解密

AES 是一种对称密钥迭代型分组密码算法，将明文的数据分组看成是字节的二维矩阵(状态矩阵)，该矩阵有 4 行，Nb 列(Nb 为分组长度除以 32)。在状态矩阵上进行若干次迭代，实现对明文的混乱和扩散，达到加密的目的。其加密和解密的流程，如图 3.1 所示：

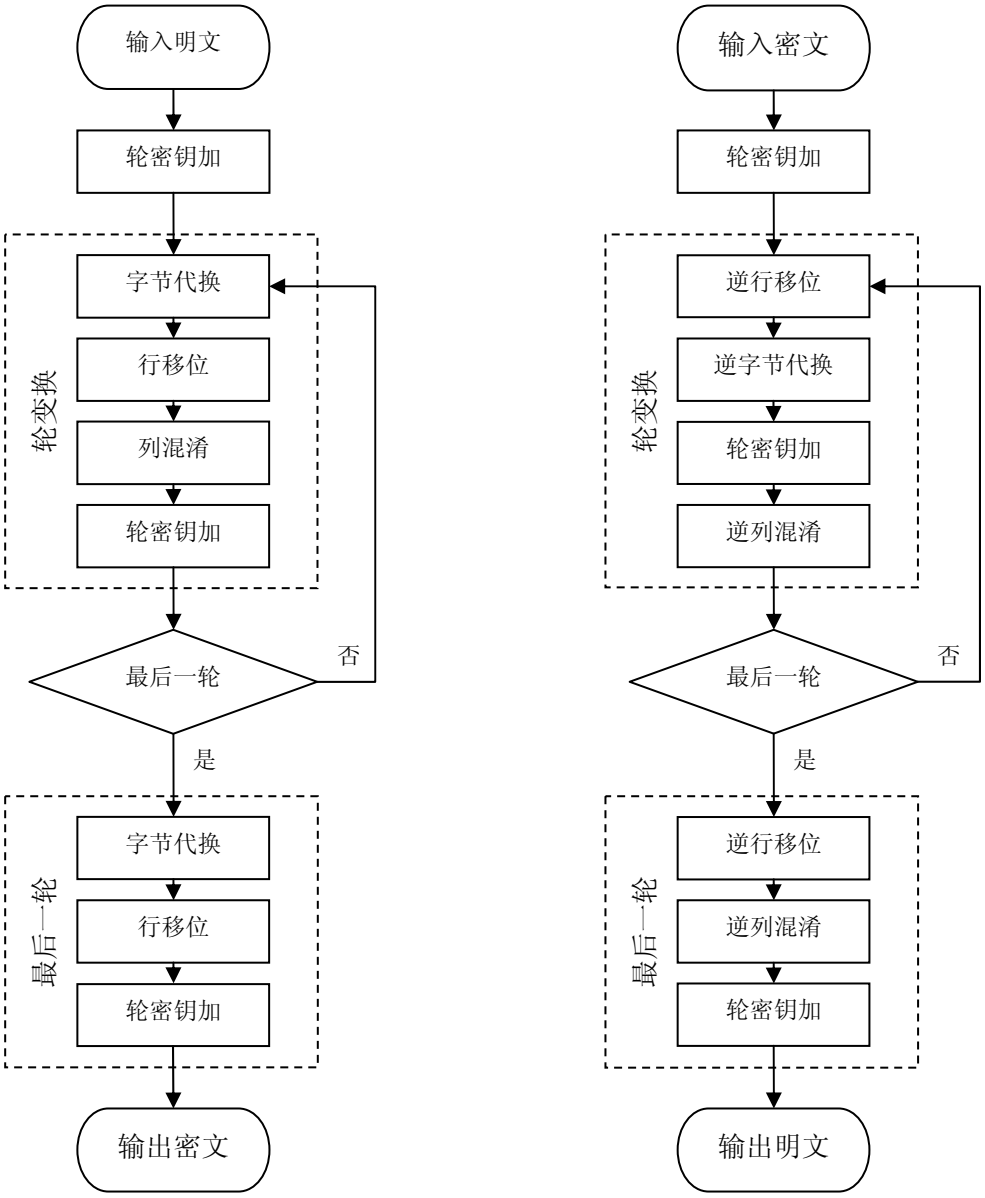

图 3.1 AES 加密解密流程图

3.3 ECC密码体制概述

3.3.1 ECC算法数学基础

自从 1976 年 Diffie 和 Hellman 提出了公钥密码之后各种密码体制纷纷出台，他们几乎均是基于某个数学难题的，而其中大数分解和离散对数问题是公钥密码最核心的两个难题[13]。

椭圆曲线加密系统就是建立在有限域上椭圆曲线的离散对数问题 ECDLP 的计算困难性上，现有的攻击算法表明，基于有限域椭圆曲线上的离散对数问题的困

难性要高于一般乘法群上的离散对数问题的困难性。

1．射影坐标系

通常来说两条平行直线永远不相交，不过到了近代这个结论遭到了质疑。平行线会不会在很远很远的地方相交了？事实上没有人见到过。所以“平行线，永不相交”只是假设。既然可以假设平行线永不相交，也可以假设平行线在很远很远的地方相交了。即平行线相交于无穷远点 P∞如图 3.2 所示：

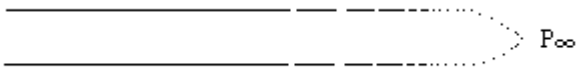

图 3.2 平行线相交于无穷远点 P∞

直线上出现 P∞点，所带来的好处是所有的直线都相交了，且只有一个交点。这就把直线的平行与相交统一了。为与无穷远点相区别把原来平面上的点叫做平常点。

以下是无穷远点的几个性质：

(1)直线 L 上的无穷远点只能有一个；

(2)平面上一组相互平行的直线有公共的无穷远点；

(3)平面上任何相交的两直线 L1，L2 有不同的无穷远点；

(4)平面上全体无穷远点构成一条无穷远直线；

(5)平面上全体无穷远点与全体平常点构成射影平面[14]。

下面我们就来了解一下射影坐标系， 射影平面坐标系是对普通平面直角坐标系(仿射坐标系)的扩展。我们知道普通平面直角坐标系没有为无穷远点设计坐标，不能表示无穷远点。为了表示无穷远点，产生了射影平面坐标系，而原有的平常点在射影平面坐标系下同样能很好的表示。

对普通平面直角坐标系上的(x,y)做如下改造：

令 x=X/Z ， y=Y/Z (Z≠0)；则(x,y)点可以表示为(X,Y,Z)。 此时就变成了有三个参量的坐标点，这就对平面上的点建立了一个新的坐标体系。

在普通直线坐标系下的直线方程是 ax+by+c=0，同样在新坐标系下直线的方程是 aX+bY+cZ=0，那么平行直线的方程是：

$$aX+bY+c_1Z=0;aX+bY+c_2Z=0 \quad (c_1\neq c_2)$$

求解得到无穷远点用这种形式(X,Y,0)表示。由此就可以得到平常点为 Z≠0，无穷远点为 Z=0。

2．椭圆曲线的定义

椭圆曲线来源于椭圆积分的研究，为此引入了椭圆函数[15]。椭圆曲线定义：设 F 是一个数域，则由 Weierstrass 方程如式(3-12)所示:

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6 \qquad (3\text{-}12)$$

其中 a_1，a_2，a_3，a_4，$a_6 \in K$。满足 Weierstrass 方程的解集称为 F 域上的椭圆曲线的点，椭圆曲线通常用 E 表示，它是曲线 E 上的所有点，再加上无穷远点 O 称为 F 上的椭圆曲线[16]。

若 F 的特征为 2，则方程可以变换为如式(3-13)所示:

$$y^2 + xy = x^3 + ax^2 + b \tag{3-13}$$

若 F 的特征为 3，则方程可以变换为如式(3-14)所示:

$$y^2 = x^3 + ax^2 + bx + c \tag{3-14}$$

若 F 的特征大于 3，则方程变换为如式(3-15)所示:

$$y^2 = x^3 + ax + c \tag{3-15}$$

设 F(x，y)=0 为上述各椭圆曲线方程的隐式方程，则若点 P(x，y)使得 $\partial F / \partial x$ 不同时为零，称 P 为非奇异点。

在常见的椭圆曲线加密体制中，利用定义在有限域上的椭圆曲线，其方程形如：

$y^2 = x^3 + ax + c \pmod p$，这里 p 是一个素数或 2 的幂，且 $4a^3 + 27b^2 \neq 0$

3．椭圆曲线运算

椭圆曲线密码体制总共分 4 个层次。如图 3.3 所示：

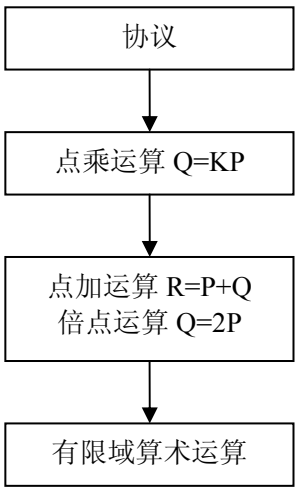

图 3.3 椭圆曲线运算层次

最上层的协议层，也就是所说的密码体制协议。

椭圆曲线的关键运算为点乘运算，通过椭圆曲线上的点加与倍点运算来实现，而点加与倍点运算又进一步通过有限域的算术运算来实现，如模加、模乘、模逆、模平方等运算来实现[17]。

(1) 点加运算：

设 $P(x_1，y_1)$，$Q(x_2，y_2)$是 E 上任意两点，L 是 P、Q 两点之间的连线，$P \neq \pm Q$，

称 P+Q 是椭圆曲线上的点加运算；若 P 和 Q 重合于一点，即 P=Q，则 L 为过 P 点的切线。若 L 和椭圆曲线相交于另一点 R′，该点关于 x 轴的对称点 R 也在椭圆曲线上，则 R=P+Q=2P，称椭圆曲线上的倍点运算。

点加运算

设 $P(x_1, y_1)$，$Q(x_2, y_2)$，$P \neq Q$，$P+Q=(x_3, y_3)$，如图 3.4 所示。它们都是椭圆曲线上的点，则点加公式如公式(3-16)所示：

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \qquad \text{其中 } \lambda = \frac{y_2 - y_1}{x_2 - x_1} \tag{3-16}$$

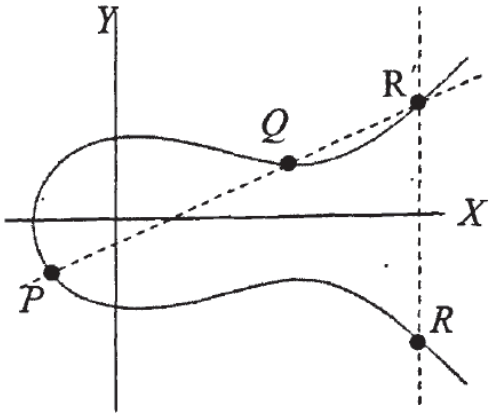

图 3.4 点加运算

倍点运算

设 $P(x_1, y_1)$，$P+P=2P=(x_3, y_3)$，均是椭圆曲线上的点，如图 3.5 所示。则其点加的公式如公式(3-16)所示：

$$\begin{aligned} x_3 &= \lambda^2 - 2x_1 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \qquad \text{其中 } \lambda = \frac{3x_1^2 + a}{2y_1} \tag{3-17}$$

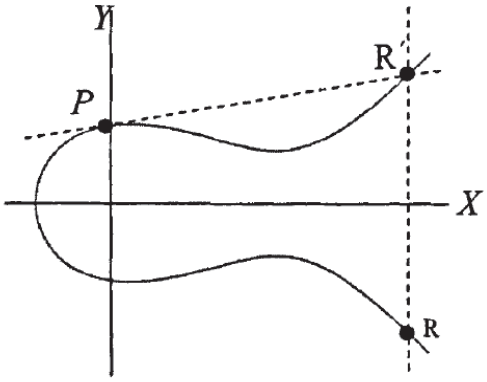

图 3.4 倍点运算

(2) 点乘运算

标量乘法是椭圆曲线的核心计算，所谓点乘，就是对于给定椭圆曲线上的点 P 和正整数 k，并且 k 不能大于 P 的阶。求 k 个 P 相加的结果，记作公式(3-18)：

$$Q = kP = P + P + P + \cdots + P \tag{3-18}$$

由于点乘运算是椭圆曲线密码体制中的主要运算，它的运算速度直接决定了系统的性能，因此设计椭圆曲线密码系统的关键就在于点乘的实现，而点加运算与倍点运算归结到底是为了点乘服务的。

目前基于点乘算法的快速实现方法主要有：二进制表示法、m 进制表示法、以及滑动窗口算法等。其中尤以二进制表示法的思想简单，就是把整数 k 用二进制表示，然后进行点加、倍点运算。因此本文采用二进制算法计算点乘 kP。

椭圆曲线密码系统还要用到大数乘、大数取模、模乘、模加、模逆、模幂这些基本运算。其中大数乘、大数取模属于整数域上的运算，模乘、模加、模逆、模幂属于有限域上的运算，而点加和点乘属于椭圆曲线的点加群上的运算。

$GF(2^m)$上椭圆曲线点的普加需执行的域运算有 1 次求逆、2 次乘、1 次平方和 8 次加；点的倍加需执行的域运算有 1 次求逆、2 次乘、2 次平方和 5 次加。

$GF(F_p)$上点的普加，需执行的域运算有 1 次求逆、2 次乘、1 次平方和 6 次加；点的倍加需执行的域运算有 1 次求逆、2 次乘、2 次平方和 8 次加。

4．椭圆曲线离散对数问题

(1) 离散对数问题

离散对数问题被定义为在群 Z_p 上寻找对数的问题。给定一个 n 阶的元素 $g \in Z_p$，并给定 $h \in Z_p$，寻找一个整数 $x(0 \le x \le n-1)$，使它满足 $gx \equiv h(\mathrm{mod}\ p)$，这一整数 x 被称为 h 基于 g 的离散对数。例如：取 p=17，g=10 是 Z_{17} 中阶为 n=16 的元素，h=11，由于 $1013 \equiv 11(\mathrm{mod}\ 17)$，故 h 基于 g 的离散对数是 13。

这个概念使用于任意群。假如 G 为一 n 阶的群，a 为 G 中的一个元素，对于 G 的离散对数为：给定元素 $\alpha^x=\beta$。

一个群用作加密的应具备两个因素：

a) 群中的操作在硬件及软件方面较其它群易于实现；

b) 群中的离散对数问题比 Z^*_p 中的离散对数问题更为难解。因此，在相同的安全级别下人们会使用比 Z^*_p 更小的群 G，这样会使得秘钥长度短，节省带宽和执行速度快。

(2) 椭圆曲线离散对数问题

如果 q 是一个素数幂，那么 Z_p 表示包含 q 个元素的有限域。椭圆曲线离散对数的问题是：取定一个定义在有取域 Z_p 上的椭圆曲线 E，给定一个 n 阶的点 $P \in E(Z_p)$和一个点 $Q \in E(Z_p)$，寻找一个整数 $x(0 \le x \le n-1)$使得 Q=xP，那么 x 称为 Q 基于 P 椭圆曲线离散对数问题。

椭圆曲线密码系统(ECC)的安全性取决于椭圆曲线离散对数问题。到现在为止还没有有效的算法解决椭圆曲线离散对数问题。事实上，椭圆曲线密码系统的一个优势在于它比整数因数分解问题和以 P 为模的离散对数问题更为难解。因为在椭圆曲线上点加操作比较复杂，所以这些方法总体来说比较慢。

3.3.2 ECC算法参数选取

椭圆曲线域参数(Domain Parameters)是指定义在域上的一条椭圆曲线及其与密码学应用相关的参数的集合[18]。椭圆曲线的阶定义为曲线上点的个数，记作#E(F_p)。由于椭圆曲线群上点的有限性和点数目的难确定性，从而给数据加密带来许多好的特性，因为这些曲线只是包含了一些离散的点，攻击者不知如何把这些点连成曲线，也就无法知道如何应用几何关系[19]。

假定曲线上一点 G，它的阶定义为满足 rG=O 的最小正整数。若 k 和 l 是整数，且 kG=lG，则必有 k=l(mod r)。椭圆曲线密码体制的实现由椭圆曲线的域参数决定，这些参数包括椭圆曲线的基域、曲线方程、曲线的基点、定义在曲线上的运算以及域和基点的阶，可用一个六元组 T 表示：T=(p, a, b, G, n, h)，其中：

p 是一个足够大的一个素数，表示一个有限域 F_p；

参数 a,b∈F_p 为椭圆曲线方程的系数，且满足 $4a^3+27b^2\neq0$；

G 为椭圆曲线 E (Fp)上一个点，称为基点；

n 满足 nG=O，并且 n 是一个大素数；

h= #E(F_p)/n，记为比例因子，它是一个小整数。

在以上椭圆曲线的域参数中，这里主要的安全性参数是 n，故 ECC 密钥的长度就定义为 n 的二进制位数。因此 ECC 密钥的长度就定义为 n 的长度。由以上的参数可以确定唯一一个椭圆曲线。

ECC 域参数的选取和验证是 ECC 安全性的重要保证，必须选取合适和安全的域参数以高效的实现 ECC 并抵御各种已知的攻击[20]。

3.3.3 ECC算法加、解密

1．密钥对的产生

选取适当椭圆曲线域参数，设椭圆曲线的域参数 T=(p, a, b, G, n, h)，点 G(x, y)是椭圆曲线域参数中定义的基点，G 的阶数为 n(n 为一个大素数)。在[1,n-1]之间随机地确定一个整数 K_S，计算 $K_P=K_SG$，且 K_P 为椭圆曲线 E_p(a, b)上的一点，由此就确定了密钥对(K_S, K_P) [21]。K_S 为私钥，K_P 为公钥。密钥对是安全通信的关键，任何使用 ECC 的用户都有一套生成密钥对和验证密钥对的算法来确保密钥对的正

确性，从而保证通信的正常进行[22]。

2．加密、解密

椭圆曲线加密体制只是一种思想，要将它应用于实际的密码系统，还要定义密码算法。ElGamal 是一个比较成熟的算法，将其移植到椭圆曲线上，从而可以得出基于椭圆曲线的加密、解密与数字签名算法[23]。

在对明文 M 加密之前需要将 M 映射到椭圆曲线的有限域 F_p 的任意一点上[24]。如果 M 较长则需要分段处理，设 m (m 需满足 $0\leq m\leq [p/256]-1$)为 M 的一个分段，后将 m 映射到点 $P_m(x, y)$上使得：$256\ m\leq x\leq 256(m+1)$，$P_m(x, y)\in F_p$

(1)加密算法：

发送方 A 随机选取 $r\in\{1,2,\ldots,n-1\}$；

计算 R=rG 的值；

将明文用编码函数嵌入椭圆曲线；

计算 $C=P_m+rK_{BP}$；

向 B 发送(R,C)。

(2)解密算法：

接收方 B 计算 P_m，$C-K_{BS}R=P_m+rK_{BP}-K_{BS}rG=P_m+rK_{BP}-rK_{BP}G=P_m$；

反向编码函数将明文 M 从 P_m 中恢复出来。

(3)数字签名过程：

选取一个公开消息摘要函数；

计算消息明文的摘要 H(m)；

发送方 A 取随机数 s，$s\in\{1,2,\ldots,n-1\}$；

计算 $R_2=sG=(x_2,y_2)$，$e=x_2H(m)$,$k=s+eK_{AS}$,$w=kG$，由此产生二元组(w,e)作为发送方 A 对消息签名；

计算 $R=w-eK_{BS}=(x_r,y_r)$,如果 $e=x_rH(m)$成立则签名有效，否则无效。

3.4 AES和ECC混合密码体制

3.4.1 AES和ECC混合密码体制的提出

在对称加密体制中，使用同一个密钥对数据进行加、解密，它具有运算开销少、速度快、便于实现等优点，但在网络传输过程中，密钥容易泄露。另外如果网络上有 n 个用户需要互相传输加密数据，则需要 n(n-1)/2 个密钥，从而使得密钥的分发和管理比较困难。

在非对称加密体制(公钥加密体系)中，数据加密和解密采用不同的密钥，而且用加密密钥加密的数据只有采用相应解密密钥才能正确解密数据，并且由加密密

钥来求解解密密钥十分困难。在实际应用中，用户通常对外公开加密密钥(公钥)，而秘密持有解密密钥(私钥)，从而使得公钥密码体系不仅能适应网络的开放性要求，密钥分发和管理简单，而且能方便得实现数据签名和身份验证等功能，是目前电子商务等技术的核心基础[25]。但是，相对于对称加密算法而言，公钥加密算法比较复杂，实现的速度比较慢、效率也较低。所以一般不用于加密大块数据，通常用于传输密钥、数据签名等方面。

由于对称密码和公钥密码都具有自身的局限性，而其局限性又恰好可以通过另一种密码体制来弥补。将两者相结合，形成一种新的密码体制—混合密码体制，即用对称加密算法加密报文数据，用公钥加密算法生成包括数字签名和对称加密算法所使用的加密密钥的数字信封，从而既完成了数字签名，又保障了数据在网络传输过程中的安全性。由于明文数据一般较大，如果使用 ECC 加密计算量太大，影响加密速度，使用 AES 不但能保证加密的安全性，且能更大的发挥其高效性。采用 ECC 对 AES 密钥进行加密，可以避免通信双方事先交换密钥，更大的保证了其安全性。使用 ECC 进行密钥管理和数字签名，保证了信息传递的不可否认性，同时克服了 AES 密钥管理困难的问题。

混合密码体制通过结合了两种加密体制，克服了各自的局限性，从而成为人们研究的重点。随着人们对加密算法的要求提高，使用单一的加密算法已经不能满足人们的需求。AES 出现以来，依其明显的优势压倒了 DES，已经开始逐渐代替 DES。同时 ECC 算法作为最优秀的公钥加密算法之一，随着人们对其研究的深入也开始渐渐代替目前市场的主流 RSA 算法。但是由于两种算法的自身的局限性，影响了它们的应用范围，将两种算法结合起来，克服各自的局限性，得到的混合密码体制必将在密码学领域有着广泛的前景。

所以，AES 和 ECC 相结合的混合密码体制适应目前社会的需求，有着重要的商业价值和军事价值，值得研究。

3.4.2 AES和ECC混合密码体制的性能分析

AES 和 ECC 混合密码体制的性能与其他密码体制相比，有着独特的优点，这也就是我们研究这种密码体制的出发点。下面我们通过与其他的密码体制相比较的方法来说明这种混合密码体制的优越性。

首先，混合密码体制与单一的密码体制相比较。混合加密体制与对称加密体制相比：对称加密体制有着算法实现后效率高,速度快的优点，但是对称加密体制的密钥的管理过于复杂，同时，由于数据加密和解密采用的都是同一个密钥，所以其安全性就很难得以保障。混合密码体制中使用公钥加密算法对密钥加密，使用公钥加密体制进行密钥管理，从而克服了对称加密体制中密钥管理的问题，同

时提供了数字签名和身份验证等功能，从一定程度上提高了安全性，扩展了应用的范围。

混合密码体制与公钥密码体制相比：由于公钥加密体制中，数据加密和解密采用不同的密钥，而且用加密密钥加密的数据只有采用相应的解密密钥才能解密，更重要的是从加密密钥来求解解密密钥十分困难。公钥密码体系能适应网络的开放性要求，密钥管理简单，并且可方便地实现数字签名和身份认证等功能。但是非对称加密体重有着算法复杂，加密数据的速度和效率较低，一般不利于大数据块的加密。在混合密码体制中，数据明文是采用的对称加密算法，速度和效率明显高于单一的公钥加密体制，同时也克服了公钥加密算法不利于大数据块加密的问题。

由此可知，虽然混合加密体制的速度和效率比单一的对称加密算法慢一些，但是比公钥加密算法要快很多，同时它可以克服了对称加密算法密钥管理困难的问题，使得两种加密算法的应用范围得到了扩展。与 RSA 和 DES 相结合的混合密码体制相比:我们分别通过 ECC 和 RSA、AES 和 DES 相比较,得出 ECC 和 AES 算法分别优于 RSA 和 DES 算法，同样可以得出结论：AES 和 ECC 相结合的混合加密体制也优于 RSA 和 DES 相结合的混合密码体制。

1．对称加密算法，即 AES 和 DES 相比较

(1)加解密效率：

AES 算法拥有加密高效的特点,其加解密速度与 DES 相比有着明显的优势。AES 分组长度为 128 比特,密钥长度也为 128 比特,DES 分组长度为 64 比特，密钥长度为 56 比特。

AES 的加密速度是 DES 的三倍左右,而解密速度也接近 DES 的两倍，AES 的效率明显优于 DES，如表 3-1 所示。

表 3-1AES 和 DES 加解密效率比较

算法	加密速度	解密速度
AES	5.0Mb/s	3.0Mb/s
DES	1.7Mb/s	1.7Mb/s

(2)安全性：

首先，作为分组密码，DES 的加密单位仅有 64 位二进制，这对于数据传输来说太小，因为每个分组仅含 8 个字符，而且其中某些位还要用于奇偶校验或其他通讯开销。

其次，DES 的密钥的位数太短，只有 56 比特，而且各次迭代中使用的密钥 Ki 是递推产生的，这种相关必然降低密码体制的安全性，在现有技术下用穷举法寻找密钥已趋于可行，所以若要保护 10 年以上的数据最好不要用 DES 算法；最

后，DES 不能对抗差分和线性密码分析。

此外，与 DES 相比，AES 标准支持可变分组长度，分组长度可设定为 32 比特的任意倍数，最小值为 128 比特，最大值为 256 比特。

AES 的密钥长度比 DES 大，用穷举法是很难破解的，假设取密钥长度为 128 比特，那么要破解 2128 个可能的密钥的密文平均要尝试的密钥的个数为 2127 个。也就是要进行 2127 次解密运算，使用一台 PC 机用穷举法破解大概的时间是 2.7×1026 年，这实际上是不可行的。最后，AES 算法的设计策略是宽轨迹策略(Wide Trail Strategy 即 WTS)，WTS 是针对差分分析和线性分析提出的，可对抗差分密码分析和线性密码分析。

可见，AES 无论在安全性还是在效率上都明显优于 DES，同时它还比 DES 更加灵活。所以，在混合加密系统中使用 AES 要好于使用 DES。

2．公钥加密算法，即 ECC 与 RSA 相比较

(1) 安全性：

加密算法的安全性能一般通过该算法的抗攻击强度来反映。ECC 和 RSA 相比，其抗攻击性具有绝对的优势[26]。椭圆曲线上的点群离散对数问题(ECDLP)计算困难性在计算时间复杂度上目前是完全指数级，而 RSA 是亚指数级的。这体现 ECC 比 RSA 的每 bit 安全性能更高。Pollardrho 方法是以知求椭圆曲线对数的最快方法，具体比较见表 3-2。

表 3-2ECC 和 RSA 在计算量上的比较

RSA 密钥大小 bit	ECC 密钥大小 bit	破解时间 MIPS 年
512	106	10^4
768	132	10^8
1,024	160	10^{11}
2,048	210	10^{20}

由表可知，ECC 使用的密钥比 RSA 中使用的密钥要短的多，在密钥长度相同时，ECC 与 RSA 所执行的计算量也差不多。因此，与具有同等安全性的 RSA 相比， ECC 所需要的计算量比 RSA 少。

(2) 计算量和处理速度：

在相同的计算资源条件下，虽然在 RSA 中可以通过选取较小的公钥(可以小到 3)的方法提高公钥处理速度，即提高加密速度，使其在加密速度上与 ECC 有可比性，但在私钥的处理速度上(解密)，ECC 远比 RSA 快得多。因此 ECC 总的速度比 RSA 快得多，同时 ECC 系统的密钥生成速度比 RSA 快百倍以上。因此在相同条件下 ECC 则有更高的加密性能。

(3) 存储空间占用：

ECC 的密钥尺寸和系统参数与 RSA 相比要小得多。160 位 ECC 与 1024 位 RSA 具有相同的安全强度，210 位 ECC 则与 2048 位 RSA 具有相同的安全强度。这意味着 ECC 所占的存储空间要小得多。这对于加密算法在资源受限环境上的应用具有特别重要的意义。

(4) 带宽要求：

当对长消息进行加解密时， ECC 与 RSA 密码系统有相同的带宽要求，但应用于短消息时 ECC 带宽要求却低得多。而公钥加密系统多用于短消息，例如用于数字签名和用于对对称系统的会话密钥传递。带宽要求低，使 ECC 在无线网络领域具有广泛的应用前景。

可见，ECC 无论从安全性，效率，实用性上都优于 RSA 算法，所以在混合加密体制中引入 ECC 算法肯定优于引入 RSA 算法。

通过比较可以看出，AES 和 ECC 相结合的混合加密体制，结合了对称密码体制和非对称密码体制的优点，在速度和效率上优于公钥加密体制，同时克服了对称加密算法中密钥管理困难，安全性低的问题，使两种算法应用更加广泛。同时，AES 和 ECC 相结合的混合加密体制无论在安全性还是在效率和实用性上都优于 DES 和 RSA 相结合的混合加密体制。所以，该混合加密体制是一种优秀的加密体制，必然有广泛的应用前景。

3.4.3 AES和ECC相结合的混合密码体制的描述

对称加密算法具有速度快、强度高、便于实现等优点，尤其适合加密大块数据，但密钥分配与管理比较困难，而非对称加密加密算法具有密钥分发与管理简单、速度慢等特点，一般用于加密少量数据、如传输密钥、数字签名等。我们将对称加密算法(AES)和非对称加密算法(ECC)混合使用，得到的混合加密体制，即基于 AES 与 ECC 的混合密码体制，既可有效地提高效率，又使网络传输更安全。

其实现的具体流程如下：

1．密钥的产生

使用 ECC 算法生成私钥和公钥，具体过程为：

(1)选取椭圆曲线参数 a 和 b，根据 $E_P(a, b)$选取点 G (x, y)，其中 G 的

阶为 n(n 为一个大素数)；

(2)在[1, n-1]上随机的确定一个整数 K_S，确定点 K_P，使得 $K_P=K_SG$。从而确定了密钥对(K_S, K_P)，其中 K_S 为私钥，K_P 为公钥。

2．加密过程

明文的加密：采用 AES 加密算法对明文进行加密生成密文 c。

AES 密钥的加密：采用 ECC 加密算法加密 AES 密钥生成密钥密文。K_A 为 AES 密钥，发送方 A 取随机数 r，其中 r∈{1, 2, ……, n-1}，接收方 B 的密钥对为 (K_{BS}, K_{BP})。根据密钥对(K_{BS}, K_{BP})得 $u=rK_{BP}$，其中 K_{BP} 为 B 的公钥。$R_1=rG=(x_1, y_1)$，$v=x_1K_A$。产生二元组(u,v)。将二元组传给接收方 B。

3．解密过程

AES 密钥的解密，由$(x_1, y_1)=K^{-1}u$ 得 x_1，其中 u 为二元组(u,v)中的 u，K_{BS} 为 B 的私钥。再由 $K_A=x_1^{-1}v$ 得到的 AES 密钥 K_A，其中 v 为二元组(u,v)中的 v。密文解密，经过 AES 密钥的解密，得到的 AES 密钥 K_A，利用 AES 算法得到明文 m。

4．数字签名过程

在签名生成时，发送方 A 取随机数 s，其中 s∈{1,2,……n-1}，(n 为密钥产生时的大素数)计算 $R_2=sG=(x_2, y_2)$，$e=x_2H(m)$，(其中 H(m)公开消息摘要函数计算得到的明文消息摘要)$k=s+eK_{AS}$，其中 K_{AS} 为发送方 A 的私钥。w=kG，从而得到二元组(w, e)。将二元组(w, e)作为发送方 A 对消息的签名。

在身份验证过程，根据 $R=w-eK_{AP}=(x_r, y_r)$，如果其中的 e 满足 $e=x_rH(m)$，则表示签名有效，否则签名无效。

整个体制数据的发送和接收过程如图 3.5 和图 3.6 所示。

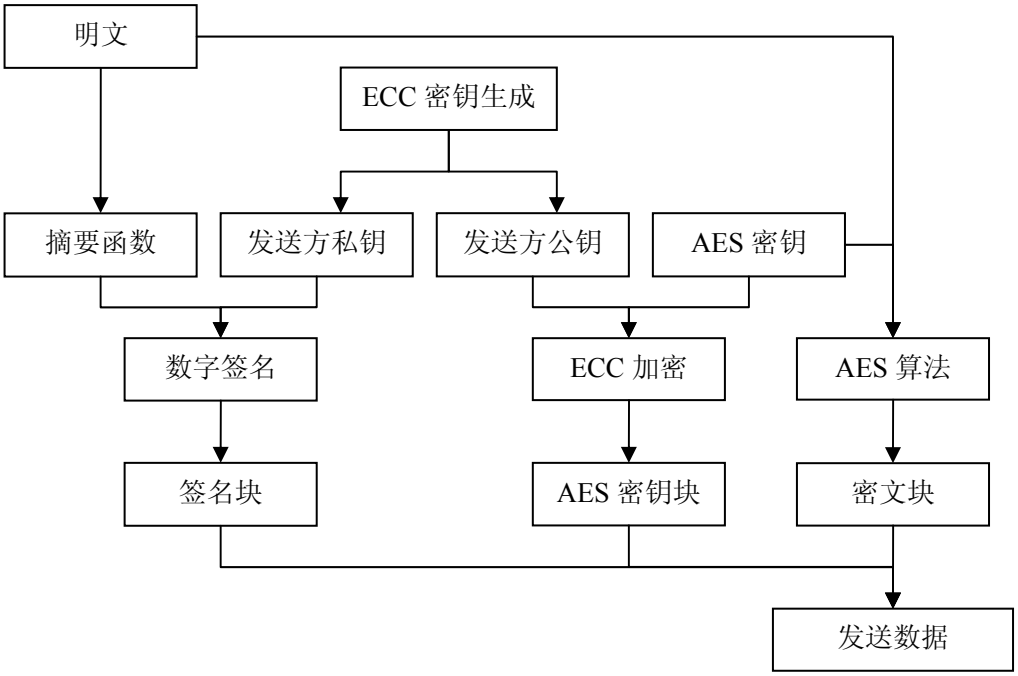

图 3.5 数据发送过程

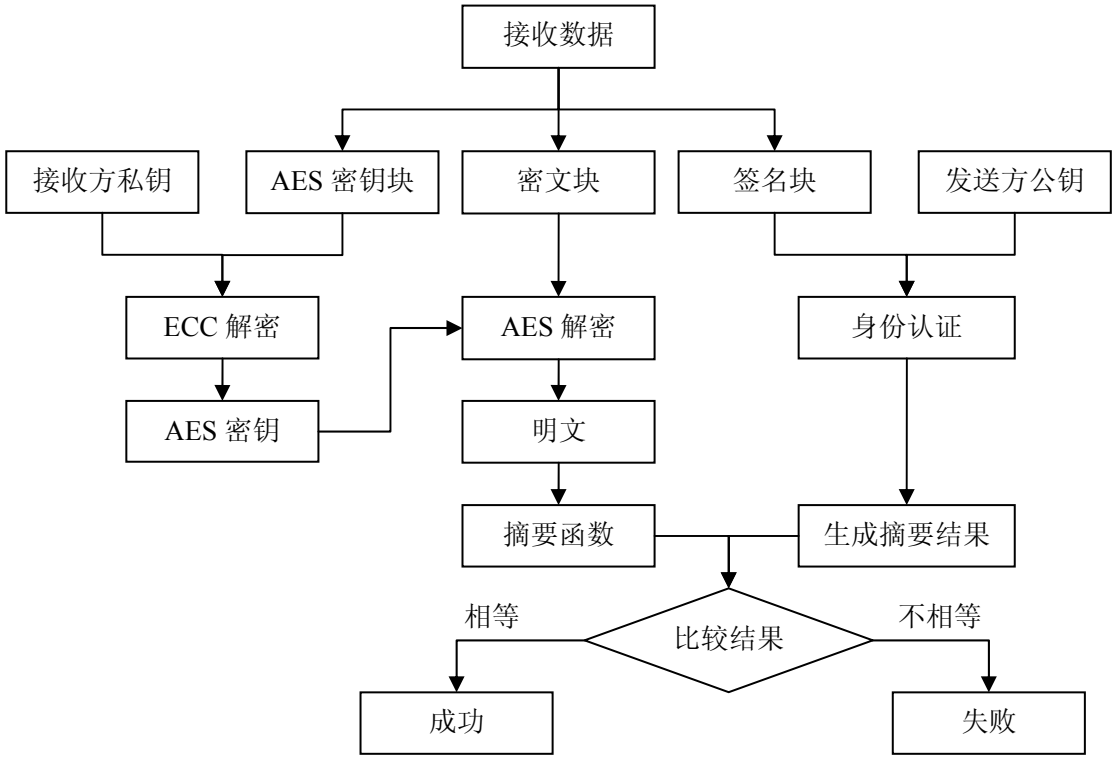

图 3.6 数据接收过程

3.5 本章小结

本章是整篇论文的核心部分。首先，研究了 AES 和 ECC 算法的数学原理和实现方法；然后，提出了 AES 和 ECC 相结合的加密体制；最后，通过对该混合加密体制与其他加密体制的比较，得出该混合加密体制是一种安全、高效、实用的加密体制，有着广泛的应用前景。

第四章 硬件电路设计

硬件设计主要包括 EZ-USB FX2 CY7C68013 外围电路设计、DSP 外围电路设计、CY7C68013 与 DSP 的连接方式设计和电源电路设计等。

4.1 电路总体框架

硬件通信模式搭建基于 PC、USB、DSP 的主从式系统，当系统上电完成各模块固件配置和硬件初始化后，可由 PC 发出控制命令，以外部信号触发 DSP 进入相应的中断处理程序，自动完成数据流 USB 下载、DSP 处理和 USB 回传的多次循环过程，PC 显示处理后的结果。如图 4.1 所示。

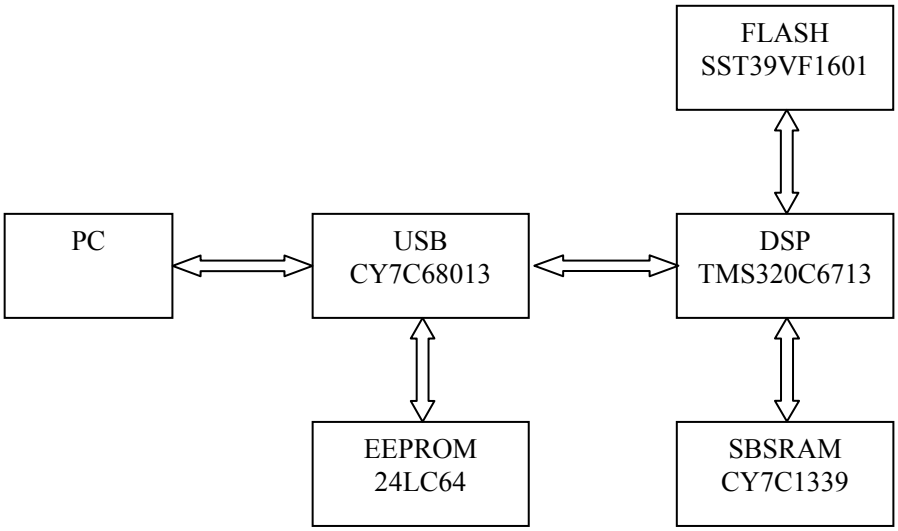

图 4.1 系统结构框图

4.2 USB外围电路设计

4.2.1 USB2.0 控制芯片介绍

随着USB的应用日益广泛，各个开发商也相继推出了各自的符合USB相应协议的USB控制器芯片，尽管各种芯片都是严格遵循USB的相关协议，但不同的厂商推出的产品还是有着一定的性能和用途差异，各种USB控制器芯片的结构可分成3种：

1．专为USB设计的芯片。这类控制器是为USB应用专门设计的芯片，能够使

USB的应用达到最优化。

2．与现有的芯片兼容。这类控制器芯片与现有的芯片兼容，这样开发者已经熟悉现有的芯片结构和开发指令，因此开发起来会比较容易。最常见的USB控制器都是与8051微处理器兼容。

3．需要外部微处理器接口的芯片。这类USB控制器只处理USB通信，而且必须由外部的微处理器来控制，因此这类控制器需要两个芯片，而其他种类的USB控制器则只需一个芯片(MCU和USB控制器在同一个芯片上)。

4.2.2 USB控制芯片选型

根据设计的需要，在此选用的USB控制芯片是EZ-USB FX2系列的CY7C68013(以下简称68013)芯片。该芯片是针对USB 2.0的，而且和USB 1.1兼容，它支持两种传输速率：全速(Full speed)12Mbps和高速(High speed)480Mbps，该芯片包括带8.5KB片上RAM的高速8051单片机、4KB FIFO存储器以及通用可编程接口(GPIF)、串行接口引擎(SIE)和USB2.0收发器，无需外加芯片即可完成高速USB传输[27]。

68013其芯片固件存贮在主机上而不是芯片内部，显著特点是代码升级容易。为了满足不同用户的需要，Cypress公司为68013提供了4种封装形式，分别为：128-pin TQFP、100-pin TQFP、56-pin QFN、56-pin SSOP，这些同种类不同封装的芯片内部结构是相同的，只是不同的封装形式引出的外部引脚数量有所不同[28]。考虑到本系统所需要的接口情况，本设计中选用的是100-pinTQFP封装的68013，它的组成部分主要包括：USB2.0收发器、SIE(串行接口引擎，Seria l Interface Engine)，带8.5KB RAM的增强型8051，4KB的FIFO存储器、I/O口、数据总线、地址总线和通用可编程接口(GPIF)。

68013拥有独特的结构，其串行接口引擎(SlE)负责完成诸如数据的编解码、差错控制、位填充等与USB协议有关的功能，它将嵌入式MCU(增强型8051)解放出来，简化了固件代码的开发[29]。68013中还包含一个通用可编程接口(GPIF)，它支持所有通用的总线标准，如ATAPI(PIO和UDMA)、IEEE 1284(EPP并行口)和UTOPIA等，并可与外部ASIC，DSP等直接连接，如图4.2所示。

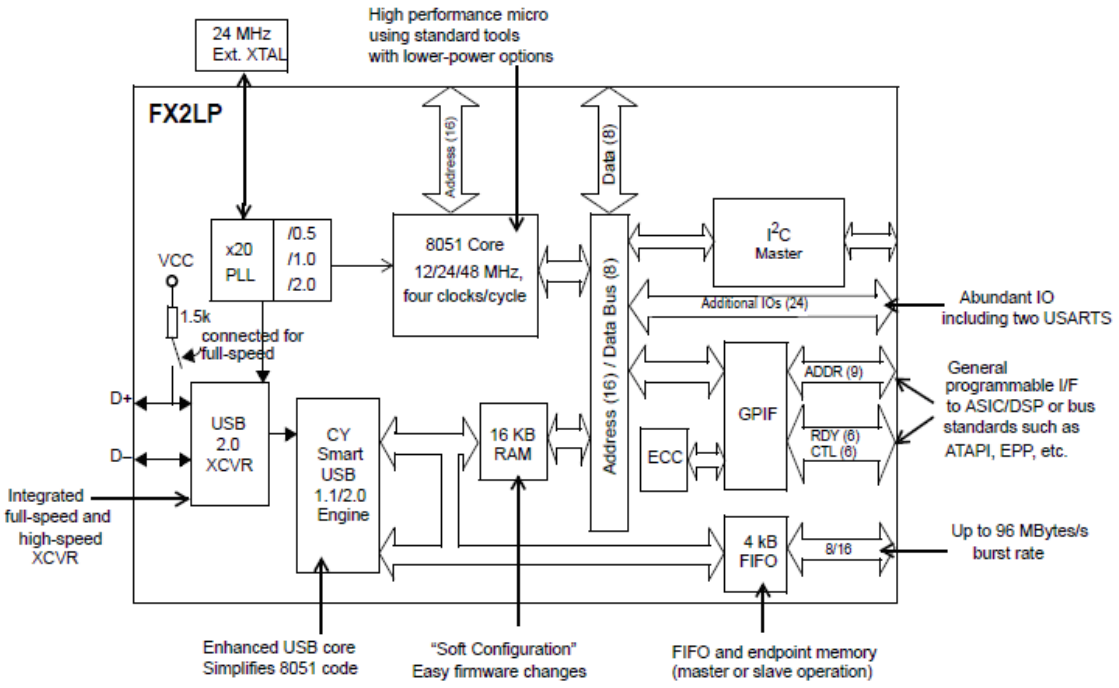

图4.2 68013内部结构框图

4.2.3 USB电路设计

1．68013列举模式

列举是USB设备的一个非常重要的机制。是在初始阶段必须经历的阶段，只有列举完成了，USB设备才能实现具体功能。

当68013上电并脱离复位状态后，首先检查芯片的I^2C总线上是否连接有串行EEPROM，如果存在EEPROM，将读取第一个字节以决定其列举模式[30]。根据是否存在EEPROM及其首字节的内容，可分为三种列举模式：无串行EEPROM、EEPROM的首字节为0xC0(“C0加载”)、EEPROM的首字节为0xC2(“C2加载”)。本设计中采用C2加载模式，C2加载模式数据格式如表4-1所示。

表4-1 68013的列举模式

EEPROM的首字节	不是0xC0、0xC2	0xC0	0xC2
68013内核的操作	68013内核提供USB描述符、PID/VID/DID，并设置RENUM=0	68013内核提供USB描述符，EEPROM提供PID/VID/DID，并设置	加载EEPROM内容至68013的RAM，并设置RENUM=1，由8051提供

本设计中采用C2加载模式，设计一个EEPROM用以保存68013的固件程序，并且第一个字节为0xC2，通过I^2C总线连接到68013。在这种加载模式下，68013脱离复位后包含VID/PID/DID码值的固件程序将自动从EEPROM下载到68013芯片上的

RAM运行。该模式下RENUM位自动设置为l，由该固件程序来处理随后的设备请求。

I^2C总线控制器使用两引脚SCL和SDA进行控制。SCL(Serial Clock)为串行同步时钟，而SDA(Serial Data)为数据线。给每个连接到总线的设备分配唯一的地址。该总线是真正的多主机总线，具有冲突检测，和当多个主机同时通过总线传输数据时，防止破坏数据的仲裁机制。总线是双向串行的，一次传输8位数据，标准速率为100kbps，快速模式为400kbps，及高速模式3.4Mbps。一次可连接到I^2C上的外设数量受最大的400pF电容的限制。

每一设备被唯一地址所标识，这是由设备地址和端点号给出的。并且既可以作为发送者也可以作为接收者。主机是初始化总线上数据传输，并产生时钟信号，接收地址的一方为Slave。

SDA和SCL是双向线，当空闲时，读总线为高电平。连接到总线上的输出阶段必为漏极开路(Open-Drain)和集电极开路(Open-Collector)以完成线与(Wire-And)功能。I^2C上的数据可以标准模式l00kbps，400kbps的快速传输模式，以及3.4Mbps的高速模式传输。在本设计中SDA和SCL有2.2K的外部上拉电阻，外部EEPROM设备地址引脚必须适当配置，如表4-2所示。

表4-2 外部EEPROM设备

字节数	EEPROM举例	A2	A1	A0
16	24LC00	无效	无效	无效
128	24LC01	0	0	0
256	24LC02	0	0	0
4K	24LC32	0	0	1
8K	24LC64	0	0	1

由于68013具有I^2C的结构特点，本设计采用的EEPROM芯片选用的是Microhip公司研制的具有I^2C总线接口的E2PROM芯片24LC64，其容量为64KB，可重复擦写100万次，掉电状态下数据可以保存多达100年(甚至200年)。24LC64功耗极小，工作电压为2.5-5.5V，读操作时的最大工作电流约400μA。24LC64一次可写入32字节，且可任意或连续的读出8K字节数据。同时，由于具有I^2C总线接口可以很简单的实现和68013的电路连接。具体电路连接如图4.3所示。

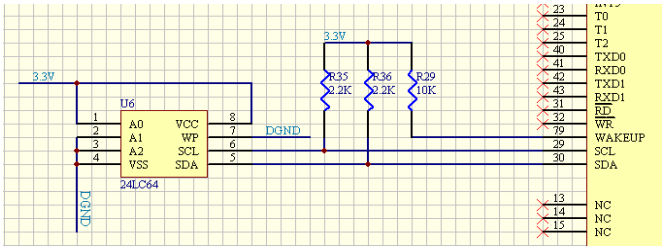

图4.3 I^2C接口设计

2．时钟电路设计

68013具有一个使用外部24MHz晶体(±100 ppm)的在片振荡器线路，它具有如下特性：并行谐振。基本模式。500微瓦驱动电平。20-33 pF (误差5%) 负载电容。在片锁相环将24MHz 振荡器倍增到480 MHz, 满足收发器/PHY需要, 同时内部计数器将它向下分频作为8051的时钟。默认的8051时钟频率为12 MHz。8051的时钟频率可以通过CPU的CS寄存器动态地修改。通过内部的控制比特可以将CLKOUT引脚置为三态或反向，该引脚在8051的时钟频率48, 24, or 12 MHz上以50%的占空比输出。配置异步从FIFO(Asynchronous Slave FIFO)模式，采用24M赫兹的时钟源。如图4.4所示。

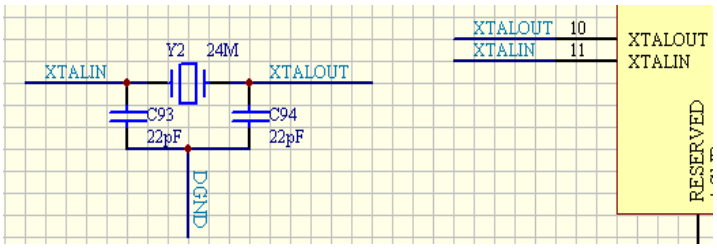

图4.4 时钟设计

3．端点缓冲区设计

由于USB是一个串行总线，所以设备端点实际上是一个用USB数据字节不断取空盒填满的FIFO。所有的传输都是传送到一个设备端点，或是由一个设备端点发出。在USB的规范中，设备端点可以定义为：“一个USB设备中的惟一可寻址部分，是主机与设备之间通信的来源或目的”。当USB设备连接到主机时，系统为该设备分配一个唯一的地址，设备用端点号标识端点，这个端点号也是惟一的。主机通过发送4位地址和一个方向位来选择端点，所以USB能够选择32个具有特定地址的端点(IN0-IN15以及OUT0-OUT15)，分为批量端点、中断端点、同步端点和控制端点，分别对应批量传输、中断传输、同步传输和控制传输4种传输类型。68013从OUT端点缓冲区读主机发出的数据，并向IN端点缓冲区写要传输给主机的数据。

68013芯片内部包含3个固定的64字节端点缓冲区(0xE740-0xE7FF)和4KB的大端点可配置端点缓冲区空间(0xF000-0xFFFF)，如图4.5所示。

图4.5 片内数据存储区0xE000-0xFFFF

68013的端点具有以下特点：

(1)量子FIFO结构，CPU无需参与数据传输；

(2)分为大小端点缓冲区，大端点有若干重缓冲；

(3)大端点可以配置缓冲区大小和类型。

其中，3个64字节缓冲区分别用于EP0、EP1IN和EP1OUT，只能被CPU存取，不能直接与外部设备连接，4KB的大端点缓冲区用于EP2、EP4、EP6和EP8，可以配置其缓冲深度(2，3或4)。 EP4和EP8固定为512字节的双缓冲，而端点EP2和EP6配置较灵活，可为512或1024字节的双、三或四缓冲。大端点缓冲区的数据一般由FIFO接口控制，当然CPU也可以存取这些4KB的大端点缓冲区。图4.6的每一列均代表一种配置方式。

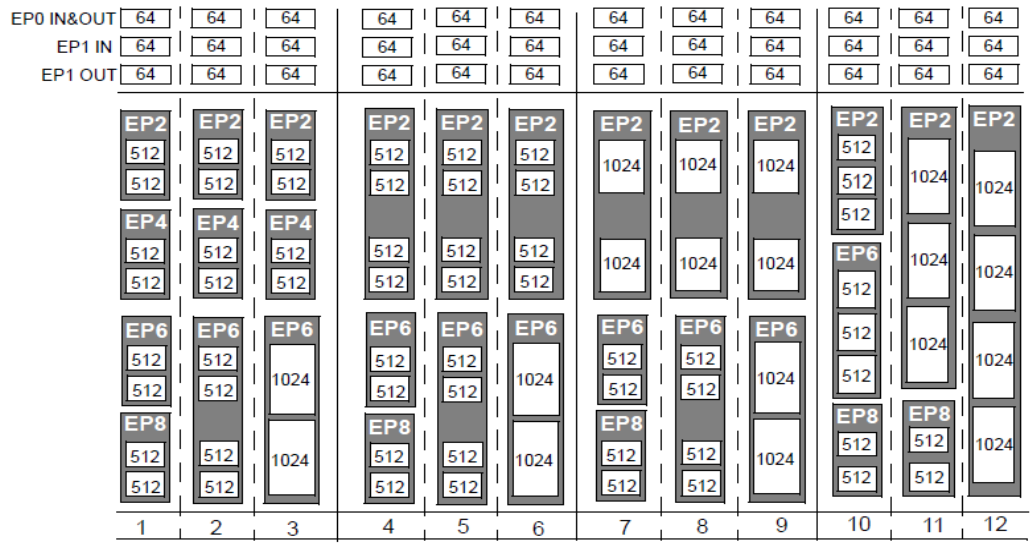

图4.6 68013端点缓冲区配置方式

端点0默认为控制端点，其OUT和IN数据共享一块存储空间，是惟一的控制传输的双向端点；端点1可配置为中断和批量端点，EP1IN和EPlOUT占用不同的缓冲区；EP2、EP4、EP6和EP8是大容量高带宽的数据传输端点，无需CPU参与即可同外围电路完成高速数据传输，可以配置为中断、批量或同步端点。各端点具体的可配置情况如图所示，图中每一列代表一种配置方式，其黑体方框中包含2个、3个或4个512B(或1024B)的数据缓冲区，它们分别表示端点可配置成双重缓冲、三重缓冲和四重缓冲。多重缓冲结构可以在数据读写双方速度相似时有效的提高USB带宽的性能，平滑带宽抖动，并减少双方的等待时间。

地址在0xE6B8-0xE6BF 的独立的8字节缓冲器保持控制传输的设置数据。EndPoint4和EndPoint8作为双向传输的管道，分别对应缓冲FIFO4和FIFO8存放USB需要接收与下传的数据，它们均采用批量(BULK)传输方式，如表4-3所示。

表4-3 数据传输

交替设置	**0**	**1**	**2**	**3**
EP0	64	64	64	64
EP1OUT	0	512批量	64中断	64中断
EP1IN	0	512批量	64中断	64中断
EP2	0	512批量输出(2x)	512中断输出(2x)	512同步输出(2x)
EP4	0	512批量输出(2x)	512批量输出(2x)	512批量输出(2x)
EP6	0	512批量输入(2x)	512中断输入(2x)	512同步输入(2x)
EP8	0	512 批量输入(2x)	512 批量输入(2x)	512 批量输入(2x)

在本设计中采用设置FIFO4、FIFO8为自动方式，即在数据传输过程中无需68013的8051内核参与，以保证持续、高速、有效的数据传输。

4．复位电路设计

图4.7是本设计中的复位电路设计，输入引脚(RESET#)复位芯片，该引脚具有磁滞作用且为低有效，其内部的锁相环在VCC达到3.3V后稳定约200微秒。为了使芯片初始化正确，本文使用外部RC网络(R =10千欧姆, C = 0.1微法拉)来提供，从而满足RESET#信号复位要求。

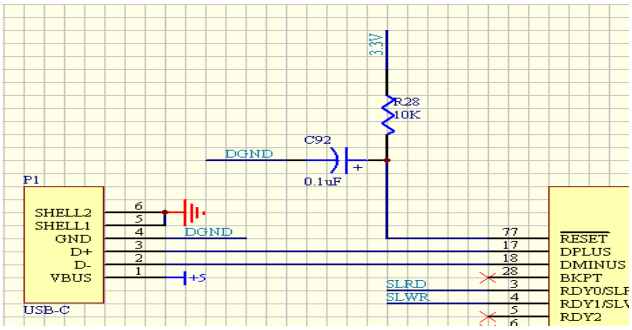

图4.7 RESET信号设计

综合以上各个部分的设计，USB电路设计具体如图4.8所示：

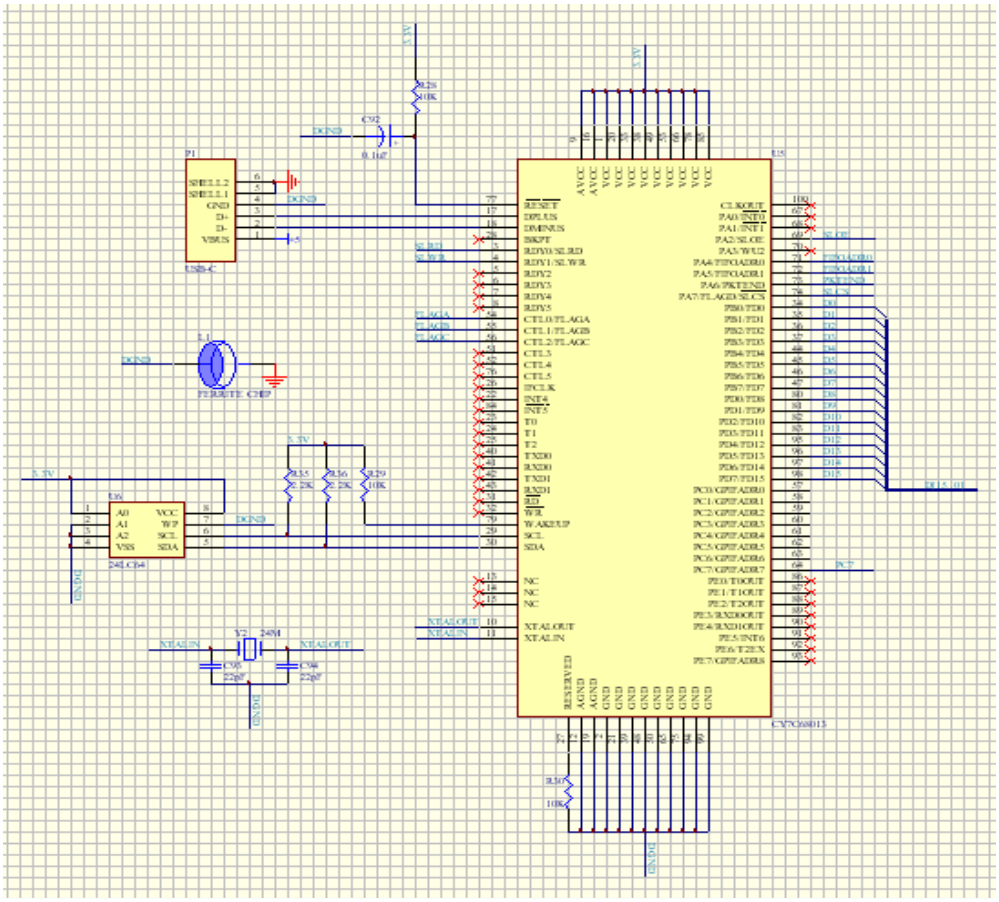

图4.8 USB模块总体设计图

固件就绪后通过Cypress公司提供的工具Control Panel加载至EEPROM中，当需要修改固件时，就可以在不改动硬件的情况下将主机上修改好的固件重新下载一次。

4.3 DSP外围电路设计

由于加密体制，运算量很大，所以 DSP 选片倾向于高性能定点 DSP，TI 的 TMS320C6000 系列的 DSP 比较符合需求。其中有单核和多核之分 TMS320C6000 系列是 TI 公司推出的运算能力最强的处理器。在本设计中采用 TMS320C6713(以下简称 C6713)芯片，它是 32 位定浮点 DSP，包含 CPU 和外设资源两大部分。采用 VLIW 的体系结构及流水线技术，具有两级 Cache 缓存结构，而且运行速度快，精度高。其工作主频可达 200 MHz，单指令执行周期仅 5 ns。片上共有 264 KB×8 位存储器，其中含有 4 KB×8 位 L1 P Cache，4 KB×8 位 L1 D Cache 和 256 KB×8 位 L2 RAM/Cache。片上外设资源丰富，其中含有两个多通道缓冲串口(McBSP,Multichannel Buffered Serial Port)、两组 I^2C(Inter-Intergrated Circuit)总线、两个 32 位通用定时器、一个 16 位主机接口(HPI,Host-Prot Interface)等。此外，

TMS320C6713 还有 32 位的外部存储器扩展接口(EMIF,External Memory Interface)总线，分为 4 个存储空间(CE0-CE3)，可访问 8 位、16 位或 32 位数据宽度，每个空间均可与 SDRAM，SBSRAM 以及 SRAM 等连接，如图 4.9 所示。

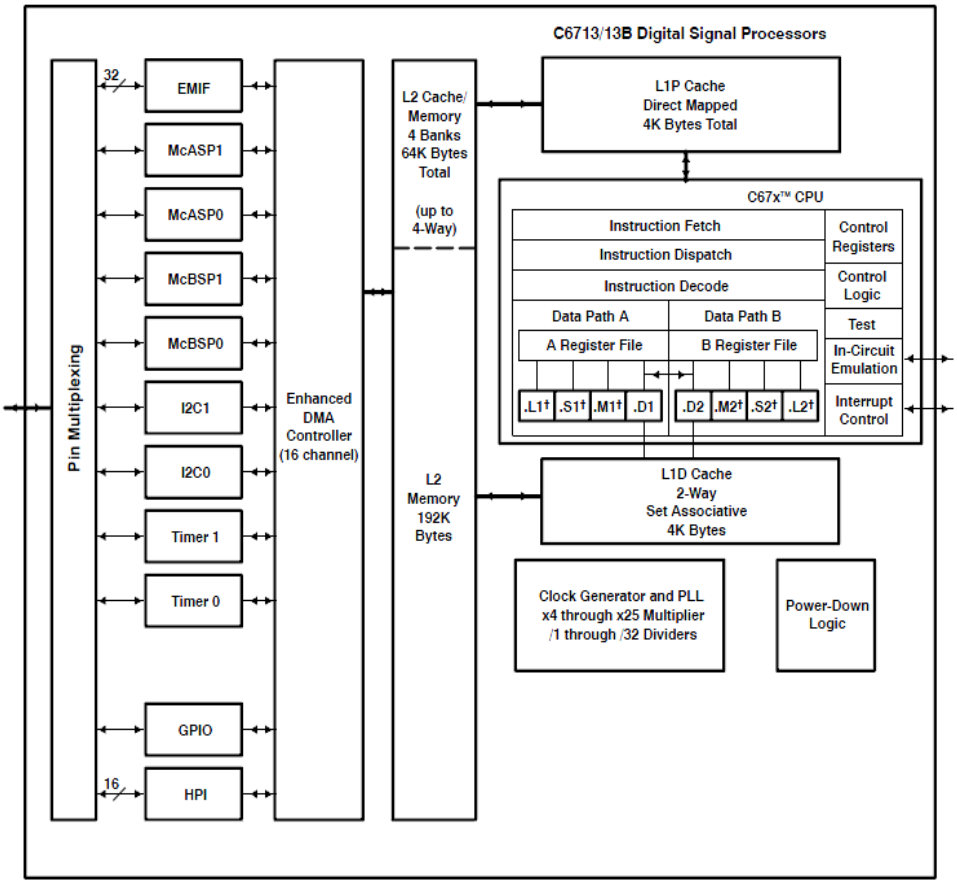

图 4.9 C6713 内部结构

4.3.1 C6713CPU结构

C6713CPU 采用哈佛结构，其程序总线与数据总线分开，取指令与执行指令可以并行运行[31]。程序总线宽度为 256bit，每一次取指操作都是取 8 条指令，称为一个取指包[32]。在执行时，每条指令占用 1 个功能单元，取指令、指令分配和指令译码都具备每周期读取并传递 8 条、32 位指令的能力。C6713 的 CPU 有 2 个类似的可进行数据处理的数据通路 A 和 B，其结构如图 4.10 所示，共包含下述物理资源：

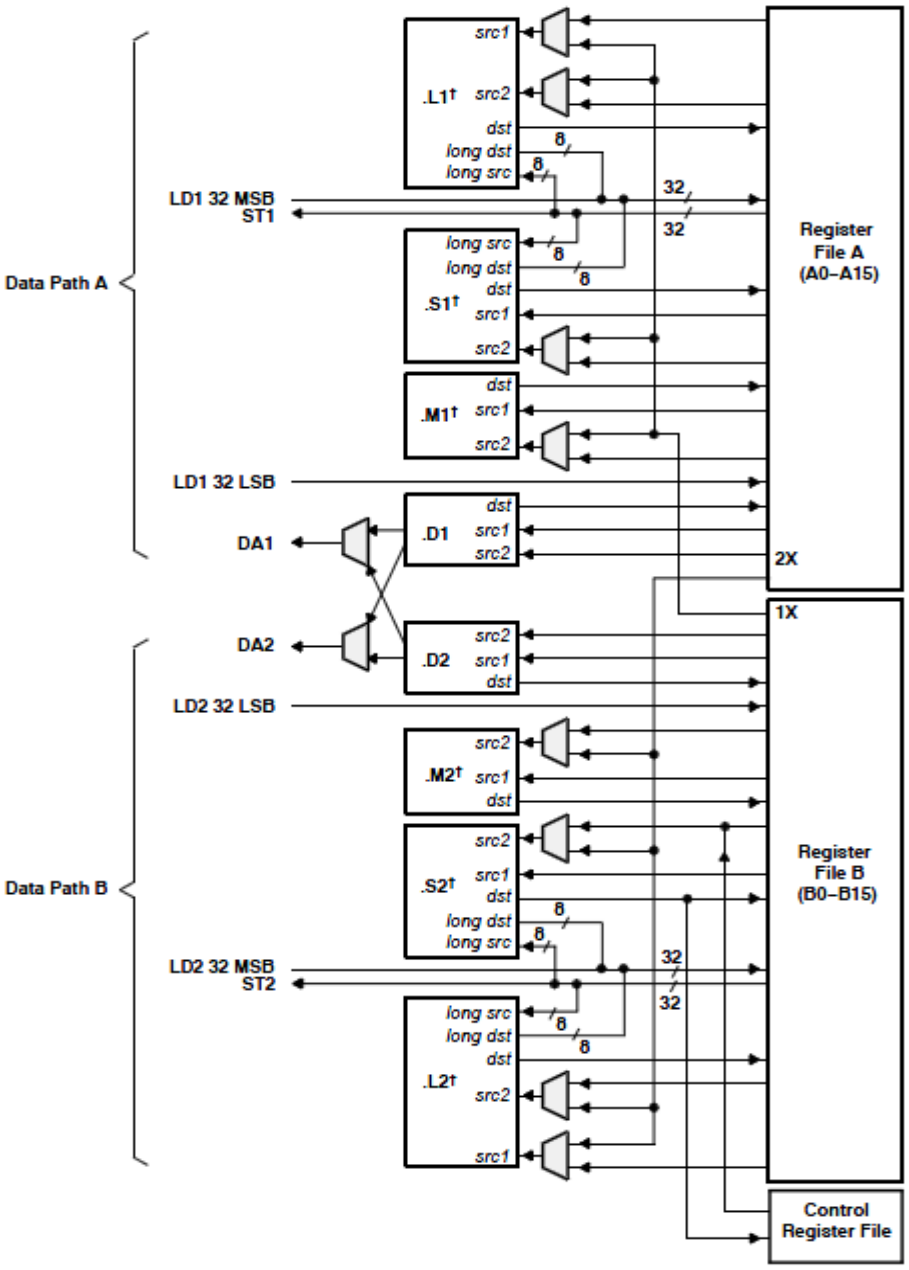

图 4.10 C6713 数据通路

1．2 个通用寄存器组(A 和 B)每个通用寄存器组包括 16 个 32 位寄存器，其作用是：

存放数据，作为指令的源操作数和目的操作数。

作为间接寻址的地址指针，寄存器 A4、A5、A6、A7 和 B4、B5、B6、B7 还可以以循环寻址方式工作。

A0、A1、A2、B0、B1 和 B2 可以当作条件寄存器。

2．8 个功能单元(.D1、.M1、.S1、.L1 数据通路 A 和.D2、.M2、.S2、.L2)数据通路 B

C6713 每组数据通路有 4 个功能单元，两组数据通路功能单元的功能基本相同。.D 单元是唯一能产生地址的功能单元，.M 单元主要完成乘法运算，.S 与.L 是

主要的算术逻辑运算单元(ALU,Arithmetic Logic Unit)。

3．2 个数据读取通路(LD1 和 LD2)和 2 个数据存储通路(ST1 和 ST2)

LD1 和 LD2 将数据从存储器读取到寄存器；ST1 和 ST2 则是将各组寄存器的数据存储到数据存储器中。

4．2 个寄存器组交叉通路(1X 和 2X)

1X 交叉通路允许数据通路 A 的功能单元从寄存器组 B 读取它的源操作数，2X 交叉通路则允许数据通路 B 的功能单元从寄存器组 A 读取源操作数。每个功能单元可以直接与所处数据通路的寄存器组进行读写操作，也可以通过 1X 和 2X 交叉通路与另一数据通路的寄存器组相连。

5．2 个数据寻址通路(DA1 和 DA2)

寻址通路与两侧数据通道都相连，这使一个寄存器组产生的数据地址能够支持任意一侧寄存器组对数据存储器的读写操作。

6．控制寄存器组

用户可以通过对控制寄存器组编程来选用 CPU 的部分功能。编程时应注意，仅功能单元.S 可通过搬移指令 MVC 访问控制寄存器，对其进行读写操作。C6713 的控制寄存器有：

寻址模式寄存器 AMR；

控制状态寄存器 CSR；

程序计数器 PCE1；

中断控制寄存器 IFR、ISR、ICR、IER、ISTP、IRP 和 NRP。

C6713 不仅具有很高的运算速度，而且在片内集成了许多外设，支持多种工业标准的接口协议，能够提供高带宽的数据 I/O 能力。这些特点使 C6713 获得了很强大的综合性能，其高集成度也给系统设计人员带来了很多方便。

由图 4.9 所示，C6713 的片内外设资源包括：1 个 EDMA 模块、2 个 I^2C 总线、1 个通用输入输出模块 GPIO、2 个 32 位通用定时器、1 个主机接口 HPI、1 个外部存储器接口 EMIF、2 个多通道缓冲串口 McBSP 以及 2 个多通道音频串口 McASP。

4.3.2 C6713 芯片配置

C6713 有一系列管脚用于芯片工作模式的设置，芯片复位时，首先检测这些管脚的输入电平，以决定 DSP 的时钟模式，Endian 模式以及引导模式等。

需要指出的是，C6713 的配置管脚并不都是专门用来进行配置的，而是与其它外设模块共用的复用管脚。另外，由于芯片体积和管脚数量的限制，还有一些管脚也是由不同外设共享的复用管脚，这就要求用户根据需要对外设进行必要的选

择。

4.3.3 C6713 引导方式

C6713 提供了 2 种引导方式：主机加载和外接 Flash(ROM)加载[33]。当选择主机加载模式时，核心 CPU 停留在复位状态，芯片其余部分保持正常状态。在引导过程中，外部主机通过主机的接口来初始化 CPU 的存储空间。当完成所有的初始化工作后，主机向接口控制寄存器 DSPINT 位写 1，结束引导过程。CPU 退出复位状态，开始执行位于地址 0 处的指令。主机加载模式下，可以对 DSP 所有的存储空间进行读/写[34]。

当选择 Flash(ROM)加载模式时，CPU 在复位信号撤销之后，仍保持复位状态。此时位于外部 CE1 空间的 FLASH 中的代码通过 EDMA 被传输到地址 0 处[35]。传输完成后，CPU 退出复位状态，开始执行位于地址 0 处的指令。用户可以指定外部加载 Flash 的存储宽度，EMIF 会自动将相邻的 8bit/16bit 数据合成为 32bit 的指令。Flash 中的程序存储格式应当与芯片的 Endian 模式设置一致。

在实际应用中，为了获得较高的运行速度，通常要把低速 FLASH 中的代码传送到高速 RAM 中执行，但大部分应用程序都要超出 1KB，显然上述的 FLASH 引导过程不能满足全部程序传输的需要，这就需要开发人员自己编写一段“二级引导程序”来完成剩下的传输工作。此时需要注意的是，“二级引导程序”要被放在 CE1 空间 FLASH 的起始处。整个 FLASH 引导方式的工作过程如下：

1) 设备复位，CPU 从 CE1 空间的起始处拷贝 1KB 数据到地址 0 处。所拷贝的这些数据就包含用户编写的二级引导程序。

2) 拷贝结束，CPU 退出复位状态，从地址 0 处开始运行二级引导程序。该引导程序按要求将 FLASH 中的应用程序拷贝到 RAM 的指定位置[36]。完成后，引用 C 程序入口函数 c_int00()。

3) c_int00()函数初始化 C 语言运行环境，然后开始运行应用程序。

4.3.4 C6713 电路设计

1．时钟电路设计

为DSP提供时钟有两种方法：一是利用DSP芯片内部的晶振电路，与无源晶振、起振电容一起连接组成振荡器来产生时钟；二是采用现成封装好的晶体振荡器，将外部时钟直接连接到CLKIN管脚[37]。第一种成本低，而第二种产生的时钟信号更稳定一些，在本设计中，由于时钟要求比较严格，所以采用第二种方法。如图4.11所示。

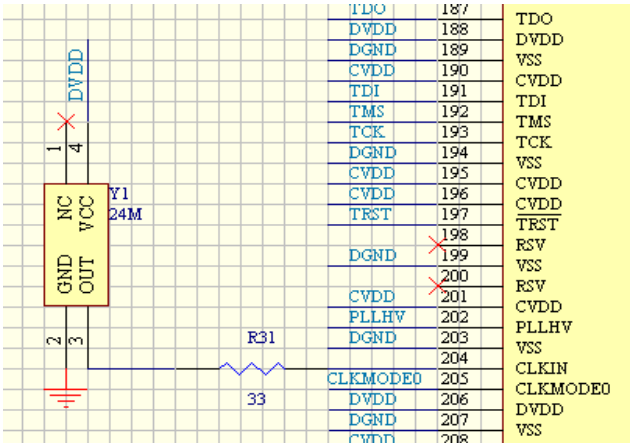

图4.11 时钟电路设计

2．存储空间设计

存储器是 DSP 系统中最重要的部件之一，从使用功能的角度，存储器可分为断电后数据丢失的易失性存储器；断电后数据不会失去的非易失性存储器。在传统的观念上前者叫做 RAM，包括 SRAM(静态 RAM)和 DRAM(动态 RAM)：后者叫做 ROM，这种存储器只能脱机写入，使用时只能读不能写。SRAM 是基于触发器原理的，读写速度快，但集成度低，成本高，功耗大[38]。由于在本设计中采用加密体制平台系统中的高速数据要求，所以采用的是当前应用比较广泛的 SBSRAM。

C6713 为哈佛结构的 DSP，其存储器结构使用了两级 Cache 结构，第一级 Cache 由 4KB 直接映射的程序 Cache (L1 P)和 4KB 的两路组关联数据 Cache (L1 D)组成；第二级片内存储器(L2)为 256KB 大小的程序和数据共享存储区。L1 P 与 CPU 之间有一个 256 位宽度的通道，可以支持最大 8 个 32 位的指令同时读取；L1 D 允许对 CPU 的两组数据通道同时访问。在 L1 D 和 L2 之间有一条 64 位宽度的写总线。C6713 的 L2 是一个统一的 256KB 程序和数据存储空间，其中 192KB 直接映射到存储空间，另外的 64KB 空间可以灵活配置成二级 Cache/SRAM，其有 5 种工作模式，分别为：256KB 的 SRAM、16KB 的 Cache 和 240KB 的 SRAM、32KB 的 Cache 和 224KB 的 SRAM、48KB 的 Cache 和 208KB 的 SRAM、64KB 的 Cache 和 192KB 的 SRAM。在实际应用时，系统设计者可以根据自己的需求，灵活配置二级 Cache 的大小。

C6713 的整个存储空间分配，如图 4.12 所示。

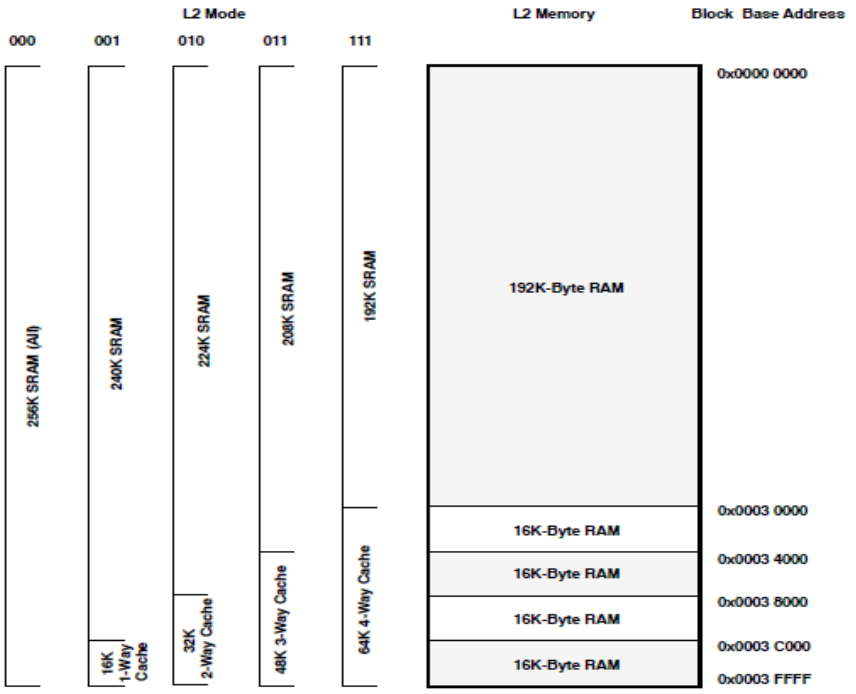

图 4.12 内部存储结构

由于本系统为实时数据加密系统，数据量会特别大，C6713 自带的存储空间显然无法满足要求，故在 CE0 空间连接一块容量为 128K×32 的 SBSRAM(CY7C1339)以扩充容量。

SBSRAM 是一种同步突发式静态存储器，其最大的优点在于读写速度高，而且是静态 RAM，不需要刷新，EMIF 接口提供了标准的 SBSRAM 的支持。具体的设计如图 4.13 外部存储器设计。

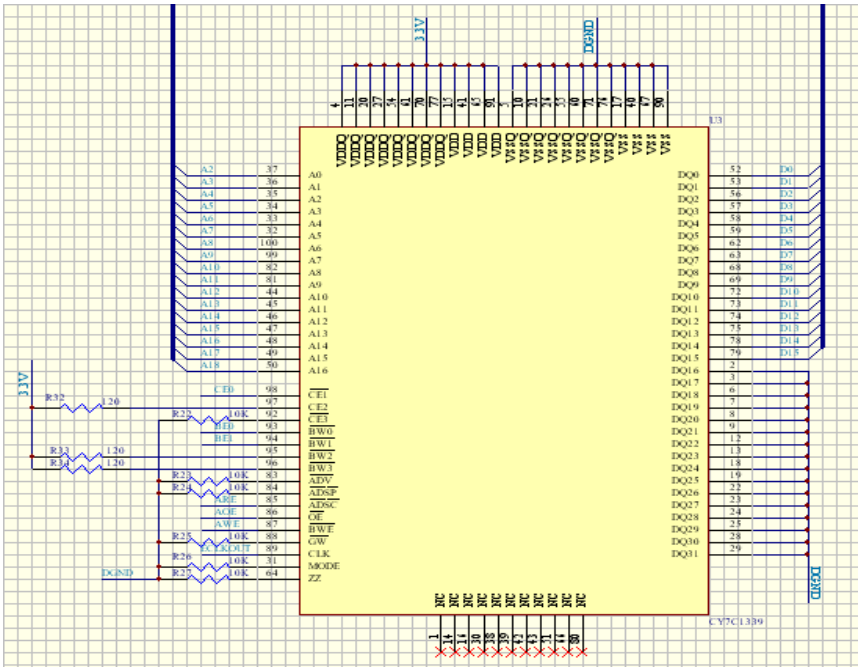

图 4.13 外部存储器设计

3．Flash 模块设计

首先要介绍存储器宽度和字节对齐。C6713 能直接和 8/16/32 位存储器无缝接口，其内部以字节进行编址，外部存储器地址，由 EMIF 根据所接口的存储器的宽度，自动对逻辑地址进行移位产生，如图 4.14 所示。

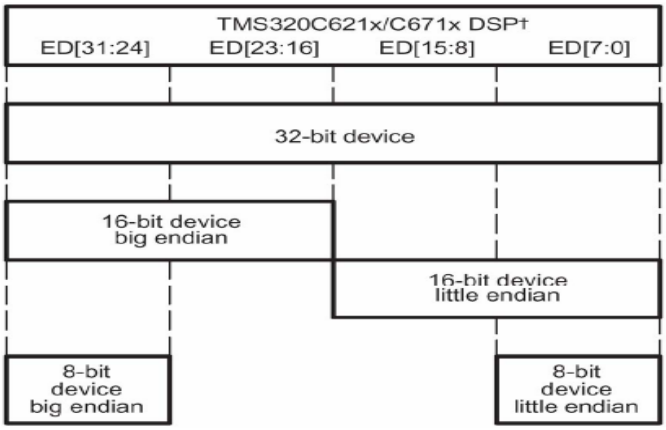

图 4.14 存储器字节对齐方式

由于 DSP 内部不带 Flash，所以必须在外部连接 Flash，这样下电时程序就存放在 flash 中。在 C6713 的 EMIF 的 4 个 CE 空间中，程序只能从 CE1 空间启动，故本设计使用的 Flash (SST39VF1601,1M×16)接在 CE1 空间中。将 C6713 的 HD8 引脚上拉，选择 Little Endian 模式。将 HD12 上拉，设置数据出现在 ED[7:0]，或下拉忽略 Endian 制式。CLKMODE0 引脚必须上拉，选择使用外部晶振的时钟输入。HD4 引脚上拉，HD3 引脚下拉，即配置为 10 的模式，选择 C6713 从外部异步 ROM 启动引导系统，如图 4.15 所示。

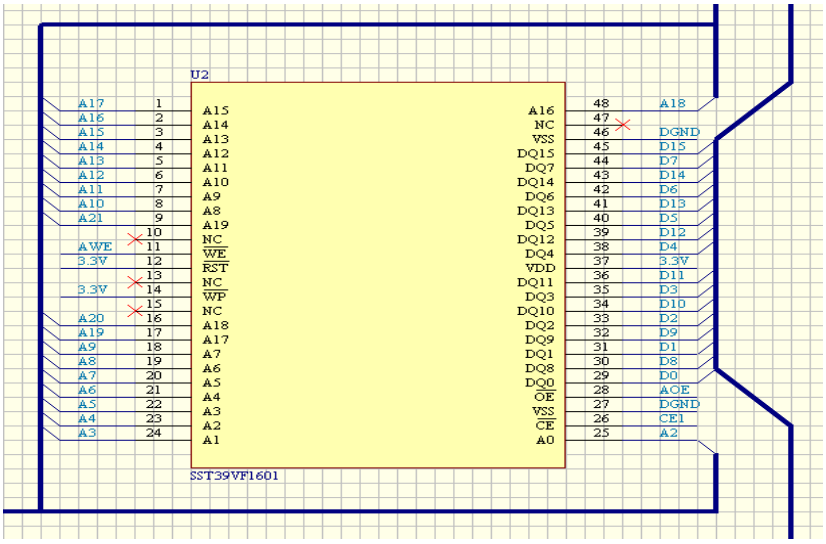

图 4.15 Fash 设计

Flash 的选择，主要考虑以下几个方面的因素：

(1)可靠性：Flash 一般选择可重复擦写 10 万次，保存时间在 100 年左右。

(2)容量：根据系统要求。

(3)和 DSP 芯片的兼容性：包括读写速度、电压、时钟以及硬件连接等方面，一般选择和 DSP 能达到最佳配合的芯片。

4．JTAG 仿真接口

一个 DSP 应用系统为了方便的进行软硬件仿真和调试，一般都要设计一个仿真器接口，以便于系统与仿真器连接。JTAG(Join Test Action Group)是基于 IEEE 1149.1 标准的一种边界扫描测试方式，为方便仿真器和 DSP 目标板连接以进行仿真和调试，TI 公司的大部分 DSP 芯片都设计了 JTAG 仿真接口，JTAG 仿真接口也是调试过程中装载数据和代码的唯一通信通道，C6713 芯片也内嵌了一个遵循 IEEE 1149.1 标准的 JTAG 仿真接口模块。

目前，市场上常用的 JTAG 仿真器都符合 IEEE1149．1 标准，例如合众达电子公司的 XDS510、XDS510USB、XDS560 等以及闻亭公司的 TDS560USB、TDS510USB 都满足该格式，可方便的选用。JTAG 仿真接口是一个标准的 14 针接口，在 JTAG 中 l3 和 14 引脚 EMU0 和 EMUl 是仿真信号引脚，为保证仿真器和 DSP 目标系统之间仿真信号的质量，在这两个引脚设计上拉电阻以增强驱动能力，JTAG 接口电路如图 4.16 所示。

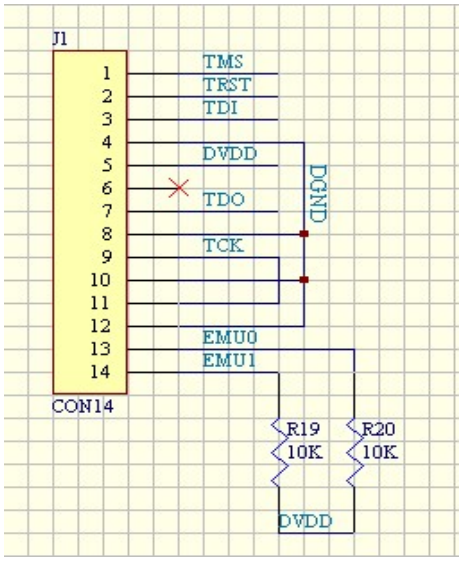

图 4.16 JTAG 接口电路

5．电源和复位电路设计

低功耗是以后发展的方向理念，近年来推出的 DSP 芯片大部分采用低电压供电方式和低功耗工作模式以降低芯片功耗。主控 DSP 芯片 C6713 核心电压为 1.2V，外围电压为 3.3V，外部时钟频率为 24MHz。由于实际常用的只有 5V 电压，所以必须选用电压转换芯片。这里本设计选用 TI 公司专门设计的 TPS70451 电源芯片，在这里需要注意的是在满足供电电压要求的同时还必须满足供电的上电和下电次序，上电时首先核心电压上电然后外围电压上电，下电时同样也是核心电压下电后外围电压下电，具体电源设计如图 4.17 所示。

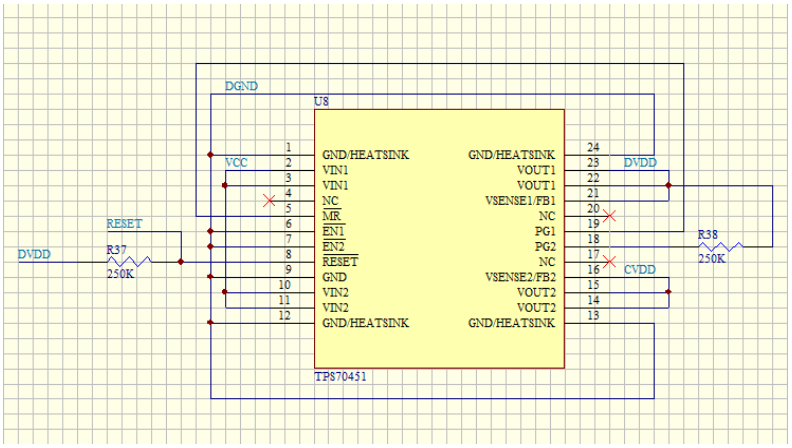

图 4.17 电源设计

4.4 USB与DSP连接方式设计

68013 提供了两种接口模式：从属 FIFO(Slave FIFOs)和通用可编程接口(GPIF,General Programmable Interface)。本设计中采用 Slave FIFOs 方式，DSP 作为主控制器，向 68013 发送控制信号。另外，对 68013 的 FIFO 读写，本文选择了异步模式。其最大的特点就是外部控制器(在这里指 DSP 芯片)读写 68013 的多层缓冲 FIFO 与读写普通 FIFO 一样，即可以实现对 68013 的端点 2、端点 4、端点 6 和端点 8 的数据缓冲区的方便读写。外部逻辑或外部控制器直接连接 68013 的 FIFO 端点，根据具体情况设定为同步或者异步，工作时钟可选为内部产生或外部输入，其它控制信号也可灵活的设置为高有效或低有效[39]。

实时加密系统平台上的 USB 设备是符合 USB2.0 协议标准。将 68013 配置在 C6713 的 EMIF 的 CE2 接口上。图 4.18 为 Slave FIFOs 模式下 68013 和 DSP 主控制器的典型连接电路原理图。

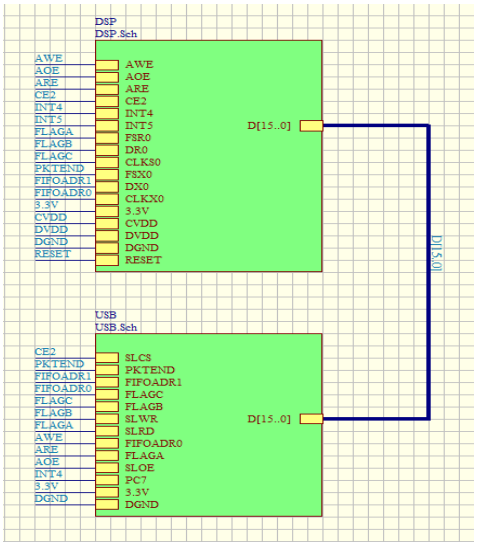

图 4.18 USB 与 DSP 连接方式

68013 的 16 位 FIFO 数据线和 DSP 的数据线直接相连，传输为双向的；采用 FIFOADR[1:0]引脚选择端点 FIFO 的地址：EP2、EP4、EP6、EP8；IFCLK 为接口时钟，可由芯片内部产生，也可由外部输入，本设计采用 48MHz 的外部时钟；SLCS# 为 Slave FIFOs 的片选信号，低电平时有效；SLOE 为输出使能控制引脚；SLRD 和 SLWR 分别作为 FIFO 的读和写选通信号；PKTEND 是包强制结束控制引脚，当 DSP 要向 FIF0s 写入短数据包时控制器使能 PKTEND 管脚，使其不考虑该包的长度；FLAGA-FLAGD 为 FIFO 的状态标志管脚，指示 FIFO 的当前状态；当 DSP 向 FIFO 写数据之前，DSP 应先读取 FLAGA-FLAGD 标志位，以判断其状态，若该端点不空，则等待，若空，则进行写数据操作。对 DSP 所需的 24 MHz 有源晶振和 CY7C68013 用到的 24 MHz 有源晶振要进行铺铜处理。

为使 68013 复位后由默认的“端口”模式转换到 Slave FIFOs 模式，IFCONFIG 寄存器的 IFCFG[1:0]必须设置为 11，此时 Slave FIFO 接口引脚被认为是外部主控制器。

4.5 布线结果

由于原理图文件比较大，所以利用 Protel 99se 软件将它分别画在几个 SCH 文件里通过画层次原理图将其联系在一起，完成了原理图的设计。接下来在 Protel 自带的功能将原理图导入，利用 PCB 制图原则完成 PCB 布线。具体制图原则如下所示：

1．各层导线应该相互垂直、斜交、或弯曲走线，避免相互平行，以减小寄生耦合;作为电路的输人及输出用的印制导线应尽量避免相邻平行，以免发生回授。

2．走线拐角尽可能大于 90 度，杜绝 90 度以下的拐角，也尽量少用 90 度拐角。

3．同是地址线或者数据线，走线长度差异不要太大，否则短线部分要人为走弯线作补偿。

4．PCB 设计对差分对的布线方式应该要适当的靠近且平行。所谓适当的靠近是因为这间距会影响到差分阻抗的值，此值是设计差分对的重要参数。需要平行也是因为要保持差分阻抗的一致性。若两线忽远忽近，差分阻抗就会不一致，就会影响信号完整性及时间延迟。另外有两点要注意，一是两条线的长度要尽量一样长，另一是两线的间距(此间距由差分阻抗决定)要一直保持不变，也就是要保持平行。平行的方式有两种，一为两条线走在同一走线层，一为两条线走在上下相邻两层。

根据以上 PCB 制图原则，得到图 4.18 布线结果。在实际布线的过程中制图的经验是很重要的。

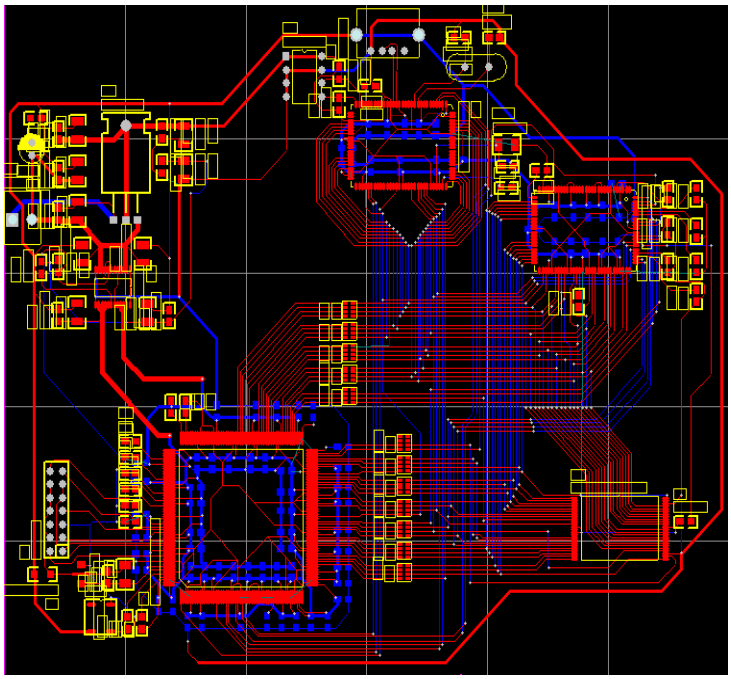

图 4.18 布线结果

4.6 仿真结果

4.6.1 仿真器

本系统使用合众达公司研发的 XDS560 仿真器与主机相连。仿真程序会利用开发系统将代码下载到 C6713 芯片中，我们编写的程序是在芯片上运行，仿真软件只是把运行结果读出来并显示。但是相对于软件仿真，硬件仿真的优点是仿真速度快，仿真结果与系统实现一致。所使用的 XDS560 仿真器的性能如下：

1．高速的实时数据传输速率，达 2MB/s；

2．目标 DSP 运行时，可以对变量进行实时的观测；

3．可以实时数据交换，和多种数据格式兼容，比如 Excel，Matlab，LabView 等；

4．只占用很少的 DSP 资源；

5．设置硬件断点和观察点；

6．事件和时序的管理；

7．精确地测量调试的时序；

8．实现的条件是：具有高速 JTAG 的 DSP，XDS560，CCS V3.3；

9．快速的代码下载速度，高达 0.5MB/s。

4.6.2 测试结果

平台配置为：

DSP：TI 公司的 TMS320C6713；

Flash：SST 公司的 SST39VF1601，1M×16；

SBSRAM：Cypress 公司的 CY7C1339，128K×32；

USB：Cypress 公司的 CY7C68013

测试主要涉及到存取器、Boot 过程的测试。

1．存储器测试

(1)Flash 测试

Flash 的测试主要包括 FLSAH 的擦出/写入操作，以及存储空间中的映射是否正确。测试的方法为从 Flash 起始地址 0x9000 0000 写入 4K 个从 0 开始的连续数值，将写入的数据与读入数据缓冲区的数据比较，如果正确则认为操作正确，如图 4.19 所示。

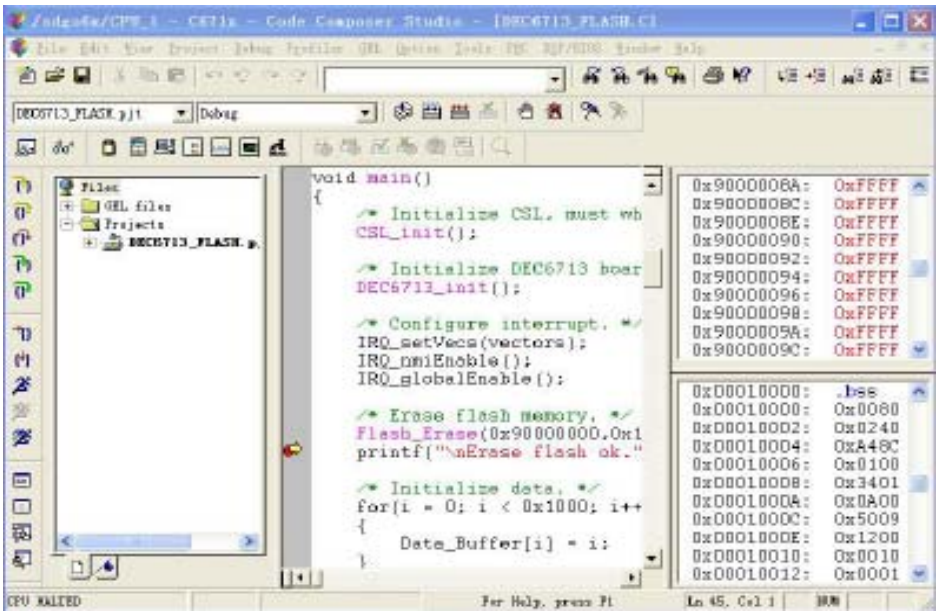

图 4.19 Flash 测试

得到测试结果为存储空间 0x9000 0000 起始内容均为 0xFFFFF；数据缓冲区中的数据内容也为 0xFFFFF。说明 Flash 测试通过。

(2)SBSRAM 测试

SBSRAM 的测试主要包括其控制逻辑读/写入操作。测试方法：在 SBSRAM 起始地址 0xA000 0000 写入 1K 个从 0 开始的连续数据，将写入的数据与读入数据缓冲区的数据比较，如果正确则认为操作正确，如图 4.20 所示。

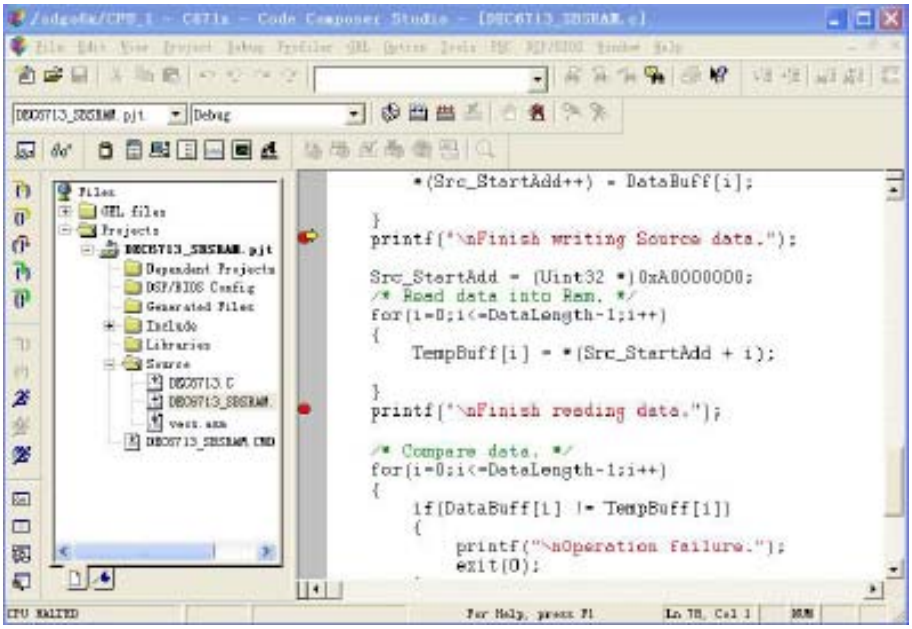

图 4.20 SBSRAM 测试

得到测试结果为存储空间 0xA000 0000 起始为 1K 个从 0 开始的连续数；数据缓冲区中的数据内容也为 1K 个从 0 开始的连续数。说明 SBSRAM 测试通过。

2．Boot 测试

在本设计中采用的是 Flash 引导过程，在测试之前首先上电复位，CPU 退出复位后执行二次 boot 程序，将程序代码复制到相应的运行地址空间中，完成复制后，自动跳转到 c_int00 处。

测试方法：启动 CCS 运行 Tools/FlashBurn，加入要转换的.cmd 文件，在 File to Burn 中加入要烧写的.hex 文件，设置逻辑地址为 0x0，设置 Flash 的物理地址和长度，Processor Type 为 C67x，运行 Program Flash，随后退出 CCS，重新打开电源就能观察到平台上的 LED 灯在闪烁，说明 Boot 成功，如图 4.21 所示

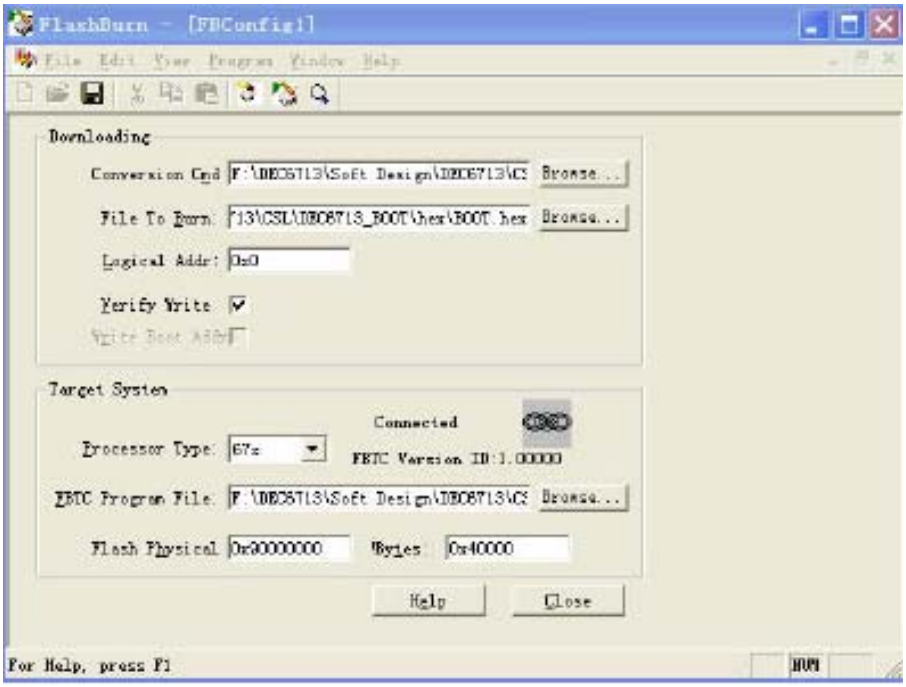

图 4.21 Boot 测试

4.7 本章小结

本章主要介绍了本设计中的各种相关硬件。首先介绍了以 DSP 为核心的总框图，然后分 USB 和 DSP 两部分给出了详细的介绍。USB 部分主要讲述了 CY7C68013 的工作原理和特征，给出了 USB 控制器外围电路设计和说明；DSP 部分主要阐述了 TMS320C6713 的结构和外设以及配置、引导、电路设计；随后对 USB 与 DSP 之间的通信部分给出了具体设计；在此基础上给出了具体布线结果；并在仿真器上进行了验证。说明该硬件系统结构简单，小巧，便携，可以作为多数保密通信场合的硬件支持平台。

第五章 混合加密体制在保密通信中的应用

5.1 保密通信背景介绍

本文所研究的课题是国内某研究所为了保密通信开发的专用系统。通信作为Internet 上最重要的服务同时，也是安全漏洞最多的服务之一。通信服务之所以是一种最脆弱的服务，是因为它可以被 Internet 上的非法分子利用技术手段窃取信息。由于缺乏有效的安全措施通信会给人们的隐私和信息安全带来威胁。

5.2 通信面临的安全问题

5.2.1 窃听

窃听一直是网络安全面临的一个严重问题，攻击者通过窃听，能够获得大量的传输信息，信息窃听的情况如图 5.1 所示。

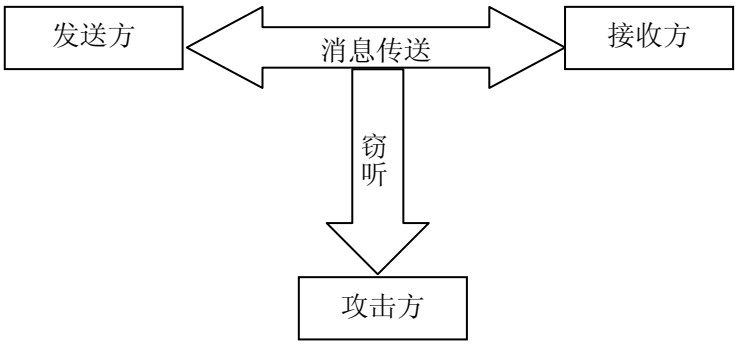

图 5.1 通信窃听

因为通信在网络上进行目标。所以它们更容易成为窃听的，根据网络环境的不同，在某些情况下进行进行窃听是很容易的。目前的网卡大都提供一种混杂模式，在这种模式下可以查看广播网段上所有的数据包，而不仅仅是查看进出该机器的包。这个窃听有时候称为 Snooping 或 Sniffing，并且以明文传送的任何数据都是它的目标。网络交换机可以减少这个问题，因为交换机仅把包发送到这些包要到达的端口，但对于网络管理员而言，就根本没有这个限制。窃听的问题不仅仅限制于局域网，在客户机和服务器之间的所有节点上都可能存在。防止窃听的一个比较有效的方法是使用密码，在网络上发送消息前加密机器之间的通信链路或者加密传送的数据。有时候仅保护数据是不够的，某些应用还要求保证发送方或接收方的身份可鉴别，或者进行互相保护。这样一来，即使通信在传输过程中被

窃听，窃听者也没有办法看到传输的内容，从而保护了用户之间正常的通信。

5.2.2 假冒身份

一般来说，对于计算机用户来讲，验证都不是一个陌生的名词。它提供一定程度的保证，证明用户就是他们所说的他们自己而不是别人。如果一个用户的身份被不怀好意者假冒，那么这个假冒者可以做真正的用户可以做的任何事情。身份假冒的情况如图 5.2 所示。

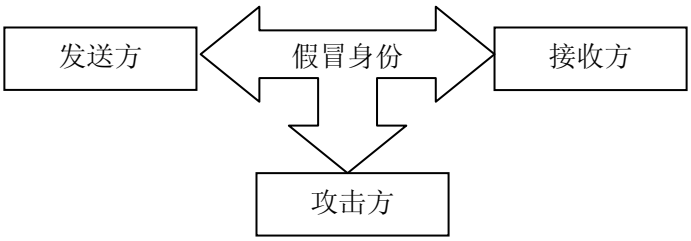

图 5.2 假冒身份

最简单的验证机制涉及使用一个标识身份的字符串(如帐号名)和密码，服务器为每个用户保存这些一一对应的值。为了进行验证，用户给服务器提供他的身份标识字符串和密码，服务器检查这两个值，如果它们是正确的，则认为该用户验证通过。如果数据不需要在网络上传送，这个机制就是一个可行的解决方案。如果某个人在网络上窃听，则他可以看到传送的身份标识字符串和密码，然后就可以使用这个信息假冒这个用户。实现验证的大多数旧有协议都以明文的方式在网络上传送密码信息，许多年来这一直是个严重的问题，所以互联网工程任务组(IETF,Internet Enginerring Task Force)现在要求所有新的协议都要支持更安全的方法传送用户保密信息。

假设密码不再以明文传送，这并不意味着问题就解决了，另外还有两个问题需要解决。第一个问题是黑客仍然可能使用字典式攻击(Dictionary Attack)重复尝试不同的密码组合，试图进入系统。这种形式的攻击在网络上是相当常见的。保护在网络上传送的密码并不一定能防止字典式攻击，因为黑客依然可能正确地猜出密码。使用字典式攻击所不能发现的密码对于这一类的攻击是有效的，但大多数的用户并不会这么做。即使用户这样做了，仍然有对密码的蛮力组合攻击的问题。有几个用于解决这个问题的技术，例如使用足够长度的真正随机的密码，这样就使字典式攻击和蛮力组合攻击在实际计算上是不可行的；或者使用一次性的密码，也能够提供保护。

密码不再以明文方式发送的第二个问题是中间人攻击(Man In Middle)这涉及一个人或程序插入在通信的双方之间。通信通道被这样暗中破坏，第三方可以截取在两个人之间发送的任何数据，修改数据，并发送给接收方。许多验证机制都或多或少地易于受这种类型的攻击。防止身份假冒，就是要具有不可否认性

(Non-Repudiation)这是防止发送方或接收方抵赖发送或接收信件的行为。一些验证机制非常强，能够提供不可否认性功能，这对合法绑定访问是非常重要的。

5.3 AES和ECC混合加密体制在保密通信中的应用

为了解决保密通信系统的机密性、身份鉴别、完整性和不可否认性，引进了AES 和 ECC 会合密码体制从而确保通信的安全服务。其实现是通过消息加密、消息签名、消息解密和消息验证来完成。

5.3.1 通信安全服务

为了通信的安全，必须保证通信的机密性、身份鉴别、完整性和不可否认性等。

1．机密性

机密性 (Privacy)可以保护被传输的信息免受被动攻击。对于消息内容的析出，能够确定几个层次的保护。最广义的服务可以保护在一段时间内两个用户之间传输的所有用户数据，例如，如果在两个系统之间建立一个虚拟电路，广义保护将防止经由该虚拟电路传输的任何用户数据被泄漏。也能够定义这种服务较狭义的形式，包括保护单一消息中的某个特定字段。但这些改进比起广义方法来用处较小，实现起来可能更为复杂而且费用很高。机密性的另外一个方面就是保护通信量免受分析，这使得一个攻击不能在通信设施上观察到通信量的源和目的、频度、长度以及其他一些特征。在安全通信中，机密性服务主要体现为只有指定的消息接收者才能读取信息内容。这种安全服务主要通过使用传统加密技术来实现。本文采用 AES 加密来实现。

2．身份鉴别

身份鉴别(Authentication)服务确保某一个通信是可信的。在诸如产生一个警告或警报信号的单个消息的情况下，身份鉴别服务的功能是能向接收方保证该消息确实来自于它所宣称的源。在诸如一个终端与一台主机连接的这样一个正在进行的交互情况下，身份鉴别服务涉及两个方面。首先，在连接发起时，该服务确保这两个实体是可信的(即每个实体都的确是它们所宣称的那个实体)。其次，该服务必须确保该连接不被干扰，使得第三方不能假冒这两个合法方中的任何一个来达到未授权传输或接收的目的。

在安全通信中，身份鉴别服务主要体现为消息接收者可以正确鉴别发送者的真实身份，防止有人伪装发送者发信。这种安全服务主要通过非对称密钥算法来实现。

3．完整性

如同机密性一样，完整性(Integrity)能够应用于一个信息流、单个信息或某个信息中的字段。同样的，最为有用和直接的方法是对整个流的保护。一个面向连接的完整性服务是处理消息流的服务，它能够确保接收到的消息如同发送的消息一样，没有冗余、插入、篡改、重排或者延迟，而且该服务也包括数据的销毁。因此，面向连接的完整性服务用于处理消息流的篡改和拒绝服务。在另一方面，无连接的完整性服务用于处理任何没有较长内容的单个消息，通常它只保护消息内容免受篡改。

在安全通信中，完整性服务主要体现为保证消息在发送的过程中没有被第三方篡改数据。这种安全服务主要通过数字摘要实现。

4．不可否认性

不可否认性(Non-repudiation)是防止发送方或接收方抵赖所传输的信息。因此，当发送一个信息时，接收方能够证实该消息是由所宣称的发送方发出的。类似地，当接收到一个消息时，发送方能够证实该消息的确是由所宣称的接收方接收的。

在安全通信中，不可否认性服务主要体现为消息发送者对自己所发送的消息不可抵赖，同时消息的接收者对自己接收的消息不可抵赖。这种安全服务主要通过数字签名技术来实现。

5.3.2 保密通信工作流程

为了实现通信的机密性、身份鉴别、完整性和不可否认性，对安全通信系统客户端而言，须采用消息签名、消息加密、消息解密和验证来完成。以下将分别介绍。

1．ECC 消息签名

ECC 对安全通信的内容进行签名的过程为：系统在发送方写好消息，点击发送后。系统将原始消息用算法产生数字摘要；再用发送方的私钥对前面产生的数字摘要做 ECC 加密，从而产生原始消息的签名；最后将原始消息的签名和原始消息打包形成签名消息，如图 5.3 所示。

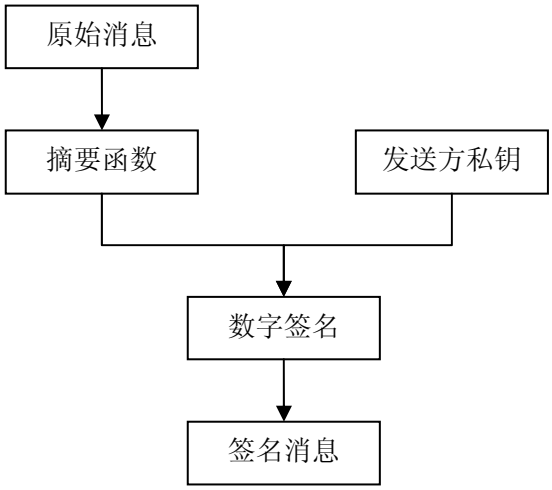

图 5.3 ECC 消息签名

2．AES 和 ECC 混合密码体制加密消息

在使用 AES 和 ECC 混合密码体制对消息加密时，其实是用 AES 加密原始消息。首先系统随机生成 AES 密钥；然后 AES 加密签名邮件；第三步系统使用 ECC 算法对 AES 密钥加密，从而形成 AES 密钥块；最后系统将加密消息和数字信封结合形成要发送的安全消息。其过程如图 5.4 所示。

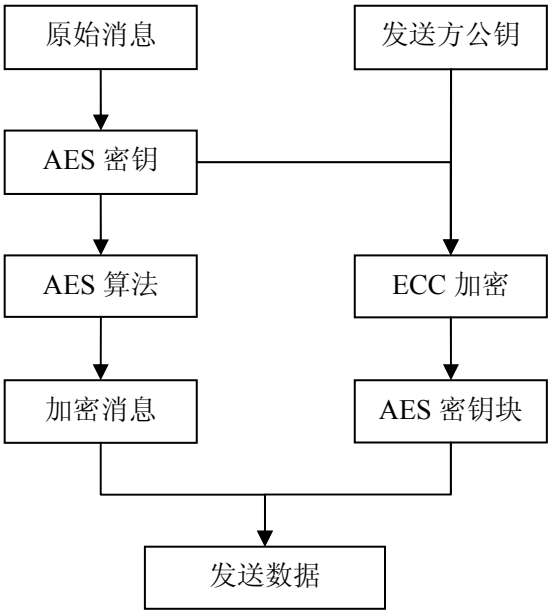

图 5.4 AES 和 ECC 混合密码体制加密消息

3．消息解密

对安全消息的解密过程是：在接收方收到消息后首先把消息分离成加密消息和 AES 密钥块；其次用 ECC 算法将 AES 密钥块解密，从而得到 AES 密钥；最后用 AES 密钥解密加密消息。其过程如图 5.5 所示。

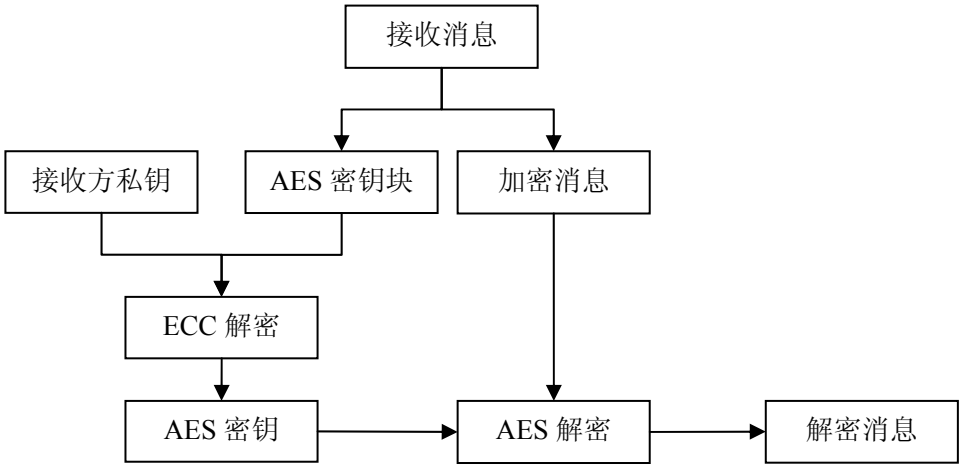

图 5.5 消息解密

5)消息验证

将消息进行解密后，为了保证消息的完整性和确认发信人的身份，需要进行验证。接收方在得到接收消息后，首先分离得到签名块，得到摘要结果；同时从解密消息中得到摘要函数；最后将上步得到的两个摘要函数比较，若相等则说明签名有效，接收方接收到的消息未被篡改和窃听，否则接收方接收到的消息不可信。其过程如图 5.6 所示。

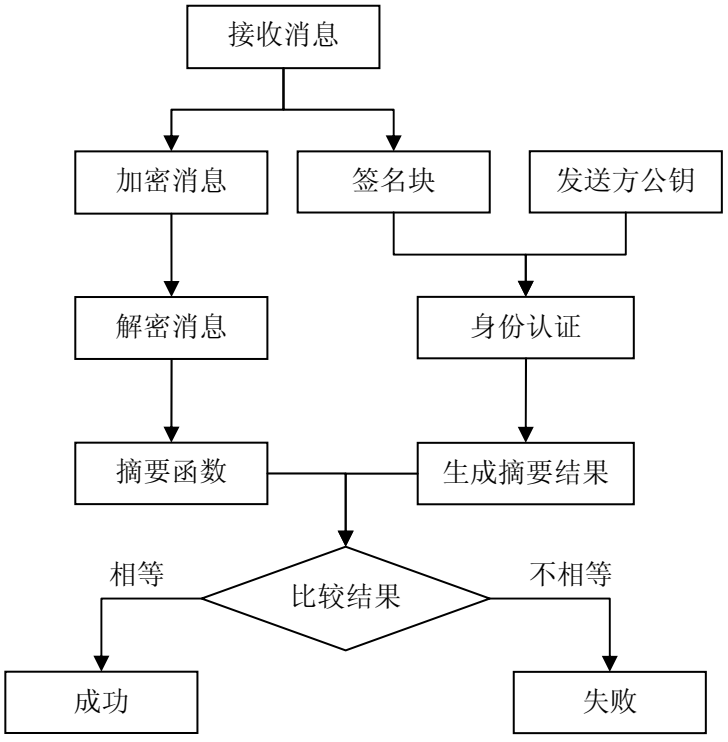

图 5.6 消息验证

5.3.3 保密通信安全标准

从信息安全角度讲，上述过程满足了通信 5 个安全方面的需求：

1．消息的完整性和签名的快速

将原始消息换算成摘要，相当于消息的指纹特征，任何对消息的修改都可以被接收方检测出来，从而满足了完整性的要求。再用发送方公钥算法(ECC)私钥加密摘要形成签名，这样就克服了公钥算法直接加密公文速度慢的缺点。

2．加、解密的快速性

选用公钥密码体制算法中速度较快的 ECC 算法，再加之整个消息传输过程只用它对通信的信息摘要和 AES 会话密钥加、解密各一次，因此加密量小，有利于提高系统加、解密的速度。而其余的加、解密工作由加、解密速度快的 AES 来完成。因此系统总体加解密的速度非常快。

3．安全的密钥交换

利用消息接收方的公钥加密通信密钥，这样就解决了对称密钥传输困难的不足。这种技术的安全性相当高。结合 AES 和 ECC 的优点，使用两个层次的加密来获得公开密钥技术的灵活性和对称密钥技术的高效性。

4．保密性和接收方的身份鉴别

消息发送方的对称密钥是用消息接收方的公钥加密并传给消息接收方的，由于没有别人知道消息接收方的私钥，所以只有消息接收方才能够对这份加密文件解密，从而满足保密性要求的同时又保证了消息接收方的身份鉴别。

5．消息发送方的身份鉴别和抗否认性

消息接收方用消息发送方的公钥解密数字签名，同时就认证了该签名的消息是消息发送方传递过来的。由于没有别人拥有消息发送方的私钥，只有消息发送方才能够生成可以用自己的公钥解密的签名，所以消息发送方不能否认曾经对该消息进行过签名并发送给收件方。

5.4 本章小结

本章在介绍了通信的机密性、身份鉴别、完整性和不可否认性的重要性的基础上，针对这些安全服务的技术，给出了 AES 和 ECC 相结合的加密体制的应用，对 AES 和 ECC 混合加密体制应用于某研究所保密通信做出了详细的阐述。

第六章 总结与展望

6.1 工作总结

随着计算机和网络的高速发展和广泛应用，人们对信息的安全保护需求愈益迫切，信息安全问题成为了人们关注的焦点。作为信息安全的核心问题，密码学发展十分迅速。非对称加密算法能适应网络的开放性要求，密钥管理简单，并且可方便地实现数字签名和身份认证等功能，但是由于其算法复杂，加密数据的速度和效率较低，往往不能满足人们的需求。同时对称加密算法的实现效率高、速度快，所以将两种加密算法结合的混合加密算法成为了人们研究的热点。

(1)给出了平台的总体设计目标和思想，根据这个思想构建了整个系统的框架以及系统硬件的设计方案，掌握了系统开发的基本方法和过程。熟练使用 Protel 99se 电路设计工具。

(2)研究了 USB2.0 接口的技术，对 USB 的体系结构、设备架构以及 DSP 技术进行了深入的探讨，并对平台系统进行仿真测试，验证了其正确性。

(3)深入研究了对称加密体制以及公开钥加密体制。借鉴了现有成功的公钥密码算法和私钥密码算法。提出了采用 AES 加密明文；用 ECC 加密 AES 密钥，进行密钥管理，同时使用 ECC 进行数字签名。

(4)针对保密通信的模式进行了详细的讨论，通过对现有数据加密体制的分析，将 AES 和 ECC 体制相结合的混合数据加密体制应用到保密通信中。从而更为高效地实现了网络通信系统中的信息加密、数据签名和身份验证。

6.2 展望

由于时间紧迫，加之笔者水平有限。DSP 平台还没有进行实际实现；算法仅仅是模拟实验，具体的算法设计可能还存在一些不足。今后将继续在算法的改进方面做进一步的研究，期望有新的突破，同时争取完善系统。并将算法实现在实际的 DSP 平台中。另外，本文对密码算法的研究主要集中在算法本身，而较少从密码分析学的角度考虑，对于算法的安全性和效率主要依赖于相关资料的数据。希望以后在研究中通过实验来证明其安全性和效率。

在电子商务和电子政务蓬勃发展的当今社会，保密通信将会越来越来受到更多的关注。应该说本文所做的工作是很有意义的。

结论

随着互联网的广泛应用，信息安全问题日益突出，以数据加密技术为核心的信息安全技术也得到了极大的发展。鉴于 AES 对称加密算法简单和加密速度快的优点，目前仍然是主流的密码体制之一。由于算法在算法实现、效率、强度等方面多都表现出良好的性能，从目前来看，还没有有效的攻击方法，因此必将成为未来对称加密体制的必然选择。另一方面，由于 ECC 公钥密码体系的密钥管理非常方便，而且能实现数据签名和身份验证等功能，从而使其成为保密通信技术的核心基础。在相同的安全条件下，基于椭圆曲线的加密和数字签名算法在计算量、处理速度、存储空间等方面都比其它公钥密码算法有着很大的优势。

本文通过对现有数据加密体制的分析，并提出了一种将算法和椭圆曲线密码体制相结合的混合数据加密体制，从而更为高效地实现了保密通信系统中的信息加密、数据签名和身份验证，解决了密码体制中速度和安全性不能兼顾的问题。并详细描述了加密平台的设计过程，并对平台系统进行仿真测试，验证了其正确性。在保密通信中，本文所做的工作是很有意义的。

但由于时间紧迫，加之笔者水平有限，故对密码算法的研究主要集中在算法本身，而较少从密码分析学的角度考虑，对于算法的安全性主要依赖于相关资料的数据和证明。

致谢

本文的最后，我真诚的感谢我的导师牛海军教授。在这三年中，他严谨的治学态度、渊博的知识、高尚的品德令我折服。在学习和生活上都给了我很大的帮助，使我受益非浅。此外，牛老师在繁忙的教学和科研工作之余，自始至终以身作则教予我做人的道理，使我少走了许多弯路。在此，对牛老师表示衷心的感谢。

衷心感谢张泽增教授、赵琳教授、毛立强老师在校期间学习和生活上给予的无私关怀和悉心教导，他们精湛的学术造诣和高尚的品格使我终生受益。

感谢王贝、胡继雷、董晓巍、王苏娅、王永强，还有在这三年学习生活中提供帮助的同学、朋友，以及实验室各位师兄、师姐、师弟、师妹对我顺利完成毕业论文的大力支持。

最后，要特别感谢我的父亲、母亲以及可爱的妹妹。感谢他们二十多年无怨无悔的关爱和付出，帮助我克服了诸多困难，解决了后顾之忧，使得自己能够顺利完成学业。

参考文献

[1] Rushdan M, Kilicman A.A Fast Software Implementation for Arithmetic Operationsin GF (2^n) [J]. Computer Science .1996(1163).65-76.

[2] Daemen J, Knudsent L R, Rijmen V. “The block cipher square,”in Fast Software Encryption [J]. Computer Science. 1997(1267).149-165.

[3] Law L, Menezes A, Qu Metal. An efficient protocol for authenticated key agreement [J]. Designs, Codes and Cryptography. 2003.28(2).119-134.

[4] Michael Welschenbach. 密码编码学——加密方法的 C 与 C++实现[M]. 赵振江等译. 北京:电子工业出版社. 2003.

[5] William Stallings. 密码编码学与网络安全——原理与实践(第三版)[M]. 刘玉珍等译. 北京:电子工业出版社. 2004.

[6] Darrel Hankerson, Alfred Menezes, Scott Vanstone 等. 椭圆曲线密码学导论[M]. 张焕国等译. 北京:电子工业出版社. 2005.

[7] Bekyel E. The density of elliptic curve having a global minimal Weierstrass equation [J].Journal of Number Theory. 2006.

[8] Cristian, Glesner, Manfred. An FPGA implementation of the AES-Rijndael in OCB/ECB modes of operation [J]. Microelectronics Journal. 2005, 02. 139-146.

[9] 俞经善, 王晶, 杨川龙. 基于 ECC 和 AES 相结合的加密系统的实现[J]. 信息技术. 2006, 30 (2).44-46.

[10] Daniel V. Bailey, Christof Paar, Johannes Buchman. Efficient Arithmetic in Finite Field Extensions with Application in Elliptic Curve Cryptography [J]. Journal of cryptology. 2000, 09(14).153–176.

[11] Mangard S, Aigner M, Dom Nikus S.A Highly Regular and Scalable AES Hardware Architecture [J]. IEEE TRANSACTIONS ON COMPUTERS. 2003.52(4). 483-491.

[12] 王宏, 许丙南, 杨铁牛等. 基于 EZ-USB FX2 的多路同步数采固件设计[J]. 微计算机信息,2006. 22(12-1).126-128.

[13] Cypress Semiconductor Corporation. EZ-USB FX2^{TM} Technical Reference Manual[R]. 2005.

[14] Texas Instruments. TMS320C6000 CPU and Instruction Set Reference Guide (SPRU189) [R]. 2003, July.

[15] Texas Instruments. TMS320C6000 Peripherals Reference Guide (SPRU190)[R]. 2003, July.

[16] Texas Instruments. TMS320C6000 DSP 外设概览参考指南(ZHCU001H)[R]. 2005,04.

[17] Texas Instruments.TMS320C6000 指令集仿真器技术参考(ZHCU002)[R]. 2005,04.

[18] Texas Instruments. TMS320C620x/C670x DSP Boot Modes and Configuration Reference Guide (SPRU642) [R]. 2003, July.

[19] 刘勇，潘燕主编.PCB 设计基础[M].北京:机械工业出版社. 2005.

[20] Texas Instruments.TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide (SPRU266E) [R]. 2008, April.

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

[22] P Ivey, S Walker, J Stern, and S. Davidson. An ultra-high speed public key encryption processor [J]. In Proceedings of IEEE Custom Integrated Circuits Conference, Boston, 1992, 19(6).1-4.

[23] IEEE STD 1363-2000 IEEE Standard Specifications for Public-Key Cryptography. Microprocessor and Microcomputer Standards Committee of the IEEE Computer Society [S]. 2000, January.

[24] K Araki, S Miura, and T Satoh. Overview of elliptic curve cryptography [J]. In International Workshop on Practice and Theory in Public Key Cryptography. 1998. 1-14.

[25] 刘晓莹, 祝跃飞, 郭艳等.椭圆曲线密码体制在 TMS320C6201 上的实现[J]. 信息工程大学学报. 2004, (0l). 35-37.

[26] Henna Pietiläinen. Elliptic curve cryptography on smart cards [D]. Helsinki University of Technology. 2000, October.

[27] 赵涛,王春迎,顾纯祥等. 基于 DSPTMS320C6201 芯片的 ECC 实现[J]. 计算机工程 . 2006, 05. 32(10).161-163.

[28] 胡予濮等编著. 对称密码学[M]. 北京:机械工业出版社. 2002.

[29] Texas Instruments. TMS320C6000 Programmer's Guide (SPRU 198) [R]. 2003, July.

[30] Kouichi Itohl, Masahiko Takenaka, Naoya Torii .Fast Implementation of Public-Key Cryptography on a DSP TMS320C6201 [J]. Lecture Notes in Computer Science.1999. 12(1717).61-72.

[31] L ElGamal.A Public Key Cryptosystem and Signature Scheme Base on Discrete

Logarithm [J].IEEE Transactions of Information Theory.1985.31.469-472.

[32]李佟鸿, 麦永浩. 椭圆曲线密码体制安全性分析[J]. 网络安全技术与应用. 2007. (7).92-93.

[33]肖世文.USB2.0 硬件设计[M].北京:清华大学出版社.2002.

[34]李方慧,王飞,何佩琨.TMS320C6000 系列 DSPs 原理与应用[M].北京:电子工业出版社. 2002.390-433.

[35]Hakim Khali, Ahcene Farah, Cost-Effective Implementations of GF (p) Elliptic Curve Cryptography Cryptography Computations [J].IJCSNS International Journal of Computer Science and Network Security.2007.7 (8).29-37.

[36]Diffie W, Hellman M. New directions in cryptography [J]. IEEE Transactions on Information Theory. 1976. 22(6).644-654.

[37]R L Rivest, A Shamir, L Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems [J]. Communications of the ACM. 1978. 21(2).126-129.

[38]Menezes A, Okamoto T, Vanstone S. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field, IEEE Transactions on Information Theory. [J].IEEE Transactions on Information Theory. 1993. 39(5).1639-1646.

[39]白国强. 椭圆曲线密码及其算法研究[D]. 西安电子科技大学 .2000.

西安電子科技大學

地址：西安市太白南路2号
邮编：710071
网址：www.xidian.edu.cn