物联网网关轻量级认证和加密技术研究

【原文对照报告-大学生版】

报告编号：1c85957d6ecbe8d0　　检测时间：2020-05-11 10:31:37　　检测字数：21,485字

作者名称：郑智聪　　所属单位：维普论文检测（河北）

检测范围：

◎ 中文科技期刊论文全文数据库
◎ 中文主要报纸全文数据库
◎ 中国专利特色数据库
◎ 博士/硕士学位论文全文数据库
◎ 中国主要会议论文特色数据库
◎ 港澳台文献资源
◎ 外文特色文献数据全库
◎ 维普优先出版论文全文数据库
◎ 互联网数据资源/互联网文档资源
◎ 高校自建资源库
◎ 图书资源
◎ 古籍文献资源
◎ 个人自建资源库
◎ 年鉴资源
◎ IPUB原创作品

时间范围：1989-01-01至2020-05-11

检测结论：

全文总相似比		复写率		他引率		自引率		专业术语
44.05%	=	35.24%	+	8.81%	+	0.0%	+	0.0%

其他指标：

自写率：55.95%

专业术语：0.0%

高频词：加密,密钥,网关,服务,务器

典型相似性： 无

指标说明：

复写率：相似或疑似重复内容占全文的比重

他引率：引用他人的部分占全文的比重，请正确标注引用

自引率：引用自己已发表部分占全文的比重，请正确标注引用

自写率：原创内容占全文的比重

专业术语：公式定理、法律条文、行业用语等占全文的比重

典型相似性：相似或疑似重复内容占互联网资源库的比重，超过30%可以访问

总相似片段：401

期刊：58 博硕：136 外文：0 综合：3 自建库：27 互联网：177

河北北方学院

本科毕业论文

物联网网关轻量级认证和加密技术研究

题 目：物联网网关轻量级认证和加密技

技术研究

姓 名：郑智聪

学 号：201642030

院 系：信息科学与工程学院

专 业：电子信息科学与技术

年 级：2016级

指导教师：郭本振

二〇二〇年 六 月

附表： 4（指导教师指导学生填写）

河北北方学院本科毕业论文任务书

论文题目：物联网网关轻量级认证和加密技术研究

学生姓名	郑智聪	学号	201642030	专业	电子信息科学与技术
指导教师	郭本振	职称	讲师	论文（设计） 起止时间	2019.11.15-2020.5.20
选题来源	教师科研课题（ ）； 生产、工程或社会实践课题（ ） 学生自拟课题（ ）； 师生共同拟定课题（ √ ） 大学生创新创业训练项目（ ）； 学科竞赛（ ）				
	1.设计构想 设计一种物联网网关与云服务器之间双向认证和秘钥协商的算法方案，并能够将通信的数据加密传输。实现一种高效、				

主 要 任 务	可扩展的秘钥管理机制。 2.设计方案 (1)云服务器端：采用Java语言进行云服务器开发，使用Netty网络通信框架，高性能、可扩展，上手容易，足以满足系统所需要的功能，可进行快速开发。 (2)模拟物联网网关： 采用TCP协议与云服务器端通信，使用Java语言模拟嵌入式设备的逻辑，通过设置JVM启动参数的方式模拟嵌入式设备较低的资源。 3.技术路线 本系统主要由云服务器端、物联网网关客户端构成。通信的双方使用基于椭圆曲线ECC的秘钥交换（ECDH）和数字签名（ECDSA）进行秘钥协商和身份认证，通过后使用AES128进行数据的加密传输。通过VUE编写Web客户端，可视化的进行网关和秘钥管理。 4.任务进度 2019.11.15-2019.12.15 进一步查阅资料，了解选题方向上国内外研究现状。 2019.12.16-2020.02.29 整理相关资料，系统学习和分析用到的技术点。 2020.03.01-2020.04.15 确定整体软件框架，实现基本的认证和加密功能。 2020.04.16-2020.05.30 对完成的程序进行测试、优化、分析并得出结论； 2020.05.30-2020.06.01 进行毕业答辩。
教 研 室 意 见	本毕业设计题目结合专业实际情况，紧扣专业知识；研究内容涉及专业多门课程，需要综合应用所学专业知识；研究内容难度适中，任务明确，工作量合理；进度安排明确，同意作为毕业设计任务。 教研室主任签字： 年 月 日
毕业论文工作领导小组意见	论文选题能够结合科研、生产实际；难度把握得当，任务量适中；指导教师对课题研究领域熟悉，能够胜任指导工作；进度合理，可以作为毕业设计任务。 组长签字： 年 月 日

物联网网关轻量级认证和加密技术研究

Lightweight authentication and encryption technology for IoT gateway

Technical research

郑 重 声 明

本人呈交的学位论文（设计），是在指导教师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。除文中已经注明引用的内容外，本学位论文（设计）的研究成果不包含他人享有著作权的内容。对本论文（设计）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文（设计）的知识产权归属于河北北方学院。

本人签名： 日

摘 要

随着物联网技术的不断发展，人们越来越重视物联网的安全问题。本课题提出了一种在基于物联网技术设计的温室环境监控系统中，实现云服务器与物联网节点设备（如传感器、控制器等）之间的双向身份认证和数据加密的安全方案。该方案使用基于椭圆曲线ECC

的数字签名(ECDSA)和秘钥协商(ECDH)算法，使用AES-128算法加密传输数据，保证了物联网系统的安全。通过实际调试和实验证明该方案可在较低的资源需求情况下，满足云服务器与物联网网关设备之间，进行双向身份认证的设计需求，对本方案所实现的功能进行了全面的功能测试，整套系统运行稳定，操作简易，界面人性化。

关键词 物联网安全，双向认证，密钥协商，椭圆曲线，安全策略

ABSTRACT

With the continuous development of the Internet of Things technology, people pay more and more attention to the security issues of the Internet of Things. This topic proposes a security solution designed to implement two-way identity authentication and data encryption between cloud servers and IoT node devices (such as sensors, controllers, etc.) in a greenhouse environment monitoring system designed based on the Internet of Things technology. The scheme uses the digital signature (ECDSA) and secret key negotiation (ECDH) algorithms based on elliptic curve ECC, and uses the AES-128 algorithm to encrypt transmission data, ensuring the security of the Internet of Things system. The actual debugging and experiment prove that the solution can meet the design requirements of two-way identity authentication between the cloud server and the Internet of Things gateway device under the condition of lower resource requirements. A comprehensive functional test of the functions realized by the solution is carried out. The whole system is stable in operation, easy to operate and user-friendly interface.

Key Words IoT security; Two-way authentication; Key agreement; Elliptic curve; security strategy

目 录

第1章 引言

1.1 研究背景

随着互联网和通信技术的快速发展，人们已经不再满足传统的人与人以及其他需要人参与交互的通信方式，物联网——这种不需要人参与，只需要机器和机器之间信息交互的通信方式应运而生。1999年，MITAuto-ID中心的Ashton 教授最早提出了物联网这个名字，2005 年在国际电信联盟（ITU）发布的《ITU 互联网报告2005：物联网》报告中，再次提出用了“物联网”的概念[1]。在十二五规划中，物联网被列为七大战略新兴产业之一，是我国重点发展的领域，在未来物联网将会彻底改变人们的生活。

在物联网蓬勃发展的同时，出现了很多针对物联网的攻击事件，人们越来越关注物联网系统的安全问题。2015年黑客针对乌克兰的电力系统发起恶意攻击，导致70多万居民家庭停电数小时；2016年的Mirai事件中，攻击者利用网络摄像头等大量的物联网设备向域名服务器发起DDoS攻击，导致大量用户无法使用网络。据Gartner调查，全球近20%的单位和部门，近年来遭受过物联网攻击。

一般的物联网体系主要由三层组成，从上往下分别为：应用层、传输层、

感知层。物联网一般的工作模式是：感知层负责感知周围的信息，并通过传输层连接上应用中心，应用中心负责数据的汇集、分析

等。其中传输层和应用层可以在现有的、成熟的架构基础上运作实施，这两层的安全保护都有成熟的认证体系，

而对于感知层，因为硬件资源受限、传感器节点分布较广泛等特点，现有的安全技术无法很好地实施，迫切需要一种轻量级的认证和加密体系。

1.2 国内外研究现状

1.2.1 国外研究现状

近年来，随着物联网中的安全问题日益暴露，世界各地越来越多的官方组织、学术机构加入到物联网安全的研究当中，旨在构建一系列的安全规范和协议。

Nicanfar 等[2]在 2015 年提出了一个基于椭圆曲线加密的认证协议。尽管他们的方案大大降低了计算复杂度，但是由于可信任的第三方需要密码表来保存用户信息而易受到表丢失或被窃取的攻击。之后，Li等[3]提出一个新的密码协议但同样也被证明易受到窃听，并且由于缺少密钥协商而易受到模仿攻击。2019年Q.Jiangetal.[4]在分析Das协议的基础上，提出了一种新的在云服务器协助下，实现可穿戴设备认证和密钥协商的安全协议。该协议使用了ECC算法进一步增强了数据的安全性，降低了协议对设备计算资源消耗。2019年Wangetal[5] 提出了一种使用ECC算法和云服务器辅助的物联网网关与用户智能手机进行双向身份认证和密钥协商的协议。

1.2.2 国内研究现状

近年来，物联网在我国的发展已经进入多行业落地阶段，同时也开始进入物联网安全建设阶段，相关科研单位及安全厂商都在积极探索物联网安全规范与技术[6] 。

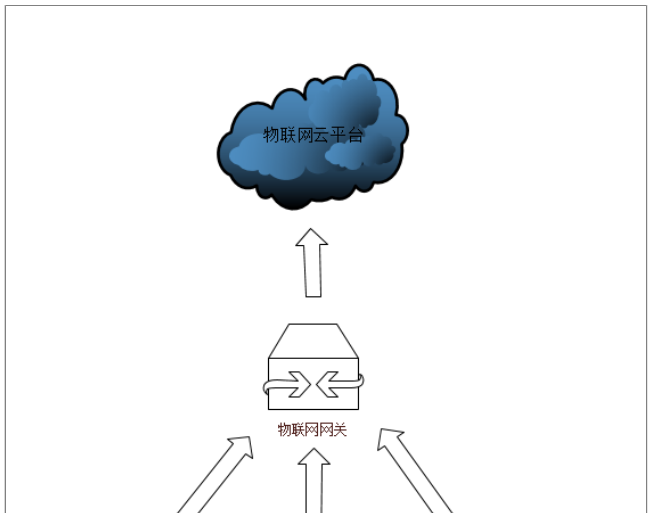
2017年，汪洋在《物联网轻量级认证和加密技术研究》中，提出了一种新的 RFID 双向认证协议，在认证信息中加入随机数和时间戳，利用椭圆曲线密码算法（Elliptic curve cryptography，ECC）对认证过程中的敏感信息进行加密，保证认证信息的机密性[1]。2018年，史冰清在《高安全性的物联网网关设计与实现》中设计实现了能够在物联网网关上使用的混沌-AES加密算法，该算法密匙混沌化、密钥空间更大、实现了“一块一密”，并且没有增加密钥管理的负担[7]. 同年，王斌在《工业物联网信息安全防护技术研究》中，针对物联网不同的层次结构，针对性地设计了应用于不同层次结构的安全防护策略[8]

1.3 主要研究内容

本文针对当前物联网系统感知层设备多数使用对称密钥协议来构建网络安全基础设施的实际场景，考虑到密钥维护工作繁多，运维人员经常接触通信密钥，容易引起密钥泄漏和内部特权人员攻击的情况。以及感知层设备计算、存储和通信资源有限，攻击者容易发起资源耗尽型的DDos攻击的情况。使用EC加密算法设计了一种系统云认证服务器（Cloud Authentication Server CAS）与物联网网关设备之间的双向认证、密钥协商的解决方案。

1.4 研究意义

对万事万物的感知能力是物联网技术的精髓，因此，感知层构成了物联网体系中举足轻重的一部分。感知层通过传感器等设备，借助于蓝牙、无线网络等传输信息到物联网中心系统，其结构图如下：



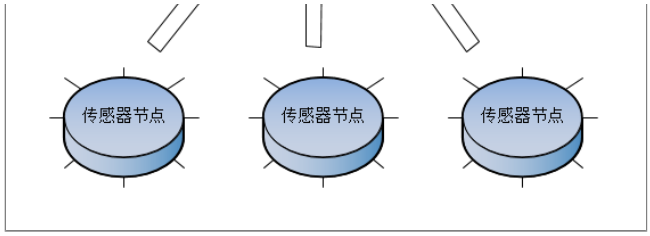


图 1.1 感知层结构图

在一般的物联网系统中，通常使用AES,DES等对称加密算法，但是对称加密算法又存在着密钥泄露的风险，一旦攻击者通过非法手段获取了密钥，后果不堪设想。而嵌入式设备又有着受限的资源，严苛的工作环境等特点，使用RSA等非对称加密算法的话，大大影响传感器的工作效率。因此，迫切的需要一种适用于嵌入式设备的轻量级认证和加密协议，在保障安全性的基础上，尽量降低带宽、内存等资源的占用，提高密钥协商和加密解密的效率，这对于物联网安全的发展有着深远的意义。

第2章 相关理论和技术

2.1 对称和非对称加密

随着网络信息技术的发展，网络信息的安全问题成为了阻碍其发展的最大因素。人们的信息在网络中传输，很容易被不法分子拦截并篡改，轻则造成人们隐私数据的泄露，重则造成人民个人财产的损失。

而对网络信息进行加密则是保证机密信息和数据泄露的主要手段，以下为密码学中的一些基本概念。

（1）加密：将数据按照一定的规则进行变换的过程

（2）解密：将加密后的数据，按照规则转换成原数据的过程

（3）明文：加密之前的数据，能够被轻易读取

（4）密文：加密之后的数据，隐藏原文本的信息

（5）秘钥：控制加密和解密过程的参数

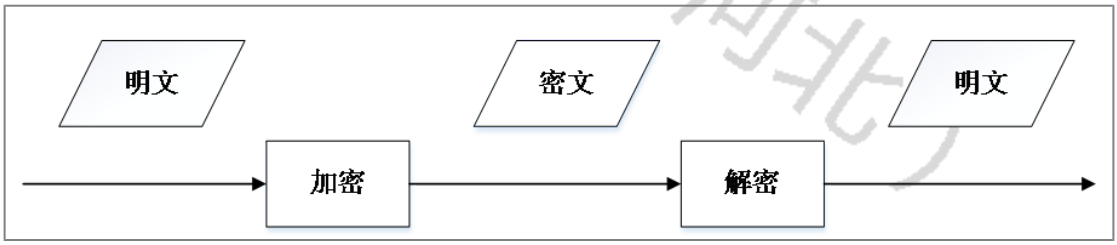
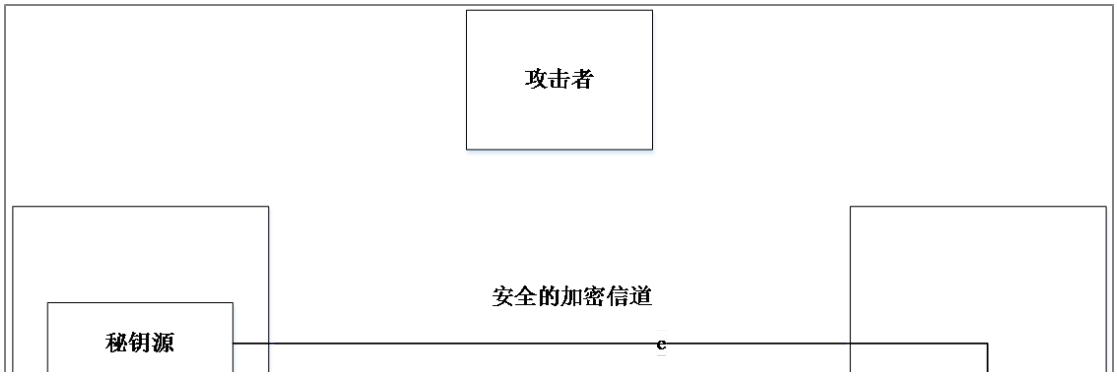


图 2.1 数据加密解密过程

2.1.1 对称加密

对称加密指的是，进行加密和解密时使用的秘钥是相同的。其特点是加密解密的速度较快，实施起来较为简单的同时也能有较好的安全性。对称加密的过程如下图所示：



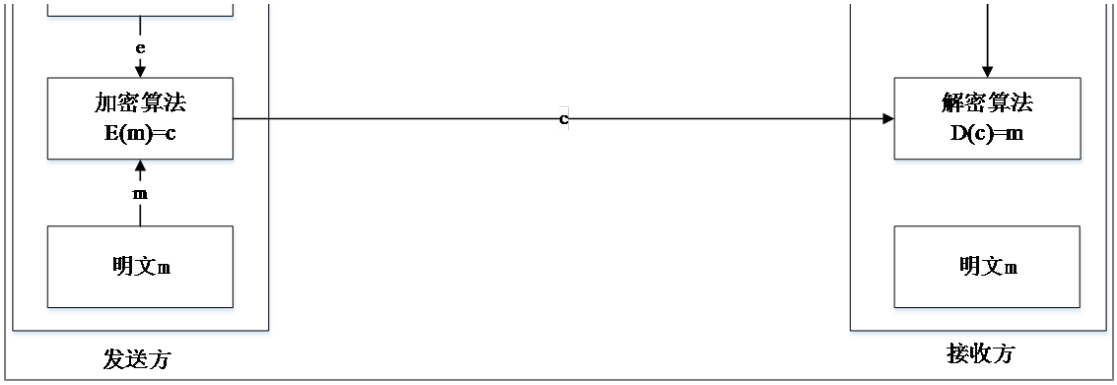


图 2.2 对称加密模型

常用的对称加密算法按照加密解密的对象进行划分，还可分为分组密码算法和流密码算法两种。分组密码的加密方式是先将待加密的数据进行编码，然后将编码的数字划分成等长的分组，再对每一个分组进行密钥加密。流密码的加密方式是对流数据的每一位或每字节进行加密处理。

由于对称加密的双方使用相同的密钥，保障算法的安全性的前提是保障密钥的安全性，双方都不能将密钥泄露出去，否则会面临密码被破解的危险。而在物联网的环境中，网关节点等距离服务器都比较远，不可避免地具有在网络中传输密钥的情况，在密钥传输的过程中很容易被窃取。

2.1.2 非对称加密

非对称加密算法最早产生于上世纪七十年代[9]，与对称加密不同的是，非对称加密需要两个密钥来进行加密和解密，分别为公开密钥（public key，简称公钥）和私有密钥（private key，简称私钥），公钥加密的信息需要私钥才能解密，私钥加密的信息需要公钥才能解密。公钥可以在网络上进行传输，任何人都可以获得公钥，不存在密钥泄露的问题。常用的非对称加密算法有RSA,ECC两种。1. RSA算法

RSA密码体制是由 Rivest、Shamir 和 Adleman 在 1977 年联合提出的[1]。RSA 算法的原理是基于一个数论事实：将两个大素数相乘很容易，但是想要对其乘积进行因式分解却极其困难，因此可以将乘积公开作为公钥。

2.ECC算法

ECC算法是在1985年由Neal Koblitz和Victor Miller分别独立提出的。

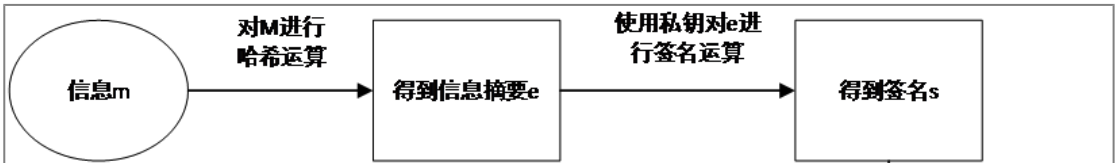
相比于RSA算法，ECC可以做到在同等的安全强度下，具有更小的密钥长度。ECC算法的原理是定义椭圆曲线上的运算，然后利用基于有限域上椭圆曲线点群离散对数分解难题，将倍点运算的结果作为公钥。本文将在2.3小节详细介绍ECC算法。

2.2 数字签名技术

在信息安全领域，数字签名是公钥加密体制的重要应用，就像在实际世界上用手写的签名和印章一样，通过在原始数据上附加一些数据,或是对原始数据做一个密码变换。这种附加数据或变换允许原始数据的接收方用以确认原始数据的来源和原始数据的完整性并保护数据,防止被中间人进行伪造。

数字签名技术的原理是将原文通过特定HASH函数计算出摘要信息，把摘要信息用发送方的私钥加密，与原文一起发送给接收方。接收方只有用发送方的公钥才能解密被加密的摘要信息，发送方用相同的HASH函数对收到的原文计算摘要，与解密得到的摘要进行对比。如果有一个字符不相同，用HASH函数生成的摘要就一定不同。如果比对结果一致，则说明收到的信息是完整的，在传输过程中没有被修改，否则信息一定被修改过，因此数字签名能够验证信息的完整性。

数字签名的总体流程如下：



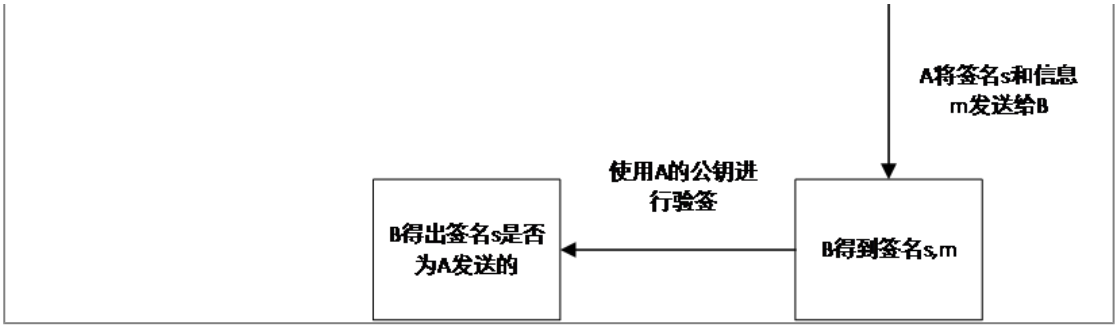


图 2.3 数字签名流程

常见的数字签名算法有RSA、DSA等，本文研究的基于椭圆曲线的数字签名算法是在DSA的基础上，利用椭圆曲线的离散对数难题，提高了安全性和计算效率。

2.3 椭圆曲线相关理论

2.3.1 椭圆曲线概述

本文研究的椭圆曲线是指满足式2-1的方程

$y^2 \equiv x^3 + ax + b \ mod \ p \ where (x,y) \in Z_p$	式（2.1）

式2-1中p是一个较大的素数，且p值越大安全性越高，计算量也就越大。a，b∈ Ep用于确定具体的椭圆曲线，且需要满足公式2-2:

$4a^3 + 27b^2 \neq 0$	式（2.2）

满足上述要求的椭圆曲线Ep(a,b)中，取点G，和整数k 则 K = kG，K也是椭圆曲线Ep(a,b)的点。在椭圆曲线中，给定G，k，容易求出 K；但如果已知K和G，很难求出k。在椭圆曲线加密算法中，G称为基点（ base point），k称为私钥（ private key），K称为公钥（public key）。确定了椭圆曲线和公钥、私钥之后，就可以用这些参数对数据进行加密

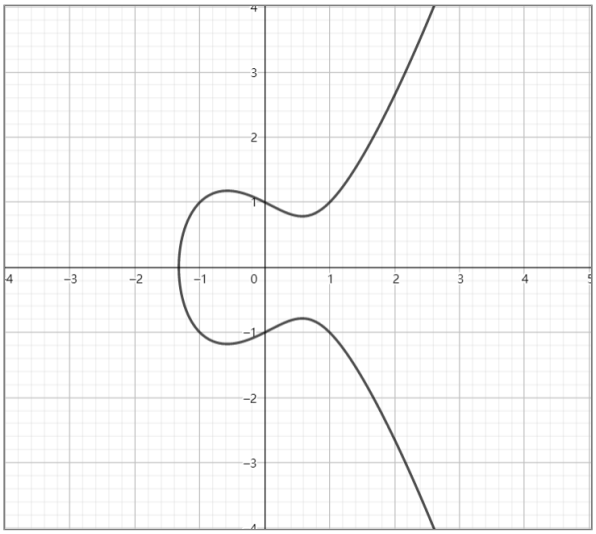


图 2.4 椭圆曲线y^2 =x^3-x+1

2.3.2 椭圆曲线上的运算

椭圆曲线上的运算，其实指的就是椭圆曲线上点的加法和乘法

1.椭圆曲线上的加法

设椭圆曲线上有A和B点两点，做过这两点的直线与椭圆曲线相交于C点，然后关于X轴对称得到D点，则D为A、B两个点的和，记作D=A+B。很明显，D点也在该曲线上。所以椭圆曲线上两点之和也是曲线上的点。

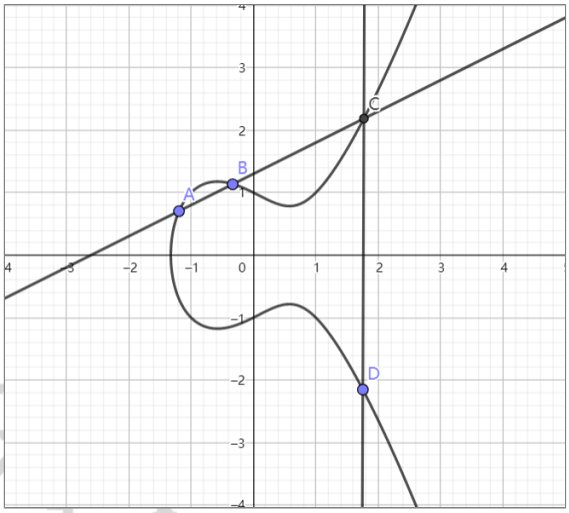


图 2.5 椭圆曲线加法

若A=B,则为过A点的切线交于椭圆曲线为R'。如下图所示。

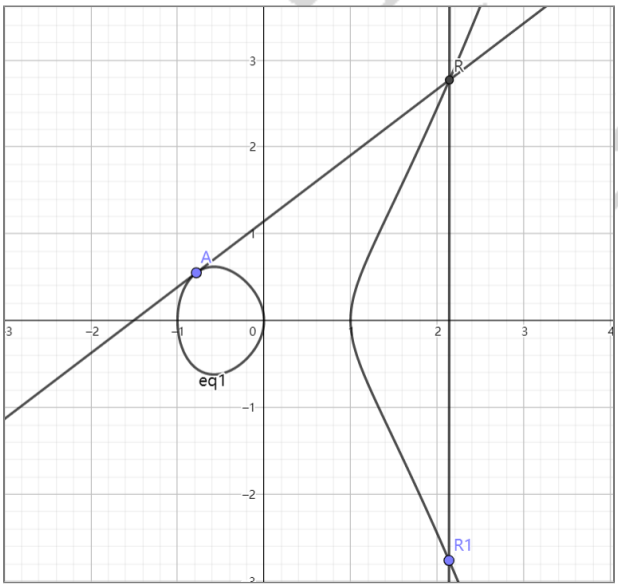
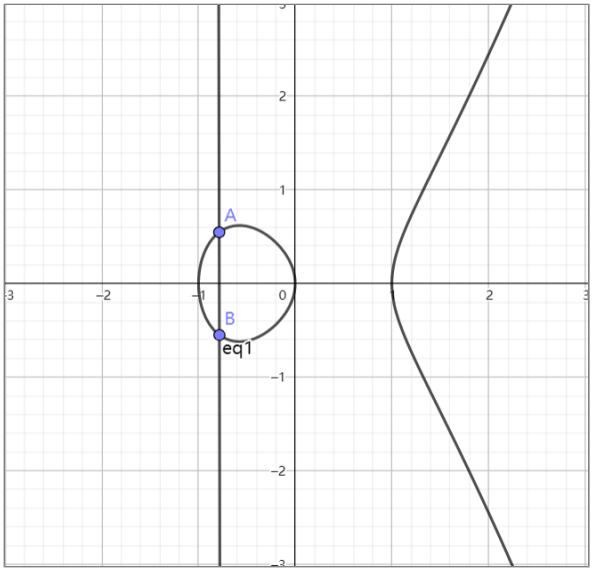


图 2.6 A=B时椭圆曲线加法

如果点A、B所在的直线刚好平行于Y轴时，根据椭圆曲线的性质，将永远不会有与曲线的第三个交点，我们定义坐标系中距离X轴无穷远点为椭圆曲线上的一个特殊点，称为O点（零点）。

图 2.7 AB垂直于X轴时椭圆曲线加法



因为椭圆曲线关于X轴对称，所以对于曲线上任意一点A，总存在另一点B使得过A、B的直线垂直于X轴，也就是该直线与曲线交于O点，所以A+B=O。因为O点是距离X轴无穷远的点，所以过A点与O点的直线是垂直于X轴的，它与曲线相交于另一点B点，那么B点关于X轴对称的点就是A点，即A点为A点和O点之和。

由椭圆曲线加法的定义可以得出，椭圆曲线的加法满足交换律（A+B=B+A）和结合律（(A+B)+C = A+(B+C)）。

2.椭圆曲线上的乘法

给定椭圆曲线上的一个点P，计算kP的过程也可以看成连续k个P相加的过程，此过程又称为倍点运算。由于椭圆曲线上的加法满足结合律，那么

P+P+P+P = 2P+2P

这样一来，假设要计算16P，则可以先计算出来2P，然后计算4P，如此类推，可以把16次的加法运算减少到4次，从时间复杂度的角度来看，这个算法是一个O(logk)的算法，这个方法被称为快速幂算法。可以大大减小倍点运算的时间复杂度，对于增加ECC算法的效率有十分重要的影响。

2.3.3 椭圆曲线上的离散数对问题

由椭圆曲线上的运算法则可以得出，当给定点P时，“已知数x求点xG的运算”不难，因为有加法的性质，运算起来可以比较快。但反过来，“已知点xG求x的问题”则非常困难，因为只能遍历每一个x做运算。这就是椭圆曲线密码中所利用的“椭圆曲线上的离散对数问题”。

2.3.4 基于椭圆曲线的DH密钥交换（ECDH）

ECDH全称是椭圆曲线迪菲-赫尔曼密钥交换（Elliptic Curve Diffie－Hellman key Exchange），主要是用来在一个不安全的通道中建立起安全的共有加密资料，一般来说交换的都是私钥，这个密钥一般作为“对称加密”的密钥而被双方在后续数据传输中使用。

ECDH的流程如下

1.A选定一条椭圆曲线E，并选取椭圆曲线上一点作为基点G。

2.A选择一个私有密钥k（k<n，n为G的阶数），并生成公开密钥K=kG。

3.A将E和点K、G传给B。

4.B收到信息后，选择一个随机整数r（r<n）。

5.B计算点R=rG，key = rK = rkG（key就是交换出的密钥）

6.B将R传给A。

7.A收到信息后，计算key = kR = krG

数学原理:key = rK = rkG = kR = krG

至此：A和B就协商出来相同的密钥，后面就可以使用这个密钥进行对称加密通信。攻击者只能获取中间在信道中传输的E和点K、G、R等，由于椭圆曲线的数学难题，无法计算出A和B的密钥k、r，那么协商出来的密钥key也就是安全的。

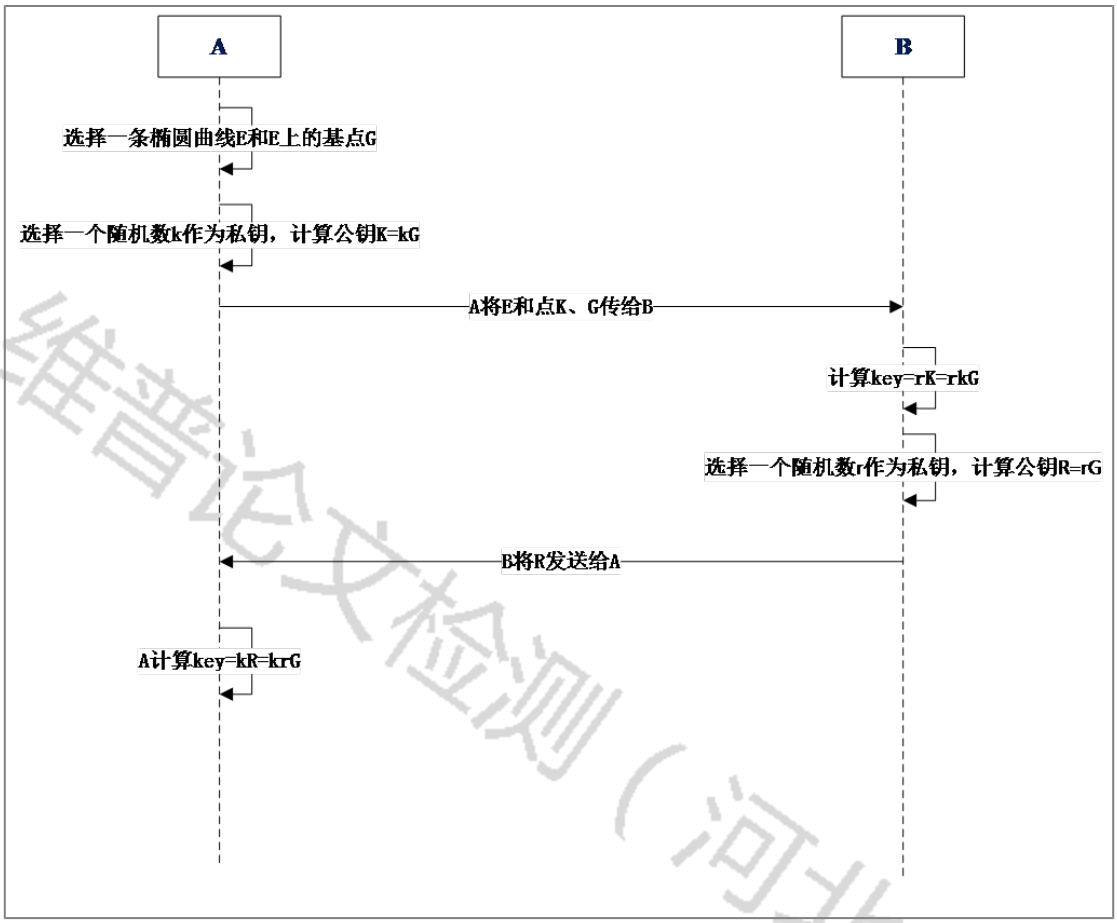


图 2.8 ECDH秘钥交换

2.4 AES加密算法

AES算法最初是由比利时密码学专家Joan Daeman和Vincent Rijmen设计和提出，又叫做Rijmen算法[10]。最开始被用来替代DES算法，AES为分组密码，分组的长度为128位，根据密钥的长度不同可以分为AES128、AES192、AES256三种，推荐加密的轮数也不同，本文使用的是AES128，每一种的加密轮数如下表

表 2.1 三种AES算法加密轮数

AES	密钥长度	分组长度	加密轮数
AES128	4	4	10
AES192	6	4	12
AES256	8	4	14

AES对软硬件的要求都相对较低，计算效率较高，容易在硬件设备上实施。AES采用轮加密的方式，加解密的关键步骤如下：

1.字节代换与字节逆代换

AES定义了一个S盒和一个逆S盒，字节代换与字节逆代换就是一个简单的查表操作，例如加密是就是把明文字节按照S盒进行映射，如下图所示

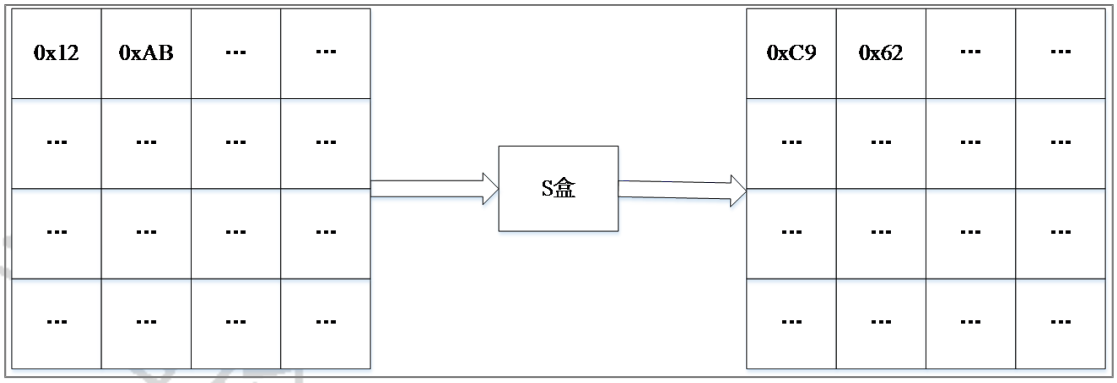


图 2.9 AES字节代换

2.行移位变换和行移位逆变换

行移位就是左循环移位操作。在AES128中，状态矩阵的第1行左移0字节，第2行左移1字节，第3行左移2字节，第4行左移3字节，如下图所示：

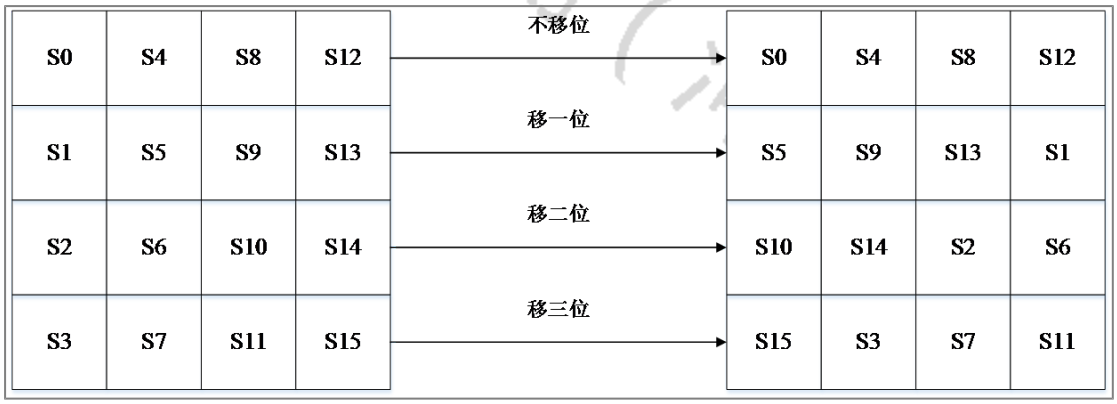


图2.10 AES行移位

行移位的逆变换就是行移位执行相反的操作。在AES128中，状态矩阵的第1行右移0字节，第2行右移1字节，第3行右移2字节，第4行右移3字节

3.列混合与列混合逆运算

列变换就是对状态矩阵中的列进行混合变换。列混合变换是通过式2-3进行矩阵相乘实现的，经行移位后的状态矩阵与固定的矩阵相乘，得到混淆后的状态矩阵。其中，矩阵元素的乘法和加法都是定义在基于GF(2^8)上的二元运算,并不是通常意义上的乘法和加法。逆变换矩阵同正变换矩阵的乘积恰好为单位矩阵。

$$\begin{bmatrix} S'_{00} & S'_{01} & S'_{02} & S'_{03} \\ S'_{10} & S'_{11} & S'_{12} & S'_{13} \\ S'_{20} & S'_{21} & S'_{22} & S'_{23} \\ S'_{30} & S'_{31} & S'_{32} & S'_{33} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix}$$

4. 轮密钥加

轮密钥加是将轮密钥同状态矩阵中的数据进行逐位异或操作，轮密钥是通过初始密钥和轮密钥产生算法共同产生的。因为异或的逆操作是其自身，轮密钥加的逆运算同正向的轮密钥加运算完全一致。

5. 密钥扩展

密钥扩展是从 初始密钥得到轮密钥的过程。首先将初始密钥放到一个4*4的状态矩阵中，如下图所示

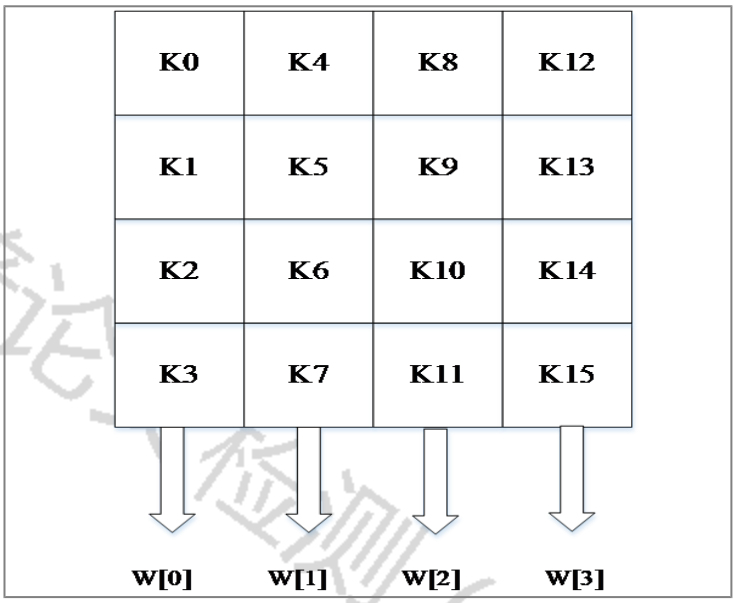


图2.11 初始密钥状态矩阵

矩阵的每一列的4个字节组成一个字，4个字一次命名为W[0]、W[1]、W[2]、W[3]，构成一个以字为单位的数组W，例如，初始密钥为“qwerasdfzxcvtgby”，则矩阵中的K0=q，K1=w，K2=e，K3=r，W[0]=qwer，接着对 W 数组扩充 40 个新列，即得到一个共 44 列的密钥扩展数组，密钥扩展过程如图 2-13 所示。

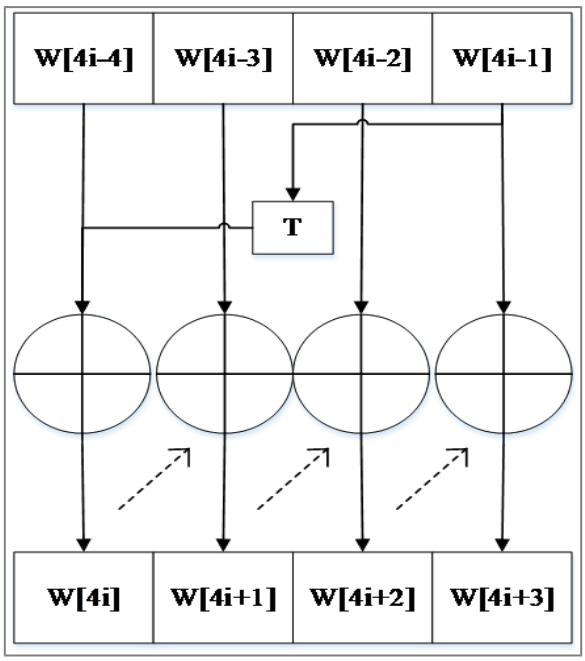


图2.12 密钥扩展过程

若i不是4的倍数，那么W[i]=W[i-4]W[i-1]，若i是4的倍数，那么W[i]=W[i-4]T(W[i-1])，其中，T是一个由字循环、字节代换和轮常量异或所组成的函数。

（1）字循环：将1个字中的4个字节循环左移1个字节。即将输入字[b0, b1, b2, b3]变换成[b1,b2,b3,b0]。

（2）字节代换：对字循环的结果使用S盒进行字节代换。

（3）轮常量异或：将前两步的结果同轮常量Rcon[j]进行异或，其中j表示轮数。轮常量Rcon[j]是一个字，Rcon[j]=（RC[i], 00,00,00,00），RC[i]的值如表 2-2所示。

表2-2 轮常量异或中的RC[i]

i	1	2	3	4	5	6	7	8	9	10
RC[i]	01	02	03	08	10	20	40	80	1B	36

6.AES加密解密的流程图如下

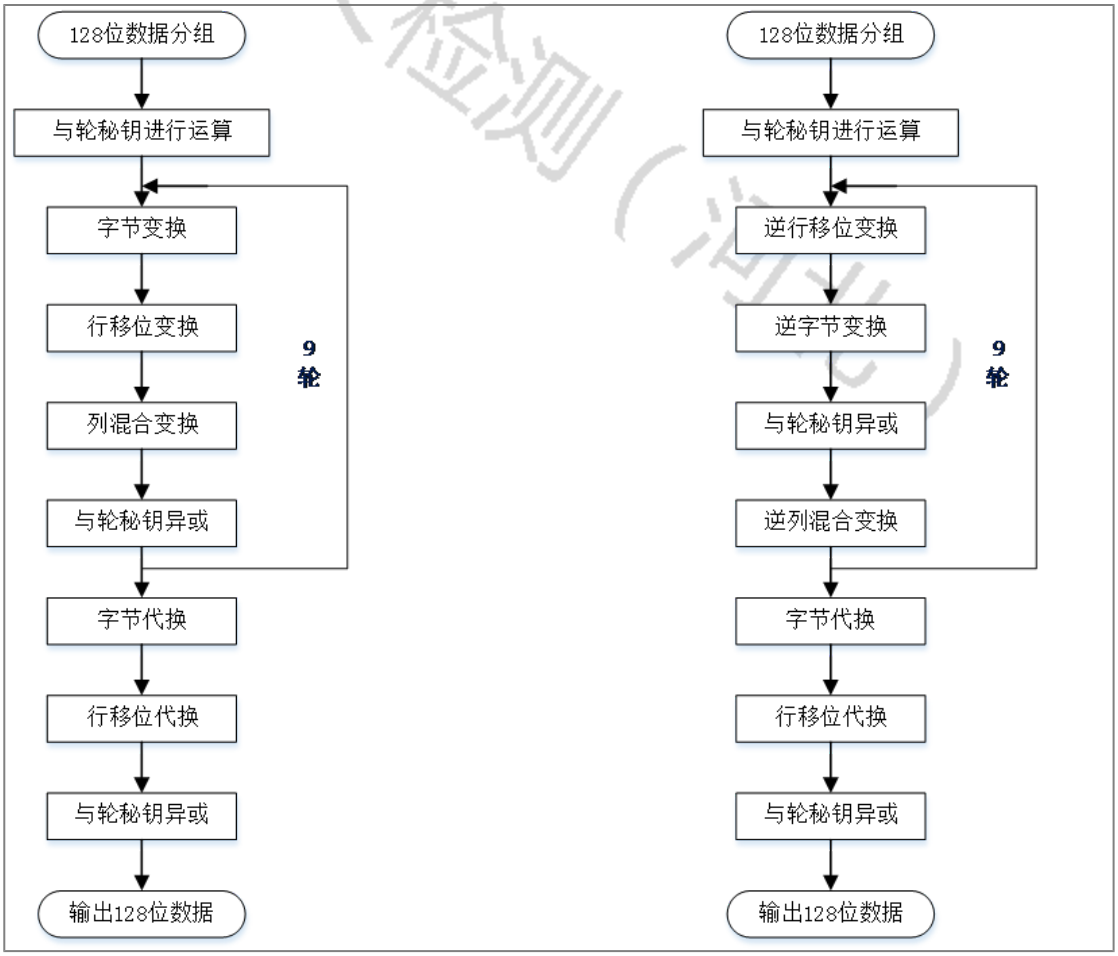


图2.13 AES加密流程图 图2.14 AES解密流程图

第3章 需求分析

3.1 可行性分析

本章主要介绍了该系统设计的需求和可行性分析，系统可行性分析的目的是：了解客户需求，市场未来发展趋势，确定项目是否值得开发；对项目功能、限制条件进行分析，在已有的硬件资源和技术条件下，确定项目是否能够实现。

3.1.1 技术可行性分析

技术上的可行性分析主要分析能否应用现有的技术完成对系统的设计。本设计利用开源的Netty框架实现网络通信，通过设计合理的数据结构来完成认证加密流程，使用ECC进行密钥协商、数字签名，使用AES进行加密解密。系统中应用到的技术已非常成熟，设计中遇到的难题可方便的查找相关资料，因此在技术上的分析是可行的。

3.1.2 经济可行性分析

本设计的核心在于运行于嵌入式设备上的认证和加密算法。其中软件部分的投入主要是软件开发和人力消耗，软件开发所使用的工具均为现有的免费工具，人力消耗可通过利用课余时间开发解决。硬件部分的投入如下表：

表3.1 硬件成本价格预算表

主要设备列表	数量	单价/元	合计/元
STM32F407芯片	1	7	7
电路板	1	30	30
外围器件	1	10	10
屏蔽双绞线	2	4	8
总计			55

从上表可以看出，单单从硬件成本计算的话，只有55元。因此本设计在经济上是可行的。

3.1.3 操作可行性分析

从操作上来讲，农业物联网的网关可以随机部署在农场的各个地方，随机性比较大，网关上电后，可自行与云服务器进行密钥协商与双向认证，后续的数据都是加密传输。用户可在PC查看当前以及历史的大棚温湿度数据，也可直接通过Web端直接控制大棚内的设备，例如电灯、风机等。简单方便，操作性强。

3.2 农业物联网安全体系及需求

农业物联网系统为农业生产提供高效、自动化和远程的管理服务，减少人工开销、增加产量、降低风险、提高监管水平。但是，多数系统没有深入考虑安全问题，存在安全隐患，如果出现问题，后果十分严重，例如：

1.传感器参数篡改，例如氨气浓度参数被篡改，导致错过报警时机

2.设备非法启动、停止，例如风机无故启动、湿帘无故关闭

3.传感器冒充，收到非法的监测数据

此外，考虑到成本、农业物联网系统的节点，设备一般采用计算能力有限、存储容量较小的芯片（比如，STM32），造成节点和设备难以或无法使用复杂的安全方案。因此，本文结合农业物联网设备处理能力有限的特性，提出一种轻量级的双向认证和加密方法。

以下为本文实现的主要功能

1.物联网网关与服务器之间密钥协商

物联网网关上电之后，会主动向服务器发送网关认证请求，网关与服务器之间使用基于椭圆曲线的密钥协商（ECDH），协商出网关与

服务器之间的数据传输密钥。

2.网关与服务器之间加密通信

网关与服务器之间的通信是全双工通信，网关向服务器上报数据时，使用之前协商出来的加密密钥进行AES128加密。网关收到服务器数据时，进行AES128解密。

3.连接管理

服务器上维护着一个网关的连接列表，可以进行批量的数据推送，网关流分析与监控等操作。

4.心跳管理

网关和服务器都要监听彼此之间的通信信道，当信道在一段时间内没有数据读写时，向对方发送心跳。服务器如果长时间没有接收到对方的心跳应答，会将对应的网关标记为下线，后续可以生成网关异常告警，自动生成工单并指派工程师维修等。而网关会标记服务器崩溃，并开启定时任务进行自动重连。

5.秘钥刷新

用户可以在web页面上点击按钮，刷新指定网关的加密密钥。

4章 基于ECC、AES的轻量级认证加密算法

4.1 ECC算法的性能瓶颈分析

与RSA等其他非对称加密方式相比较，在相同安全性的要求下，ECC算法使用更短的密钥就可以达到RSA相同的效果。但是与传统的AES、DES等对称加密体制比较，ECC算法的时间复杂度要高得多，这也是限制ECC算法发展的重要原因。而在本文研究的农业物联网安全通信中，其大部分设备都是计算资源有限的嵌入式设备，并且指令传输的实时性要求较高，如果直接应用传统的ECC算法来实现农业物联网中的双向认证和数据加密等，将对通信实时性造成较大影响。

根据前文对椭圆曲线算法的原理分析可知，在椭圆曲线中，最耗时的计算就是标量乘计算，即kG的计算，标量乘计算的效率决定了实现椭圆曲线密码体制的效率。

4.2 改进的ECC算法

传统的标量乘计算过程主要是包括点加运算和倍点运算，点加运算就是对不同点的相加运算，倍点运算就是对相同点的相加运算。阅读文献可知，现有的改进ECC标量乘运算方法有两类：

1.将整数k基于某种形式展开来表示，通过控制展开式中非零元素的个数，从而使得点加和倍点运算次数大幅降低。这种方法主要是对标量k的表示进行变换，将其变换到不同的表示域上进行运算，通过这种方法降低标量乘中的点加和倍点运算的次数，进而有效地减 少标量乘的运算量，提高其运算效率。例如双基链表示法[1]、二进制算法[9]等。

2.以空间换时间，增加算法的空间复杂度来降低时间复杂度，通过牺牲一点的存储空间进行预计算，存储与点G相关的计算结果，在后续的计算中通过查表的方式实现快速地计算kG，以降低点加和倍加运算的次数，提交运算效率。例如滑动窗口法及结合NAF方法的w-NAF窗口法[1]。

4.3 基于改进ECC-AES的混合加密方案设计

对称加密算法具有速度快、强度高、便于实现等特点，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。本文将对称加密算法(AES)和使用滑动窗口法改进的非对称加密算法(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。

4.3.1 密钥协商与网关认证

为了实现物联网网关和系统云认证服务器之间的双向认证和密钥协商，我们需要作出如下约定。

1.所有设备包括云认证服务器CAS，物联网网关之间共享一组公共的参数。其中包括一组EC参数E{a,b,p,G,n,h}；独立的身份ID和时

钟。身份标示ID用于在系统中唯一标识指定设备，其他设备需要通过ID访问指定的设备。设备时钟用于保证消息的新鲜。

2.云服务器和网关约定好了网关登录口令，用来验证密钥的正确性。

其基本流程如下。

第一步.云服务器初始化时，选取一个随机的大素数p，通过式（4.1）计算出ECC上的点P作为服务端的公钥，而p作为服务端的私钥。

$P = p \cdot G$	式（4.1）

第二步.网关上电时，选取一个随机的大素数r,通过式（4.2）计算EC上的点R作为网关的公钥，r作为网关的私钥。

$R = r \cdot G$	式（4.2）

第三步.网关初始化公钥私钥完成后，通过TCP协议连接云服务器，经过TCP三次握手建立连接，之后网关向云服务器发起认证请求，将网关的公钥R和网关ID发送到云服务器.

第四步.云服务器接收到网关的认证请求之后，首先验证网关ID的合法性,之后用一个hash映射保存网关的ID和公钥，用做后面数字签名的验签。云服务器使用自己的私钥和网关的公钥通过公式（4.3）计算出AES加密密钥K。然后将自己的公钥P发送给网关。

$K = p \cdot R = p \cdot r \cdot G$	式（4.3）

第五步.网关接收到云服务器的公钥之后，使用自己的私钥和云服务器的公钥，通过公式（4.4）计算AES加密密钥K。

$K = r \cdot P = r \cdot p \cdot G$	式（4.4）

第六步.网关使用密钥K加密登录口令并发送给云服务器，验证密钥的正确性。

第七步.云服务器接收到网关登录口令之后，使用密钥K进行解密，然后与事先约定的登录口令进行对比，如果不一致就关闭连接并删除密钥K，如果一致就把与网关的会话加入连接管理，保存密钥K并向网关发送登陆成功的消息。

第八步.网关接收到登陆成功的消息之后，保存密钥K，后续使用密钥K进行AES加密解密通信。

图4.1 网关与云服务器秘钥协商

4.3.2 数字签名与数据加密

由文献[11]中几种数字签名算法的分析比较可以得出结论，与RSA和DSA相比，无论是在安全性、运算速度和存储空间上，ECC算法都有着明显的优势。现代计算机都有着速度快，存储空间大的特点，在安全性要求不高的场景下，可以很好地应用RSA和DSA算法，但是在本文研究的农业物联网嵌入式设备中，网关的计算速度低、存储空间小，ECC算法的优势就显得更加重要，是更为理想的选择。

签名和加密的流程如下：

1. 发送方选择一个随机数k，k的范围是{1,...n-1}，n为基点G的阶。

2. 发送方计算P = kG，G是基点。

3. 发送方计算数字r = xp mod n ,xp 是P的x轴坐标，如果r=0，选择另一个k重新计算。

4. 计算明文摘要z = SHA1(m)，m是消息的明文。

5.

$$s = k^{-1}(z + rd_A) \bmod n$$

计算 ，其中dA是发送方的私钥，k-1是k mod n的乘法逆元。如果s=0，选择另一个k重新计算。

6. 使用AES128加密算法和协商出的密钥，对明文（m）进行加密，得出密文（w）。

7. 将消息的签名（r，s）与密文（w）发送给接收方

解密和验签的流程如下

1. 接收方在收到密文（w）和签名值（r，s）之后，使用AES128解密算法和之前协商出的密钥对密文进行解密，得到明文（m）。

2. 计算：sG+H(m)P = (x1,y1)，r1 = x1 mod p。

3. 验证：r1 = r mod p。

4. 如果等式成立，说明验签成功，数据没有被篡改过，否则签名无效，丢弃数据

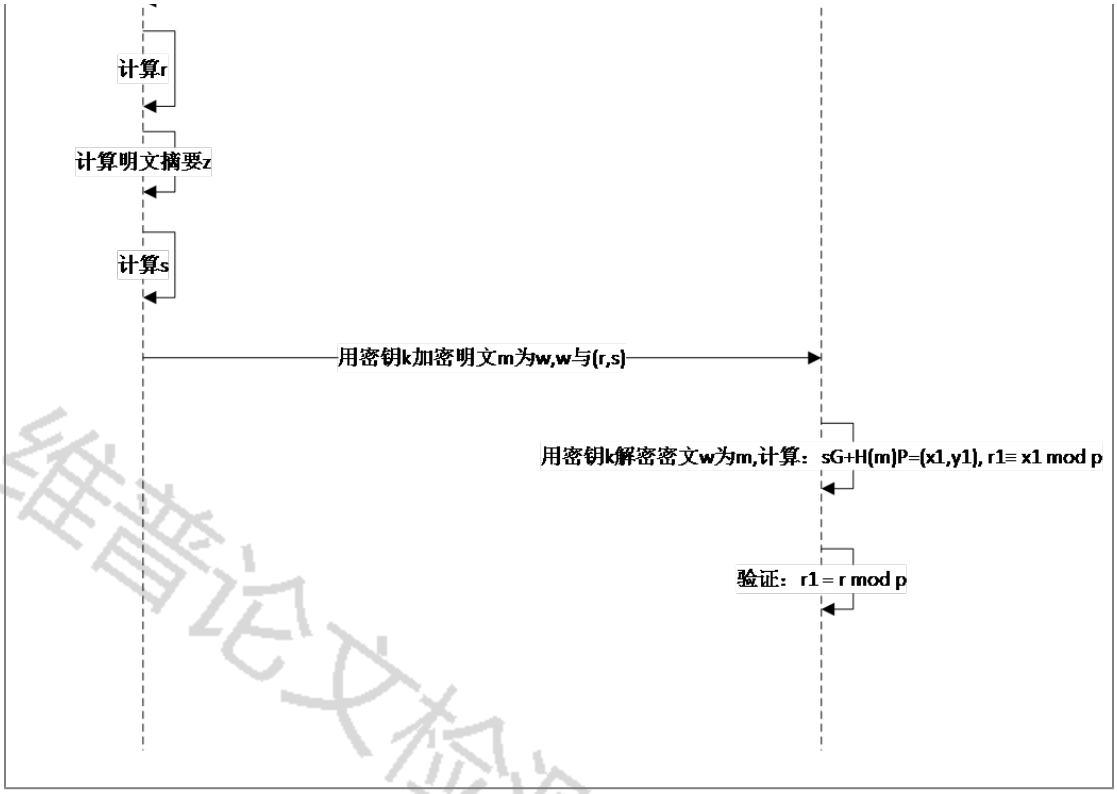
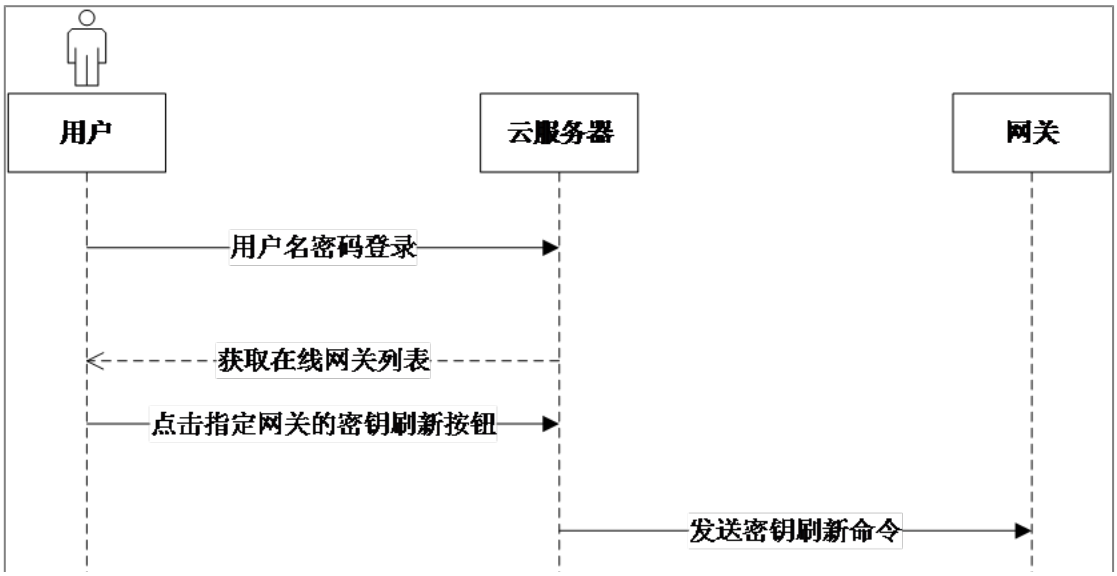


图 4.2 数字签名与验签

4.3.3 密钥更新

在安全的通信系统中，如果长期使用同一密钥，泄漏的可能性就会提高，通信系统的安全性就会下降。为了应对这个问题，需要在安全的通信系统中增加密钥更新机制，确保整体的安全性。本文设计的通信系统的密钥能够定期且根据需要进行更新，所以大幅度提高了通信系统的安全性，防止了密钥的泄露通信的安全性的损坏。密钥更新的流程如下：

1. 用户登录云服务器
2. 用户查询在线网关列表中的网关
3. 用户选择指定的网关，向云服务器发送更新网关密钥的请求
4. 云服务器验证用户和网关的合法性之后，向网关发送密钥更新命令
5. 网关接收到密钥更新命令之后，重新执行密钥协商和网关登录



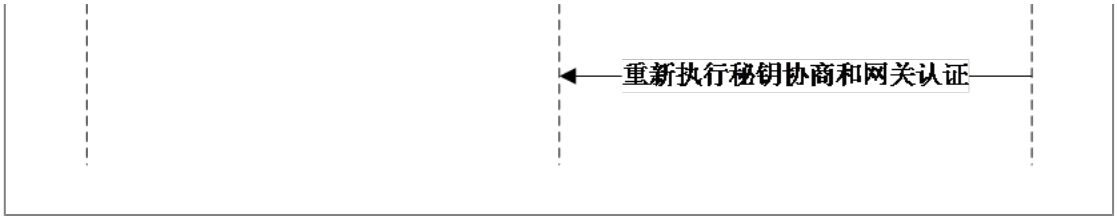


图 4.3 用户刷新网关秘钥

4.4 算法分析

4.4.1 安全性分析

1. 防窃听分析

云服务器与网关之间建立通信信道时，监听者可以获取公钥Kc-g，但无法得到双方的私钥k，即使获取密文C，也无法获取真实信息，因此协议具有防窃听能力。

3.防假冒分析。

系统中通信信息使用存有椭圆曲线 Ep(a,b) 、选定的基点G 和私钥t ，

其中私钥是保存在本地，不会在网络中传输，并且使用STM32芯片的读保护模式，使用芯片的全球唯一id作为网关ID，启动时与代码中的ID进行校验，一旦不同则进行flash数据擦除。窃听者无法伪造加密的信息，因此在本系统中，具有防假冒能力。

4.防重传分析。

重传攻击是窃听者在非法获取到网络中的通讯信息后，通过重发先前信息以非法获取信任的攻击手段。本系统消息中添加了时间戳、有效期和顺序号等信息，使得协议具有防重传的能力。

5.中间人攻击

根据攻击者没有办法伪装成其他用户进行通信，并且攻击者无法伪造出合法的认证信息，所以这个协议可以抵挡住中间人攻击。

4.4.3 能耗分析

通过算法能耗的理论分析，算法的能耗与算法时间的复杂度的变化之间存在密切的关系，时间复杂度越高，能耗越高。为此，本文针对 ECC、改进 ECC、RSA 三种加密算法的能耗展开了分析，其能耗分析结果从高到低排列依次是 RSA 算法、ECC 算法、改进 ECC 算法，可见改进 ECC 算法的能耗最低，计算结果最优。

第5章 实验与结论

5.1 实验环境及开发平台

5.1.1 硬件环境

本文在设计和实现物联网网关通信实验仿真过程中主要是通过在PC电脑与开发板之前模拟双向认证及安全通信的过程，使用的实验平台为采用意法半导体公司的STM32F407VET6微控制器的开发板，该实验开发板采用ARM 32位Cortex-M4内核CPU、最大频率为168mhz、拥有最多1mbyte的闪存、支持AES 128, 192, 256，三重DES，哈希(MD5, SHA-1)的硬件加速。在上面移植了uc-osII操作系统与lwIP协议栈。可以使用TCP协议，通过socket与服务器PC进行通信，完成密钥协商和加密数据的发送。服务器端运行在PC计算机上，该计算机使用64位windows操作系统，使用i5-7200U，2.50GHz的cpu，安装内存为8GB。配置128GB的固态硬盘。

5.1.2 软件环境

客户端的软件上使用C语言为开发语言，本文采用了开源的OpenSSL库作为加密解密和ECC算法的基础，将本文设计的ECC算法移植到OpenSSL库中，然后基于开源 OpenSSL库实现ECC各种 加密功能和密钥管理功能。服务端的软件上使用Java语言为开发语言，版本为Java8，使用开源的Netty作为网络通信的框架，实现Socket服务端的网络数据监听，使用bouncycastle作为Java加密解密库的扩展，实现了本文描述的ECC算法。

5.2 系统功能测试

整个设计方案所要实现的最终功能是能够实现物联网管关和云服务器之间密钥协商和数据加密传输，并通过Web页面进行在线网关和

密钥的管理。本系统的综合测试中，首先要对密钥协商和双向认证功能进行测试，协商出密钥之后，再对数据加密解密和数字签名进行测试，最后通过Web端登录后台管理中心，进行密钥更新测试。Web端页面截图如下：

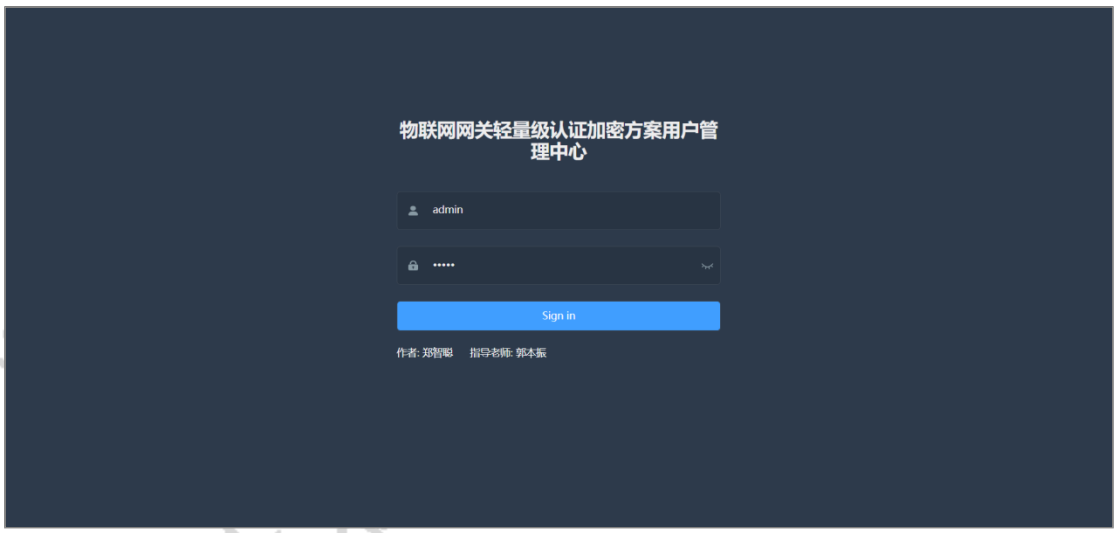


图 5.1 系统登录页

图 5.2 系统首页

5.2.1 密钥协商测试

在PC机上部署服务端，并启动网关客户端，在日志中打印网关与服务端进行密钥协商的过程。经过多次测试，在第一次进行密钥生成和密钥协商时，时间耗时在1400ms左右，后面刷新秘钥时，再次生成密钥和进行密钥协商，耗时在40ms左右，速度较快，对主要逻辑影响较小。运行截图如下：

```
: No active profile set, falling back to default profiles: default
: Bean 'org.springframework.hateoas.config.HateoasConfiguration' of type [org.springframework.hateoas.config.HateoasConfigu
: Tomcat initialized with port(s): 8081 (http)
: Starting service [Tomcat]
: Starting Servlet engine: [Apache Tomcat/9.0.17]
: Initializing Spring embedded WebApplicationContext
: Root WebApplicationContext: initialization completed in 3985 ms
: 启动 Netty 成功
: 服务端初始化公钥serPubKey【MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAET73HUlb+anKRCY/uBSgbwCMhW97/s8YbcqeDUpKO

: 服务端初始化私钥serPriKey【MIGTAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBHkwdwIBAQQgILfcmnPIihV6Hu4lwvSk/v5Uaz5B
```

图 5.3 秘钥协商过程

测试用例如下所示：

表 5.1 秘钥协商测试用例

用例描述	操作步骤	预期结果	实际结果
网关与云服务器进行密钥协商	1在PC机部署并初始化云服务器 2启动客户端程序	网关与云服务器协商出相同的密钥，云服务器管理网关连接列表。	符合预期

5.2.2 数字签名和加密解密测试

网关与云服务器协商出加密密钥，后续发送和就收数据都会使用AES128算法进行加密解密。同时，在通信过程中，为了保证消息不会被中间人篡改，通信双方都会保存对方的公钥，并且在发送数据时使用通过SHA算法对明文生成摘要，使用自己的私钥对摘要进行SHA1withECDSA数字签名；接收方接收到数据时，先使用对方的公钥验证数字签名，如果验签失败则直接丢弃数据，如果验签成功，说明数据没有被篡改，进行后续处理。运行截图如下：

图 5.4 数字签名和数据加密解密

测试用例如下：

表 5.2 数字签名和数据加密解密用例

用例描述	操作步骤	预期结果	实际结果
网关与云服务器通信时对消息进行加密解密和数字签名验签	1在PC机部署并初始化云服务器 2启动客户端程序	网关向云服务器发送经过AES加密和数字签名过的数据，云服务器AES解密数据和验签。云服务器向网关发送数据。	符合预期

5.2.3 心跳管理测试

网关与云服务器双向认证通过后，会加入心跳连接的机制，通过向对方发送心跳和心跳应答，确认对方是否在线。在云服务器端维护一个计数器，当网关三次心跳都没有应答时，判断网关下线，关闭服务器与网关的TCP连接。在网关的逻辑是，云服务器宕机连接被断开时，会启动一个定时任务，每30秒进行重连。运行截图如下：

图 5.5 心跳管理

测试用例如下：

表 5.3 心跳测试用例

用例描述	操作步骤	预期结果	实际结果
网关与云服务器之间维持心跳连接	1在PC机部署并初始化云服务器 2启动客户端程序	云服务器和网关检测socket读写空闲，空闲时向对方发送心跳，接收到心跳时发送应答	符合预期

网关心跳无应答，云服务器将其标为下线	1在PC机部署并初始化云服务器 2启动客户端程序 3阻塞客户端程序	云服务器发送三次心跳，都没有收到应答之后，将网关标记下线，关闭通信信道	符合预期
云服务器宕机，网关执行重连	1在PC机部署并初始化云服务器 2启动客户端程序 3手动关闭服务端程序	网关检测到TCP连接断开，判断云服务器宕机，执行定时重连	符合预期

5.2.4 密钥更新测试

用户在登录物联网网关轻量级认证加密方案用户管理中心后，点击在线网关模块，列表显示当前在线的网关，用户可点击网关后面的刷新密钥按钮，刷新指定网关的密钥。程序截图如下：

图 5.6 在线网关页面

图 5.7 密钥更新运行截图

测试用例如下：

表 5.4 密钥更新用例

用例描述	操作步骤	预期结果	实际结果
用户刷新指定网关的密钥	1用户登录物联网网关轻量级认证加密方案用户管理中心 2点击在线网关模块 3选择网关，点击密钥刷新按钮	提示密钥刷新成功，查看后台日志，密钥已经被更新	符合预期

5.3 系统性能测试

5.3.1 并发压力测试

现实世界中物联网网关的数量是成千上万的，云服务器需要能够支持大量的物联网网关并发地进行连接和数据发送，这对于云服务器的设计十分具有挑战性。本文设计的云服务器使用NIO的IO模型，单一线程就可以支持多个客户端连接，同时使用多线程技术，异步地执行逻辑。

测试用例如下：

表 5.5 并发测试用例

用例描述	操作步骤	预期结果	实际结果
云服务器并发压力测试	1在PC机部署并初始化云服务器 2 同时启动多个网关客户端进行身份认证和登录	云服务器能够快速响应，并且不会宕机	符合预期

测试结果如下：

表 5.6 并发压力测试结果

客户端并发连接数	服务端认证总时间	实际测试情况
10	154ms	服务器快速响应，没有卡顿现象
50	240ms	服务器快速响应，没有卡顿现象
100	512ms	服务器快速响应，没有卡顿现象
200	941ms	服务器响应较慢，没有卡顿
500	3102ms	服务器响应较慢，没有卡顿
1000	21043ms	服务器响应较慢，卡顿
3000	--	服务器出现拒绝连接的情况

测试结果：当前是在windows系统中进行测试，测试结果受CPU，内存，端口数量等限制。单机部署的情况下，合适的并发连接数在300左右，如果对性能要求较高，可以通过分布式部署、负载均衡等技术解决。

5.3.2 安全性测试

在云服务器与网关通信的过程中，进行网络抓包，截获通信过程中的数据包并进行分析，测试能否通过数据包中的信息破解出网关的秘钥。

运行截图如下：

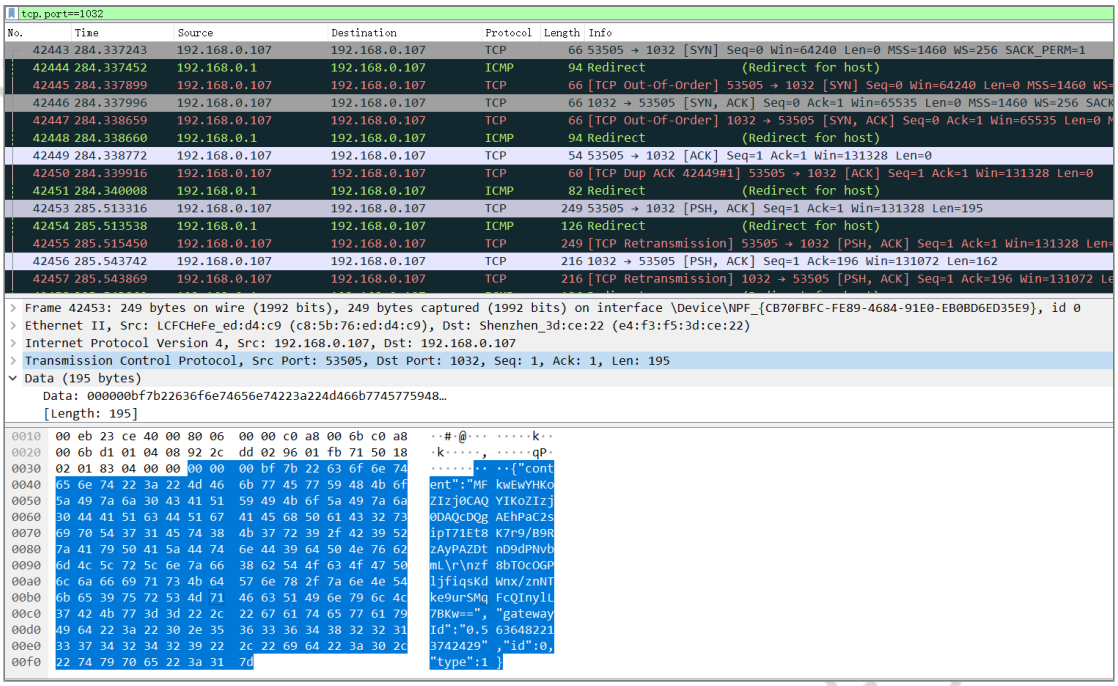


图 5.8 安全性测试抓包截图

测试用例如下：

表 5.7 安全性测试用例

用例描述	操作步骤	预期结果	实际结果
使用wireshark软件抓包，分析抓取的数据包	1部署云服务器端和网关客户端 2在通信过程中使用wireshark抓包 3分析抓取的数据包，看能否通过数据包破解秘钥	从网络数据包中无法破解出秘钥	符合预期

第六章结论

本文研究了在农业物联网的环境下，基于椭圆曲线ECC和AES128的物联网网关轻量级认证与加密方案，该系统的原理主要是基于椭圆曲线的离散对数问题进行密钥协商，基于AES128进行对称加密通信。系统实现的技术支持为Java8语言、Netty网络通信框架、Spring Boot框架开发实现的。系统中实现了物联网网关与云服务器之间进行密钥协商、非阻塞、IO多路复用的通信模型、数字签

名、心跳管理、密钥刷新等功能。可以作为一个高性能、轻量级的物联网认证与加密方案，该算法运行时占用资源较少，适合在嵌入式设备上运行，对具体的业务逻辑影响较小。

论文工作总结：

(1)需求分析与系统设计，阅读相关文献，调查物联网认证和加密系统的基本架构，分析了构建该系统的可行性，最终敲定了如何利用ECC和AES实现轻量级的认真和加密方案。

(2)通过设计基于椭圆曲线ECC和AES128的物联网网关轻量级认证与加密方案，将平时课堂上学习到的相关知识应用到了系统设计中，同时解决了一些课本上没有涉及到的开发问题，设计能力和技术水平都有了明显的提高。

(3)目前基于椭圆曲线ECC和AES128的物联网网关轻量级认证与加密方案的代码已按照正常进度开发完成，并实现了主要的功能，但系统设计中仍存在许多不足之处，需要继续完善。

谢辞

行文至此，已经到了论文的最后一步，四年的大学生涯即将结束。学生时代的生活是弥足珍贵的，因为时光是一张有去无返的单程票，关于北方学院有我带不走的记忆，更有带不走的收获。大学生活中最有价值的一件事就是在大二的时候加入信息技术研究所，在这里遇到了志同道合的老师和同学，学到了很多实际项目中的技术。

首先，感谢我的指导老师郭本振老师和科研室的王志辉老师。两位老师认真的工作态度和科学的引导给了我很大的帮助，没有两位老师的悉心指导，就没有我的这个毕业设计。两位老师在项目设计期间，给出了许多中肯的建议，在论文修改期间，针对论文中的不足之处悉心指导，在此我深表敬意。

其次，还要对大学期间每位任课老师和同学表达谢意，是你们陪我度过了人生中宝贵的四年，你们的帮助和陪伴使我不断成长和进步，让我度过了充实且快乐的大学时光。

最后感谢各位评审老师，能来参加我的毕业答辩，给我提出珍贵的建议，在此表示深深的感谢。

参考文献

[1] 汪洋. 物联网轻量级认证和加密技术研究[D].南京邮电大学,2017.

[2] Nicanfar H, Leung V C M. Multilayer Consensus ECC-Based Password Authenticated Key-Exchange (MCEPAK) Protocol for Smart Grid System[J]. IEEE Transactions on Smart Grid, 2013, 4(1):253-264

[3] Li D, Aung Z, Williams J R, et al. Efficient and fault-diagnosable authentication architecture for AMI in smart grid[J]. Security & Communication Networks, 2015, 8(4):598-616.

[4] Q. Jiang, Y. Qian, J. Ma, X. Ma, Q. Cheng, and F. Wei, “User centric three-factor authentication protocol for cloud-assisted wearable devices,” Int J Commun Syst, vol. 32, no. 6, p. e3900, Apr. 2019.

[5] ZHIHUI WANG, JIANLI ZHAO, BENZHEN GUO, JINGJINGYANG, XIAO ZHANG. Mutual Authentication Protocol for IoT-based Environment Monitoring System[j]Journal of Environmental Protection and Ecology 2019,6(2)

[6] 袁琦. 我国物联网安全发展现状和建议[J]. 现代电信科技, 2014(10):36-39.

[7] 史冰清. 高安全性的物联网网关设计与实现[D].电子科技大学,2018.

[8] 王斌. 工业物联网信息安全防护技术研究[D].电子科技大学,2018.

[9] 黎俊男. 基于AES与ECC的游戏数据混合加密研究与实现[D].华南理工大学,2018.

[10]龙辉. 基于ECC-AES混合加密的智能配电网安全通信方案设计[D].湘潭大学,2016.

[11]李春平. 基于低功耗处理器的数字签名研究与实现[D].北京邮电大学,2015.

附录

```
@Slf4j
public class ServerAuthHandler extends ChannelInboundHandlerAdapter {
@Override
```

```
public void channelActive(ChannelHandlerContext ctx) {
log.info("客户端【{}】连接，开始认证",ctx.channel().remoteAddress());
}
@Override
public void channelRead(ChannelHandlerContext ctx, Object message) {
SocketMsg msg = (SocketMsg) message;
try {
switch (msg.getType()){
case MsgType.AUTH_VALUE:
String clientPubKey = (String) msg.getContent();
String gwId = msg.getGatewayId();
if (StringUtils.isEmpty(clientPubKey) || StringUtils.isEmpty(gwId)){
log.info("网关公钥或网关ID为空，拒绝处理");
closeChannel(ctx);
return;
}
long start = System.currentTimeMillis();
log.info("2.服务端接收到秘钥协商请求,客户端公钥为：{}",clientPubKey);
EcKeys ecKeys = SpringBeanFactory.getBean("EcKeys",EcKeys.class);
SocketMsg<String> publicKeyStr = new SocketMsg<String>()
.setId(1).setType(MsgType.AUTH_BACK_VALUE).setContent(ecKeys.getPubKey());
String key = EncryptOrDecryptUtil.ecdhKey(ecKeys.getPriKey(),clientPubKey);
log.info("3. 服务端秘钥协商完成，耗时：{}ms，开始验证秘钥正确性,加密秘钥为:{}",System.currentTimeMillis()-start,
key);
//将改秘钥加入秘钥库
Gateway gw = new Gateway()
.setGwId(msg.getGatewayId())
.setGwName("第一套")
.setChannel(ctx.channel())
.setKey(key)
.setPubKey(clientPubKey);
NettySocketHolder.put(msg.getGatewayId(), gw);
ctx.writeAndFlush(publicKeyStr);
break;
case MsgType.AUTH_CHECK_VALUE:
log.info("6.服务端验证秘钥正确性");
Gateway gw1 = NettySocketHolder.get(msg.getGatewayId());
//确认秘钥的正确性
String login = EncryptOrDecryptUtil.doAES((String) msg.getContent(),gw1.getKey(), Cipher.DECRYPT_MODE);
```

```
log.info("7.服务端解密出登录口令:{}",login);
if (("login").equals(login)){
log.info("网关【{}】验证通过，加入连接列表",msg.getGatewayId());
gw1.setStatus("在线");
NettySocketHolder.put(msg.getGatewayId(),gw1);
//向客户端发送认证成功的消息，这里一定要先发消息再替换handler
ctx.writeAndFlush(new SocketMsg<String>().setType(MsgType.LOGIN_SUCCESS_VALUE));
//用AES加解密替换掉默认的编解码器
ctx.pipeline().replace(ctx.pipeline().get("decoder"),"decoder",new JsonDecoderAES());
ctx.pipeline().replace(ctx.pipeline().get("encoder"),"encoder",new JsonEncoderAES());
ctx.pipeline().addFirst("idle",new IdleStateHandler(5,0, 0, TimeUnit.SECONDS));
}else {
log.info("非法客户端，关闭连接");
closeChannel(ctx);
}
break;
default:break;
}
}catch (Exception e){
e.printStackTrace();
log.error("客户端认证出错{}",e.getMessage());
closeChannel(ctx);
}
ctx.fireChannelRead(message);
ReferenceCountUtil.release(msg);
}
private void closeChannel(ChannelHandlerContext ctx){
NettySocketHolder.remove(ctx.channel());
ctx.channel().close();
}
}
```

- **说明:**

相似片段中"综合"包括:

《中文主要报纸全文数据库》 《中国专利特色数据库》 《中国主要会议论文特色数据库》 《港澳台文献资源》 《图书资源》 《维普优先出版论文全文数据库》 《年鉴资源》 《古籍文献资源》 《IPUB原创作品》

- **声明:**

报告编号系送检论文检测报告在本系统中的唯一编号。

本报告为维普论文检测系统算法自动生成，仅对您所选择比对资源范围内检验结果负责，仅供参考。

客服热线：400-607-5550 | 客服QQ：4006075550 | 客服邮箱：vpcs@cqvip.com

唯一官方网站：http://vpcs.cqvip.com

关注微信公众号