

中图分类号: TP391.42  
密 级: 公开

单位代号: 10280  
学 号: 18721612

---

上海大学



# 专业学位硕士学位论文

---

SHANGHAI UNIVERSITY  
MASTER'S DISSERTATION

题 目	智慧燃气数据边界安全 监控的研究与实现
--------	------------------------

作 者 高 萌

学科专业 电子与通信工程

导 师 冯玉田 副教授

完成日期 2020 年 5 月

## 上海大学

本论文经答辩委员会全体委员审查，  
确认符合上海大学硕士学位论文质量要求。

答辩委员会签名：

主任：

委员：

导 师：

答辩日期：

姓 名： 高萌

学号： 18721612

论文题目： 智慧燃气数据边界安全监控的研究与实现

## 原 创 性 声 明

本人声明：所呈交的论文是本人在导师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名： \_\_\_\_\_ 日 期： \_\_\_\_\_

## 本论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（保密的论文在解密后应遵守此规定）

签 名： \_\_\_\_\_ 导师签名： \_\_\_\_\_ 日期： \_\_\_\_\_

上海大学工学硕士学位论文

智慧燃气数据边界安全  
监控的研究与实现

姓 名： 高 萌

导 师： 冯玉田 副教授

学科专业： 电子与通信工程

上海大学通信与信息工程学院

2020 年 5 月

A Dissertation Submitted to Shanghai University for the  
Degree of Master in Engineering

# **Research and Implementation of Smart Gas Data Boundary Security Monitoring**

MA Candidate: Gao Meng

Supervisor: Feng Yutian

Major: Electronics and Communication Engineering

**Communication and Information Engineering College,**

**Shanghai University**

**May, 2020**

## 摘 要

燃气是日常生活中不可缺少的重要能源，随着无线通信和网络技术的快速发展以及人工成本的逐年提高，自动抄表技术正逐步取代人工抄表，智能燃气表逐渐进入千家万户。目前常见的自动抄表方式有无线射频智能燃气表、物联网燃气表、智慧燃气表以及自助抄表等，这些抄表方式都需要将燃气数据上传到远程服务器端。如何确保燃气数据的安全性和燃气供应的稳定性已成为保障社会经济发展的重要问题。虽然防火墙和云计算服务在很大程度上提高了数据的安全性，但安全漏洞依然存在，在设备维护、网络管理等方面仍然存在许多亟待解决的问题。因此，研究燃气数据边界安全控制方法具有重要意义。

本文针对燃气自动抄表系统，对燃气数据的边界安全控制方法进行研究，以从根本上解决燃气信息系统的安全隐患，进一步加强网络安全防护和确保燃气系统的安全有效运行。本文的研究内容主要包括以下几个方面：

一、对燃气自动抄表系统的数据安全性进行分析研究，针对手持式燃气抄表系统，提出智慧燃气数据边界安全监控总体解决方案。该方案在燃气数据服务器的前端配置一套基于嵌入式系统的数据边界安全监控设备，用三层过滤方法对客户端上传到服务器的数据包进行过滤，将符合协议的合法数据包转发给服务器，而将不合法的数据包阻挡在服务器之外，使客户端和服务器之间形成一道安全屏障，避免服务器受到攻击，从根本上提升燃气数据的安全性。同时设计了基于嵌入式系统的数据边界安全监控装置的硬件，说明了监控软件的组成，并设计了实现数据包过滤的软件流程。

二、对深度包过滤技术进行分析研究。根据智慧燃气数据边界安全监控方案，采用模式匹配法对数据进行过滤。在分析研究多种模式匹配算法的基础上，提出了一种新的改进算法—AC\_HA 算法。该算法沿用了 BM 算法中的坏字符规则，对传统的 AC 算法进行文本串缩短和增大跳跃距离方面进行改进，减少匹配阶段中的匹配次数，从而减少匹配过程所消耗的时间。在仿真实验中，将该算法与传统的 AC 算法以及 AC\_BM 算法做性能对比分析，结果表

明，该算法在运行时间上相比于 AC 算法有明显提升，相较于 AC\_BM 算法有一定优势。

三、 数据加密算法的研究和实现。对现有的 RAS, AES, DES 等加密算法进行分析比较，由于 AES 算法具有密钥长度合适、安全性高、运行速度快、消耗资源少等优点，因此采用了 AES 算法作为系统的数据加密算法。对 AES 的字节替换，行移位、列混淆以及轮密钥加和密钥扩展等算法结构进行了深入研究，在此基础上，通过软件编程实现了 AES 算法，实验验证了算法的正确性。

四、 完成了数据边界安全监控装置的硬件制作和调试，设计了相关硬件驱动程序，将改进的 AC\_HA 模式匹配算法和 AES 数据加密算法移植到嵌入式系统平台上。对系统的各部分功能进行了测试，包括网络通信功能和数据包过滤功能的测试。实验结果表明，系统只将通过三层过滤的合法数据包上传到服务器，而将文件名不符、不含特征信息的数据包滤除，也能够将多次发送非法数据的 IP 地址拦截。实验结果验证了本文设计的数据边界安全监控方法的有效性。

**关键词：**数据边界安全；深度包过滤；嵌入式系统；加解密技术

## ABSTRACT

Natural gas is an indispensable and important energy source in our daily life. With the rapid development of wireless communication and network technology as well as the increase in labor costs year by year, automatic meter reading gradually takes the place of manual meter reading, thus leading to the widely-spread application of smart gas meters. Common automatic meter reading methods, such as wireless radio frequency, internet of things, smart gas meters, and self-service meter reading, are required to transfer the collected data to a remote central database for billing, troubleshooting, and analyzing. Issues concerned with the safety of gas data and the stability of gas supply are significant for social and economic development. Although cloud computing services have greatly improved the security of data, security vulnerabilities still exist. Moreover, problems involving equipment maintenance and network management still remain to be solved. Therefore, it is of great significance to study the data boundary security control methods.

In this contribution, the boundary safety control method of gas data is systematically studied for the automatic gas meter reading to eliminate the potential safety risks of the gas information system, which further strengthens the network security protection and ensures the safe operation of the gas system with high efficiency. The main contents of this research are shown as follows:

1. The data security of the automatic gas meter reading system is analyzed. An overall solution of smart gas data boundary security is proposed to monitor the hand-held gas meter reading system. This solution configures a set of data boundary security monitoring equipment based on an embedded system at the front of the gas data server, employing a three-layer filtering method to screen the data packets uploaded from the client to the server. In other words, the legal data packets that meet the protocol standard are transmitted to the server, while the illegal data



packets are blocked. As a result, a security barrier is formed between the client and the server, which prevents the server from being attacked and guarantees the safety of the gas data. The hardware of the data boundary security monitoring device based on the embedded system is also designed. Moreover, a software flow is developed to achieve the filtration of the data packets with the composition of the monitoring software explicated.

2. The filtering technology of the deep packets is researched. According to the smart gas data boundary security monitoring scheme, the pattern matching method is used to filter the data contents. Based on the analysis of multiple pattern matching algorithms, an improved algorithm — AC\_HA algorithm is innovatively proposed. The algorithm follows the rules of bad characters in the BM algorithm to promote the traditional AC algorithm by shortening text string and increasing step size. Hence, time is saved in the matching phase through the decrease in the number of matching processes. In the simulation experiments, the performance comparison between the novel AC\_HA algorithm and the traditional AC algorithm or the AC\_BM algorithm shows that the proposed algorithm completes in a significantly reduced running time compared to the AC algorithm and has certain advantages over the AC\_BM algorithm.

3. The data encryption algorithm is implemented in the research. The widely used encryption algorithms (e.g., RAS, AES, DES) are compared. The AES algorithm is selected as the system data encryption algorithm for its appropriate key length, high security, fast operation speed, and low resource consumption. The algorithm structures of AES byte replacement, row shift, column obfuscation, round key addition, and key extension are systematically studied. Next, the AES algorithm is realized through software programming and the correctness of the algorithm is verified.

4. The data security boundary monitoring is completed by the hardware production with the device debugged. Notably, the improved AC\_HA pattern

matching algorithm and AES data encryption algorithm are transplanted to the embedded system platform and the related hardware drivers are designed. The functions of each part of the system is tested, including the network communication and the data packet filtering. The experimental results demonstrate that the system only uploads the data packets that pass the three-layer filtering to the server, while the data packets that do not match the file name or contain the feature information are blocked. Furthermore, the IP addresses that send illegal data for multiple times can be also intercepted. Such findings verify the effectiveness of the data boundary security monitoring method designed in this work.

**Keywords:** data boundary security; deep packet filtering; embedded system; encryption and decryption technology

## 目 录

摘 要.....	VI
ABSTRACT.....	VIII
目 录.....	XI
第一章 绪 论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	3
1.2.1ARP 欺骗攻击.....	3
1.2.2CC 攻击.....	4
1.2.3DDoS 攻击.....	5
1.3 研究内容和结构安排.....	6
第二章 系统方案设计.....	8
2.1 安全监控方案设计.....	8
2.1.1 安全监控方案.....	8
2.1.2 智能终端数据预处理.....	9
2.1.3 数据包过滤方案.....	10
2.2 硬件设计.....	11
2.2.1 硬件整体结构.....	11
2.2.2 控制器电路设计.....	12
2.2.3 网络接口设计.....	15
2.2.4 存储器电路设计.....	17
2.2.5 串口通信接口设计.....	18
2.3 软件流程.....	18
2.4 本章小结.....	21
第三章 深度包过滤技术研究.....	22
3.1 引言.....	22
3.2 单模式匹配.....	23
3.2.1 基于前缀匹配的方法.....	23

3.2.2 基于子串匹配的方法.....	24
3.2.3 基于后缀匹配的方法.....	24
3.3 多模式匹配.....	25
3.3.1AC 算法.....	25
3.3.2AC 算法的改进算法.....	27
3.4 AC_HA 算法.....	28
3.4.1 算法流程.....	29
3.4.2 算法举例.....	31
3.5 实验结果与分析.....	33
3.5.1 实验一：模式串数量对算法性能的影响.....	33
3.5.2 实验二：文本大小对算法性能的影响.....	34
3.6 本章小结.....	35
<b>第四章 加解密技术研究.....</b>	<b>36</b>
4.1 引言.....	36
4.2 对称加解密算法.....	36
4.2.1DES 算法.....	37
4.2.2AES 算法.....	38
4.3 非对称加解密算法.....	39
4.3.1RAS 算法.....	40
4.3.2 其它加密算法.....	42
4.4 加密方案.....	42
4.4.1 加密算法比较.....	42
4.4.2AES 算法的实现.....	43
4.5 加解密算法的软件实现.....	46
4.6 本章小结.....	47
<b>第五章 系统测试及实验结果.....</b>	<b>48</b>
5.1 网络通信测试.....	49
5.1.1 实验一：手机与边界设备通信.....	49

5.1.2 实验二：边界设备与服务器通信..... 50

5.2 数据包加密测试..... 51

5.3 特征字匹配测试..... 53

5.4 IP 地址过滤实验..... 55

5.5 实验结果分析..... 56

5.6 本章小结..... 57

第六章 总结与展望..... 58

6.1 总结..... 58

6.2 展望..... 59

参考文献..... 61

作者在攻读硕士学位期间的科研成果..... 64

致 谢..... 65

# 第一章 绪 论

## 1.1 研究背景与意义

燃气是人们生活和生产不可缺少的一种重要能源，随着社会的不断进步，燃气已逐渐成为主要燃料之一，越来越多的家庭和企业使用燃气。在燃气公司的日常运行中，除了保障燃气的正常供应外，对用户燃气的抄表和计费是一项重要内容。传统的人工抄表方式效率低下，抄表人员劳动强度大，而且人工成本高，在很大程度上影响了燃气公司的效益。从用户的角度考虑，抄表和定点缴费的方式与现代掌上 APP 支付时代格格不入，大大降低了用户体验。传统抄表方式的淘汰成为了一种必然的趋势<sup>[1]</sup>。为了适应现代生活，自动抄表技术应运而生，它具有实时性强、可靠性强、数据准确等特点<sup>[2]</sup>。上世纪 90 年代初，我国开始实施自动抄表技术<sup>[3]</sup>，燃气抄表系统逐步由纯人工手工抄表到智能自动抄表模式的转变。自动抄表的基础是智能燃气表，自动抄表发展至今，已产生了多种抄表方式<sup>[4][5]</sup>：1) 无线射频智能燃气表<sup>[6]</sup>，它通过短距离无线通信，将很多用户燃气表数据定时发送到一个集中器，再由集中器通过移动网络或有线网络将用户数据发送到远程服务器。2) 专用抄表器抄表，由抄表员到小区楼下，用专用抄表器通过无线通信读取用户燃气数据，再通过手持终端设备（如手机）将数据上传到服务器。这一方式不需要抄表员到用户家中抄读，使用户免受打扰。3) 基于物联网的燃气抄表<sup>[7]</sup>，这种方式是在燃气表上安装 NB-IoT 远传模块，用户燃气数据通过 NB-IoT 物联网上传到服务器。4) 智慧燃气抄表，其核心是智慧燃气表，它除了具有数据远传功能外，还能统计用户用气习惯，并自动判断当前燃气安全状态，发现安全隐患能发出报警信息，这也是智慧燃气的一个发展方向。5) 用户自助抄表，即由用户自助读取燃气表数据，通过手机 APP 或其它手持机将数据或燃气表照片发送到服务器。

以上这些自动抄表技术都有一个共同的特点——抄表数据必须上传到远端服务器上，因此保障服务器和数据库的安全至关重要。随着网络技术和

网络环境日益复杂，燃气数据的安全面临着前所未有的挑战。如有不法分子将攻击服务器作为“突破口”进行网络攻击谋取非法利益。例如 2014 年，西班牙的三大供电服务运营商所提供的智能电表中有 30% 以上被检测到存在严重的安全漏洞<sup>[8]</sup>。攻击者可以利用这些漏洞盗取电费，甚至导致关闭电力系统，这对社会产生了巨大影响。诸如以上的案例频频发生，使得网络安全问题成为了燃气公司和城市居民不可忽视的问题。在此类事件背后，暴露出的是服务器存在巨大的安全隐患。不仅仅是智能燃气表，还有智能电表，智能水表等等，都面临着相同的问题。在中国这个人口密集，居民众多的国家，每家每户“损失”一点电费、水费、燃气费，甚至是泄露了个人隐私信息，都会对用户个人以及整个水、电、燃气系统造成无法估量的损失。

在如今的网络时代背景下，网络中的海量信息好比一个巨大的宝藏，总有不法分子想要抓住其中的安全漏洞对服务器进行攻击获利，手段也层出不穷，例如最为常见的就有 ARP 地址欺骗攻击，CC 攻击等等。但是面对互联网发展所必须面临的挑战，我们必须一一攻克，才有权利享受其为我们生活带来的便捷<sup>[9]</sup>。因此提升各种应用系统的安全性是必不可少，至关重要的。目前许多应用已经有了对应的安全防护措施，例如数据加密、控制访问权限和添加安全证书等等。但远端燃气抄表系统发展的时间较短，普及较快，许多安全技术还未成熟，让某些不法分子有利可图。而攻击者往往善于发送大量不合规的数据包攻击服务器，在没有安全防护下的服务器很容易被“击垮”，这对用户和燃气公司造成了很大的损失。所以提升燃气系统的安全性不仅对用户，也对燃气公司乃至其他相似的应用都有重要的意义。虽然很多应用系统借助互联网公司的云计算平台（如腾讯云、阿里云、华为云等），在数据中心设立服务器，由互联网公司的专业团队对服务器进行管理，通过各种防火墙等措施维护数据安全。但防火墙只能对已知的攻击有效，对新型的攻击显得无能为力，安全漏洞依然存在。总之，随着燃气自动抄表方式的普及，燃气系统面对繁杂的网络攻击时，拥有一套独立的数据边界安全监控系统来确保燃气数据的安全就显得尤为重要。

## 1.2 国内外研究现状

自动燃气抄表方式是现代化社会发展的产物，其采用的是后台计费模式，具有数据集中管理的特点。大部分的燃气表采用的是纯软件的安全设计，面对繁杂的网络环境，只能简单的抵挡基本的软件攻击，且容易被攻破。其所面临的网络安全问题主要是应用层安全问题。应用层能为智能燃气表提供服务，也可以充当中间件、通信协议的云计算，其所面临的安全威胁包括信息窃取，对 Web 服务的僵尸攻击等等。

防止网络攻击的关键在于面对网络攻击时能有相应的防御措施，而目前常见的网络攻击方式有三种，分别是 ARP 欺骗攻击、CC 网络攻击以及 DDoS 网络攻击。本小节将对这三种网络攻击方式进行分析。

### 1.2.1 ARP 欺骗攻击

ARP 网络攻击方式是利用地址解析协议（Address Resolution Protocol，ARP）的漏洞来达到网络攻击的目的，而 ARP 协议是通过 IP 地址寻找对应的 MAC 地址<sup>[10]</sup>。而攻击者通过伪装成其他主机的方式来欺骗 ARP 传输协议，从而接近目的主机获取数据并实现数据传输。

通常用来应对 ARP 欺骗攻击的防御措施有以下几种方式：1）绑定 IP 地址和 MAC 地址；2）绑定路由器，配置静态网关地址；3）打开 ARP 防火墙，保证能明确数据传输送达的目的地；4）定期使用杀毒软件进行杀毒。具体方法有 Moon D 等人提出的一种基于路由跟踪的网络安全系统，系统利用检测、保护和恢复技术防止内部网络中的 ARP 欺骗攻击，同时通过定期监视 ARP 表和路由跟踪来确认是否发生 ARP 欺骗攻击。该方法通过定时检测来确认 ARP 攻击是否产生，算法易实现，但是防御效果不佳，并不能确保重要数据的安全性<sup>[11]</sup>。Prabadevi B 等人针对 ARP 缓存表攻击方式，提出了将 ARP 和以太网报头中的 IP-MAC 对进行比较，检测是否存在任何虚假信息，若有更新缓存表，同时向网关或路由器发送警告消息，以此免受缓存中毒攻击。该方法在一定程度上可以实现对 ARP 攻击的识别，但是计算量相对较大<sup>[12]</sup>。Kaur G 等人通过启



用 DHCP 侦听和动态 ARP 检查来实现 ARP 中毒及其缓解机制<sup>[13]</sup>。为防止 ARP 攻击导致数据泄露，Jeong Y S 等人提出了一种基于密文访问控制的方法，通过秘密共享技术对重加密的过程单独执行，该方法需要的计算资源虽少，但是并不能有效准确的实现数据泄露问题<sup>[14]</sup>。程艳艳利用动态指纹检测功能，对 ARP 攻击后会造成泄露风险的敏感数据进行初步检测，并采用概率检测的方式减小计算开销，最终防止网络数据泄漏，该方法整体性能较好，但是依旧存在检测漏失的风险<sup>[15]</sup>。Chae C J 等人通过检测到的网络消息来获取 ARP 攻击之后的网络间的传输数据，并获取用户与网络之间的传输文件，而只有合法用户才能访问传输文件，该方法能够有效防止网络数据泄漏，但是计算过程相对复杂<sup>[16]</sup>。

目前，APR 欺骗攻击依旧是最常见的攻击方式，最主要的原因在于 ARP 协议自身存在的漏洞，针对漏洞实现在不可信网络中运行，最终导致出现网络安全的问题。目前针对 ARP 攻击的防护措施中最普遍使用的是让 IP 与 MAC 地址绑定，但由于此方法在大型局域网的应用中会给网络管理造成过程繁琐的问题，会采用 MAC 地址中央管理的方式进行改进，但是由于 ARP 协议的自身不足，很难从根本上解决实际问题。

### 1.2.2CC 攻击

CC 攻击的实质在于攻击者利用大量的主机不断地向服务器端发送 HTTP 请求，这使得服务器需要连续验证 IP 地址以及处理连接的请求。然而，服务器的资源是有限的，若有大量客户端同时发送请求，必将会导致服务器的资源消耗殆尽，最终造成服务器无法正常工作，从而不能再继续接受客户端的请求。因此，CC 攻击的目的并不是窃取数据并篡改数据，而是妨碍服务器正常运转<sup>[17]</sup>。

应对 CC 攻击的较为容易实现防御措施有：1) 为防止某用户对 IP 频繁发送请求，利用 Session 对话对访问 IP 的用户进行访问次数统计，对恶意请求的用户进行过滤操作；2) 尽量将网页设计成静态网址，将静态网址的资源存放在独立的服务器中，这样较有效防止对主服务器的攻击。3) 增强操作系统的 TCP/IP 栈，使得在一定程度上有效抵挡 SYN 攻击包。若检测到存在 CC 攻击

之后的应对措施有：1) 将域名解除绑定。采用 CC 攻击方式的攻击者会在攻击工具中将域名设置为攻击目标，因此防止对网页实施攻击的对应措施是在互联网信息服务上 (Internet Information Services, IIS) 取消该域名的绑定；2) 将域名解析到本地环回地址，即 IP 地址为 127.0.0.1，这样 CC 攻击的对象将转变为对自身的攻击。3) 一般情况下，Web 通过 80 端口提供服务，因此可采用更改 Web 端口的方式防止 CC 攻击；4) 检测到攻击者的 IP 后，在 IIS 中屏蔽该 IP，防止该 IP 再次对服务器进行攻击。另外，李硕等人针对 CC 攻击后服务器还会消耗部分资源的问题，采用了添加和真实系统相仿的蜜罐主机，但是该方法存在资源利用率低下的问题<sup>[18]</sup>。Chun-Tao X 等人根据 CC 攻击具有攻击网页随机分布的特点，提出了一种根据信息熵的原理实时检测攻击发生的方法，利用 HTTP 重定向和 Cookie 技术发现攻击源并及时阻止攻击的连接，同时设计并实现了基于 netfilter 框架的 CC 攻击检测与防御系统，该算法想法巧妙，但随机分布概率的计算较为困难<sup>[19]</sup>。景泓斐等人研究分析了 CC 攻击的特点，选取包速率、URL 信息熵以及 URL 条件熵作为检测 CC 攻击的评判指标，提出一种基于误差逆向传播 (Back Propagation, BP) 神经网络的方法建立二分类模型以此用来检测是否受到 CC 攻击，并在中、小型网站上能有效检测出正常访问的流量以及 CC 攻击产生的流量，但是该算法在 CC 攻击过程中采集异常流量的阈值需要另外根据数据的变化人工选择，这就造成最后检测的效果会由于阈值选取是否恰当而受到影响<sup>[20]</sup>。

### 1.2.3 DDoS 攻击

DDoS (Distributed Denial of Service) 网络攻击方式是利用大量的分布式服务器对攻击目标发起请求，造成服务器处理请求过程发生拥堵，从而导致无法正常供用户使用<sup>[21]</sup>。DDoS 攻击与 CC 攻击方式类似，主要区别在于 CC 攻击是面向应用层，而 DDoS 面向的是网络层，但最终的目的都是造成服务器瘫痪。

DDoS 攻击技术简单，但具有攻击速度快，伤害威力强的能力，也是黑客经常会选择的一种网络攻击方式。于是，在面对 DDoS 攻击时，研究学者在对应的防御措施技术上的研究层出不穷，提出了许多解决方法。董哲等人提出了一

种基于处理时间序列问题的隐马尔科夫模型（Hidden Markov Model, HMM）和混沌模型的检测算法，以此来解决检测算法中没考虑相邻时刻特征之间关联的问题，但 HMM 模型的计算量较大，导致算法相对复杂<sup>[22]</sup>。程杰仁等人定义了一个用来表征网络数据流特征的 IP 数据包统计特征（IPDCF），同时也采用了一种用来处理时间序列问题的长短期记忆模型（Long Short-Term Memory, LSTM）对 IPDCF 特征进行建模，通过该模型实现对 DDoS 攻击引起异常的检测，但由于神经网络 LSTM 模型中的参数数量庞大，该算法对设备配置有较高要求<sup>[23]</sup>。将机器学习（Machine Learning, ML）方法应用在 DDoS 攻击检测上的研究也逐渐被广泛使用，Idhammad M 等人综合机器学习中监督训练与无监督训练各自的优势，提出了一种基于网络熵估计、协同聚类、信息增益比和 Extra-Trees 算法的半监督机器学习方法，用该算法实现对 DDoS 攻击检测<sup>[24]</sup>。另外，Chen Y 等人开发了一种基于机器学习方法中的决策树模型的 DDoS 攻击检测系统，该系统具有检测高效、轻量的优点，实验结果表明，该系统可以有效地保护网络免受各种 DDoS 攻击<sup>[25]</sup>，但是基于机器学习的方法需要额外人工提取特征，并最终的检测效果极大程度受特征选择的影响。

### 1.3 研究内容和结构安排

本文针对手持式燃气抄表系统，研究数据边界安全监控方法，提出总体解决方案，根据解决方案设计基于嵌入式系统的数据边界安全监控系统。重点研究深度包过滤技术和数据加解密技术，通过仿真实验分析算法性能，采用改进的 AC\_HA 算法和 AES 加密算法。在此基础上将设计的算法移植到嵌入式平台，实现燃气数据边界安全控制功能。本文共分为六章，主要内容和结构安排如下：

第一章，绪论。主要介绍了论文的研究背景、意义和相关的国内外研究关于网络安全问题的发展现状，以及论文的章节内容安排。

第二章，系统整体设计方案，包括燃气数据边界安全监控的整体解决方案，硬件电路的设计，软件实现流程的设计等。

第三章，深度包过滤技术的研究，包括深度包解析算法中单模式匹配和多

模式匹配算法的研究，对现有的一些模式匹配算法作对比分析，提出一种改进算法——AC\_HA 算法，并与传统的算法做对比分析。

第四章，加解密技术的研究，对比分析了 AES,DES,RAS 等算法的特点以及优劣，选用了综合性能更好的 AES 算法并研究其算法实现原理，通过软件编程实现了 AES 算法。

第五章，系统实验测试，包括实验所用到的硬件电路的制作调试，网络通信的调试与测试，加解密算法和深度包解析算法在系统中的功能和性能测试，最后对测试结果进行总结分析。

第六章，总结与展望。总结了本文的研究成果，针对本论文的不足展望了改进的方向。

## 第二章 系统方案设计

本文针对手持式抄表系统，设计数据边界安全监方案。在手持式抄表过程中，抄表员到指定小区住户附近（不入户）通过无线通信进行近距离燃气表数据抄读，批量抄读完成后上传相关的册本文件。如果因为特殊原因无法在户外获取数据，并且入户时发现户主不在家时，抄表员需要留下信息，拍照保存证据并将拍摄的图像上传到远端服务器。所以手持式抄表过程中需要上传到服务器的数据有册本文件（DB 文件）和图像文件（JPG 文件）两种。为了保证上传到服务器数据的安全性，本文设计了数据边界安全监控解决方案。

### 2.1 安全监控方案设计

#### 2.1.1 安全监控方案

本设计方案在服务器前端部署一套数据边界安全监控装置，对由外网上传到服务器的数据进行监控，即在数据包上传到服务器之前进行分析过滤，只将符合协议的合法数据转发到服务器，将不合法的数据包筛除，确保上传给服务器的数据包是符合要求的安全数据包。数据边界安全监控系统如图 2-1 所示。为此本文还将设计开发数据边界安全监控装置及其软件。

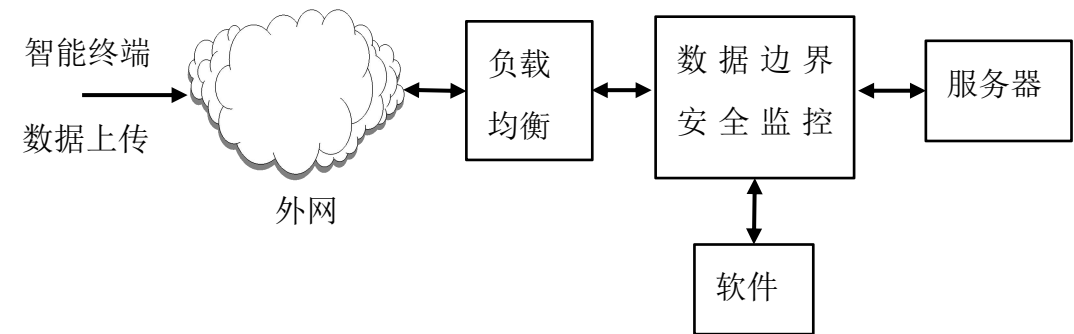


图 2-1 数据边界安全监控系统图

由图 2-1 可以看出，本设计方案是一个数据上传过程的整体设计，涉及智

能终端（手机 APP）、数据边界安全监控装置以及服务器。系统的安全控制方案如下：

- 1、智能终端在上传册本文件和图像文件之前进行预处理，即对文件名加密，并在文件内容中添加特征信息；
- 2、对上述文件内容进行整体加密后发送到远端数据安全监控装置；
- 3、数据安全监控装置对收到的数据包进行多重过滤；
- 4、将通过过滤的合法数据包转发至服务器。

### 2.1.2 智能终端数据预处理

在数据发送前，需要在智能终端（手机 APP）对册本文件和图像文件进行预处理。预处理包括以下几个步骤：

（1）对文件名称用统一格式命名，册本文件为 1XXXXXX.DB，占 10 个字节。图像文件为 XXXXXXXX（户号）+XXXXXXXXXXXXXX（时间戳）.JPG，占 27 个字节；

（2）对册本文件名和图像文件名进行加密。文件名加密过程为，先由软件产生一个随机数，将随机数取反得到 X，X 与原文件名字符逐个进行异或操作，得到加密的文件名字符串作为发送的文件名（10 字节或 27 字节）。

（3）对文件内容进行加密或由插入加密算法生成的特征信息。具体方法如下：对于册本文件，用加密算法对内容进行加密；对于图像文件，将由文件名通过加密算法产生的密文插入到图像文件内容的前面（32 字节）。

两种文件经过以上预处理后再发送，实际发送的数据包内容如图 2-2 所示。

长度	标识	文件名	随机数	内容（加密）
4B	1B	10B	1B	可变

(a) 册本文件数据包

长度	标识	文件名	随机数	文件内容（加特征数据）
4B	1B	27B	1B	32B（加密后的密文） 可变

(b) 图像文件数据包

图 2-2 数据包预处理

数据边界安全监控装置具有两个网口，一个网口与外网相连，作为服务器

接收由手持终端（客户端）发来的数据包；另一个网口作为客户端与内网的协议解析服务器相连。数据边界安全监控装置在收到客户端发来的数据包后，按照数据包过滤方案对数据包进行过滤，将通过过滤的合法数据包转发到服务器，而将不合法的数据包丢弃。

2.1.3 数据包过滤方案

本文设计数据边界安全监控系统的目的在于保证燃气自动抄表系统中服务器的安全，其核心是数据边界安全监控装置中的数据包过滤。为提高数据包过滤的可靠性，本文设计的数据过滤方法分为三个层次，如图 2-3 所示。

第三层过滤	内容过滤
第二层过滤	文件名过滤
第一层过滤	IP地址过滤

图 2-3 数据包过滤层次

- （1）第一层次：对接收的数据包的 IP 源地址进行识别，如果在短时间内（可设定）多次收到来自相同 IP 源地址的数据包（如 3 次以上），则判定为非法数据包，将其丢弃。
- （2）第二层次：文件名过滤。第一层过滤通过后，对接收的数据包中的文件名进行分析，以判断文件名的合法性。分析过程按照文件名的加密方法进行反向处理，得到实际的文件名，再根据文件名的命名规则、长度等特征判断其合法性，如果文件名不符合规范，则将此数据包丢弃。
- （3）第三层次：内容过滤。这一层是在以上两层过滤的基础上，对文件的内容进行分析和识别。识别过程如下：对于文本文件，先对文件内容进行解密，再在文件内容中查找特征字。由于在文本文件中含有特定的字段名，可设定若干个字符串作为特征字，用设计的匹配算法在文件内容查找特征字，如在文件内容中含有所有的特征字，则认定文件内容合法，否则为不合法。对于图像文件，对插入在图像文件前面的数据进行解密，再对解密后的数据进行判断，以确定

文件内容的合法性。

通过以上三个层次的过滤，可在很大程度上提高数据包过滤的准确性。为了使数据边界安全监控装置对数据包上传延时的影响减少到最低程度，需精心设计硬件电路和数据包过滤算法。

## 2.2 硬件设计

根据上述系统整体设计方案以及过滤功能，设计基于嵌入式系统的数据边界安全监控装置硬件电路。

### 2.2.1 硬件整体结构

硬件设计的主要内容是设计数据安全监控装置，其主要分为四个部分：微处理器（MCU），网络通信接口，存储器和以及串口通信和电源模块等，总体结构如图 2-4 所示。

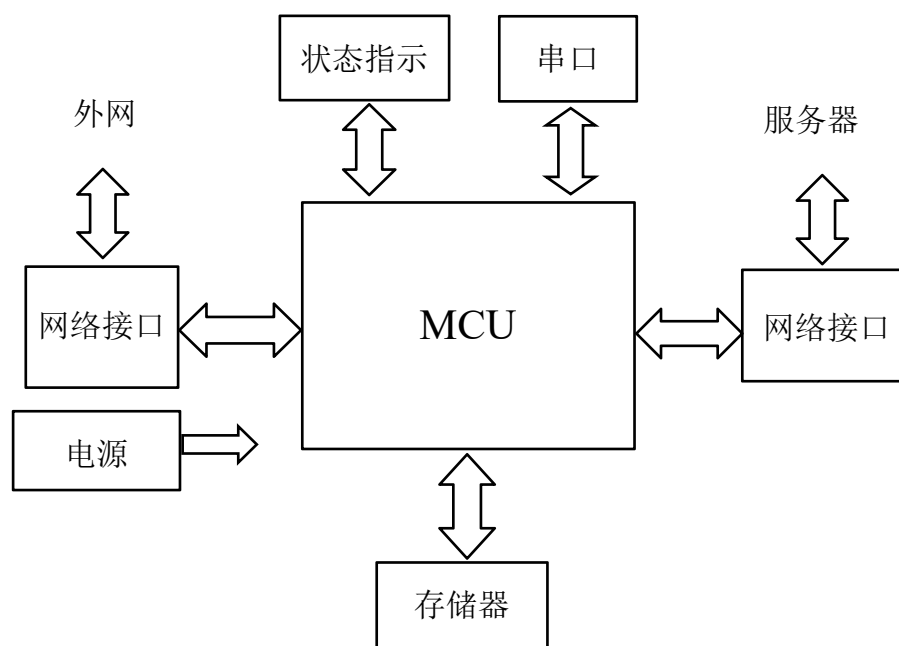


图 2-4 系统整体结构

微处理器是系统的核心，用于对整个工作过程进行控制，包括接收外网数据包、对数据包进行过滤以及将合法的数据包转发给服务器；网络接口有两部



分，一部分负责与外网连接，接收远程客户端的数据包；另一部分与服务器连接，将过滤后的合法数据包转发给服务器，另外还将统计结果通过网络上报给日志服务器；存储器用于保存接收的数据包，存储中间结果、IP 地址以及统计结果等；串口用于与上位机通信，采用 RS232 接口，以便上位机对设备进行参数配置，还可将运行状态信息发送到上位机，以便进行软硬件调试；电源模块为上述电路提供所需的电源。

### 2.2.2 控制器电路设计

控制器的性能直接影响数据边界安全监控装置的过滤效率，综合比较多种嵌入式控制器的性能、资源和成本等因素，选择 STM32H743 作为控制器。STM32H743 使用的是 Cortex M7 内核。Cortex M7 处理器是一种 32 位的高性能嵌入式处理器，拥有中断快速处理、高速处理的特点，构建于高性能处理核心，提供包括一系列和 SIMD 多重计算等功能在内的高端处理硬件功能。STM32H743 工作频率最高可以达到 400M，芯片还拥有存储保护单元并支持全套 DSP 指令，对运行时硬件的整体安全性有了很大的提升。芯片拥有 2MB 的闪存存储器和 1MB 的 RAM，对于本系统所用的文本文件和图像文件能直接存入到 RAM 中，实现高速存取。另外还具有丰富的接口和很强的扩展功能，是较为常用的高性能微处理器。本文在考虑电路实现的难易程度、处理速度、性价比等因素的基础上，选用该芯片作为主控制芯片，实现数据边界安全监控功能。

在设计控制器电路之前，首先对控制资源的使用作一个规划。根据数据边界安全监控装置的系统结构，控制器需要连接的外围电路包括两路网络接口电路、一个外存储器电路、一个 RS232 接口电路以及状态指示灯等。网络接口电路采用两个 SPI 串行接口连接；存储器采用 SDRAM 进行扩展，用 FMC 总线与控制器连接；RS232 接口利用控制器内置的 USART 接口加电平转换电路实现。以上接口所用的 IO 引脚需要软件对 STM32H743 的输入输出引脚（GPIO）功能进行配置，对控制器电路的设计主要包括控制器与上述外围电路的连接、

电源引脚的连接、振荡器电路、复位电路、调试接口以及启动模式配置等。

● GPIO 引脚配置

根据 STM32H743 控制器的资源和外围芯片的特性，对它的输入输出引脚功能的配置及连接关系见表 2-1。

表 2-1 控制器 GPIO 引脚分配

序号	引脚号	GPIO	功能定义	连接关系
1	50-53	PA4-PA7	SPI 接口 1	第一个 W5500 的 CS、SCK、MISO 和 MOSI
2	120-121	PA9-PA10	USART 串口 1	串口 TX、RX 脚
3	34	PC2	FMC_SDNEO	接 SDRAM 的 CS 脚
4	35	PC3	FMC_SDCKE0	接 SDRAM 的 CKE 脚
5	142-143	PD0-PD1	FMC_D2-FMC_D3	FMC 总线数据线 D2-D3 (SDRAM 用)
6	147	PD5	FMC_NWE	FMC 总线 NWE (WR) (SDRAM 用)
7	96-98	PD8-PD10	FMC_D13-D15	FMC 总线数据线 D13-D15 (SDRAM 用)
8	104-105	PD14-PD15	FMC_D0-D1	FMC 总线数据线 D0-D1 (SDRAM 用)
9	169-170	PE0-PE1	FMC_NBL0-1	FMC 总线 NBL0 (SDRAM 专用)
10	170	PE1	FMC_NBL1	FMC 总线 NBL1 (SDRAM 专用)
11	68-70	PE7-9	FMC_D4-D6	FMC 总线数据线 D4-D6 (SDRAM 用)
12	73-78	PE10-15	FMC_D7-12	FMC 总线数据线 D7-D12 (SDRAM 用)
13	16-21	PF0-5	FMC_A0-A5	FMC 总线地址线 A0-A5 (SDRAM 用)
14	59	PF11	FMC_SDNRAS	SDRAM 的 RAS 脚
15	60, 63-67	PF12-15	FMC_A6-A11	FMC 总线地址线 A6-A11 (SDRAM 用)
16	106	PG2	FMC_A12	FMC 总线地址线 A12 (SDRAM 用)
17	108-109	PG4-5	FMC_BA0-BA1	接 SDRAM 的 BA0 脚
18	109	PG5	FMC_BA1	接 SDRAM 的 BA1 脚
19	160	PG15	FMC_SDNCAS	接 SDRAM 的 CAS 脚
20	131-134	PIO-PI3	SPI 接口 2	第二个 W5500 的 CS、SCK、MISO、MOSI

● 电源电路

STM32H743 所需的电源电压是 3.3V，系统供电电源是 220V 交流电，因此需经开关电源变换产生 5V 电压源，再经过降压稳压器得到 3.3V 电压。

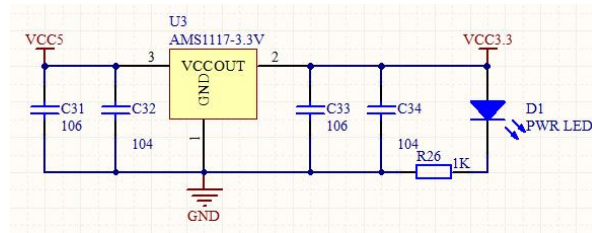


图 2-5 电源电路

本项目采用 AMS1117-3.3 集成稳压芯片，AMS1117-3.3 最大能输出 1 A 电流，输出电压精度在 0.2% 以内，能满足本项目的要求，其电路如图 2-5 所示。

将该电源的输出引脚和接地端分别与控制器的电源引脚和地相连，即可对控制器供电，电源连接如图 2-6 所示。

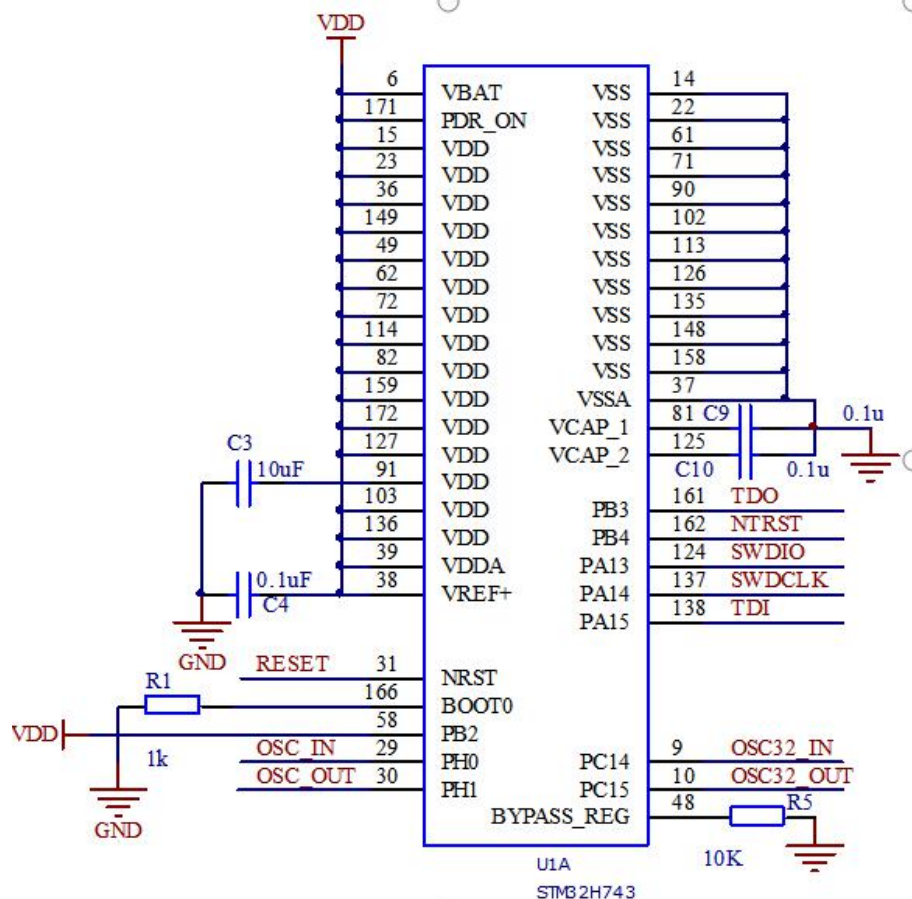


图 2-6 控制器电源连接图

#### ● 时钟电路

STM32H743 的主时钟由内部锁相环产生，外围只需接无源晶振即可。晶

振采用 25MHz 的贴片晶振，通过软件设置寄存器进行倍频即可得到 400MHz 的主频时钟。晶振电路如图 2-7 所示。

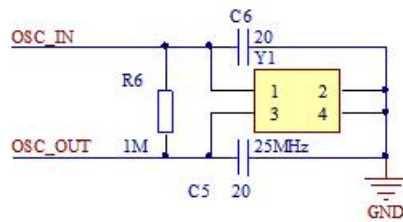


图 2-7 晶振电路

### ● 其它电路

为使控制器正常工作，还需上电复位电路，该复位电路可用简单的电阻和电容实现，如图 2-8 所示。另外，为了对 STM32H743 代码下载和进行调试，还需要设计 JTAG 接口。JTAG 接口采用 20 脚的插座，其信号引脚分别与控制器的对应引脚相连，如图 2-9 所示。

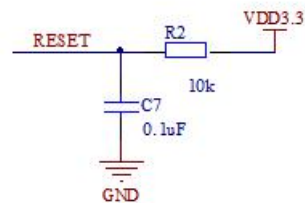


图 2-8 复位电路

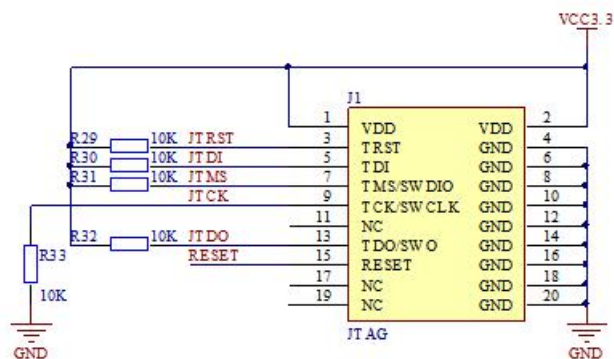


图 2-9 JTAG 接口

## 2.2.3 网络接口设计

本文的数据边界安全监控选用成熟稳定且安全性高的有线网络，使用成本

较低的双绞线作为传输介质，使用以太网为传输形式。以太网通信规定：发送时收发器对 MAC 帧数据进行编码之后再发送；接收数据时收发器需要 MAC 解码，再进行接收。物理层收发器（PHY 芯片）与 MAC 芯片的配合使用，才能完成系统的网络通信功能。

根据数据边界安全监控功能，要求系统可以独立作为客户端和服务端。作为服务器能接收手机客户端上传的数据包，进行数据分析过滤之后，再作为客户端将符合要求的数据包上传到服务器，因此需要两个独立的网络接口。本文选用两个集成 PHY+MAC 的 W5500 芯片作为网络通信接口。

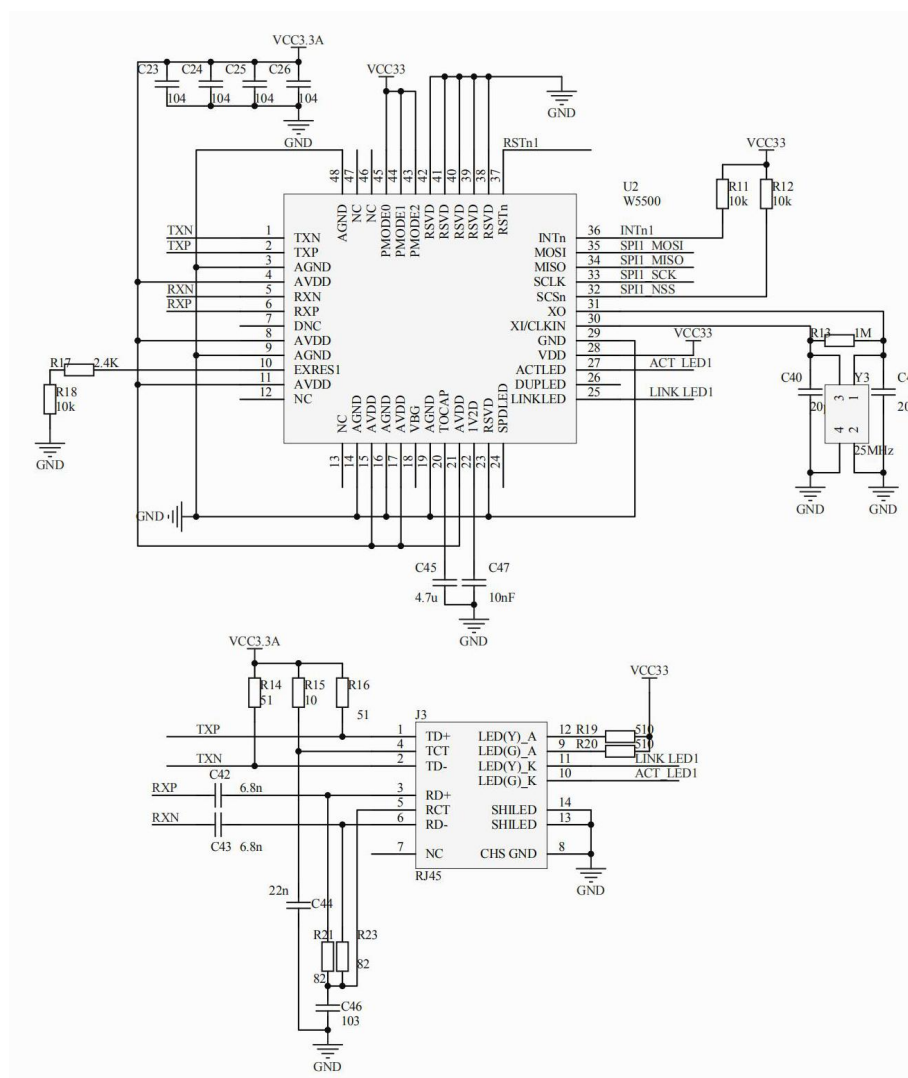


图 2-10 W5500 网络接口电路

W5500 芯片是全硬件的 TCP/IP 协议栈，支持 100Mbps 的以太网速率。其

特点在于独立于 MCU，减少通信时 MCU 的负荷，在一定程度上比软件 TCP/IP 协议栈更加安全和高效。该芯片可以通过 SPI 与控制器连接，芯片的 SPI 支持 80Mhz 的传输速度和高效的 SPI 协议，能够使网络通信更加流畅高效，以保证数据传输的效率。除此之外，该芯片还具有 8 个 Socket 端口和 32K 字节收发缓存器，控制和存取较为方便。设计的 W5500 电路如图 2-10 所示（其中一个 W5500）。

2.2.4 存储器电路设计

存储器采用 SDRAM 进行扩展，用于存储从输入输出网口接收的数据包，为算法提供所需的存储器空间，以及网络状态信息和日志数据等，其优点在于读写速度快，存储空间大，这些优点使得其常作为高速存储器使用。

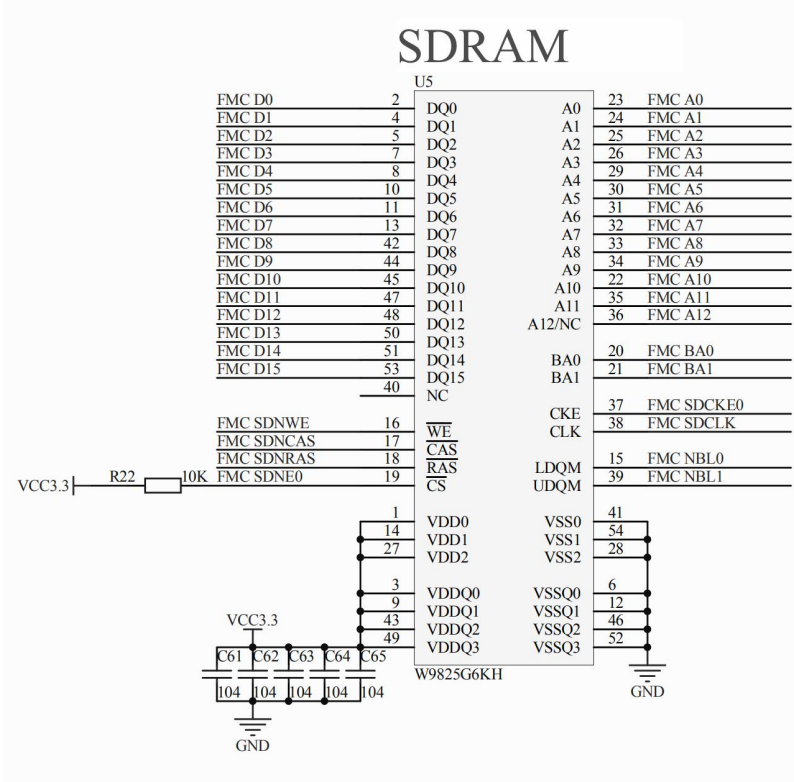


图 2-11 SDRAM 芯片原理图

由于本系统通信速率较高，通信的数据量较大，本系统选用 W9825G6KH 作为系统的外部扩展存储器，该芯片数据基带传输速率可达 166M 字每秒，数据刷新速率为 64ms 每次。访问模式设置为突发触发模式，即在同一行中对相

邻的存储单元同时进行数据传输的方式，同时传输到的存储单元数量就是突发长度，突发长度可以是 1、2、4、8 或整页。其 32M 字节容量为控制器收发数据包和对数据处理提供足够的运行存储空间，保障系统的高效运行。存储器电路如图 2-11 所示。

### 2.2.5 串口通信接口设计

STM32H743 内置 4 个通用同步/异步串行接收/发送器（USART），本系统选用其中的一个 USART 实现作为 RS232 数据收发器，实现与上位机通信。由于 USART 的信号电平为 0~3.3V，为了产生 RS232 信号电平（ $\pm 3V \sim \pm 15V$ ），需要对电平进行转换。因此选用 SP3232E 芯片作电平转换器，将 TTL 电路转换为 RS-232 的标准信号。电路设计时，将 USART 的发送和接收引脚（USART\_TX、USART\_RX）连接电平转换 TTL 电平一侧，经电平转换后再接到串口插座。在 RS232 通信接口中，只需配置 RX, TX, GND 三个引脚即可。其电路如图 2-12 所示。

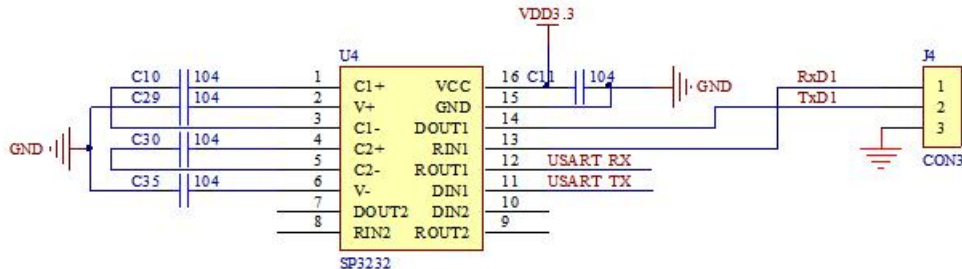


图 2-12 SP3232 与 AMS1117 芯片原理图

## 2.3 软件流程

在硬件基础上，需要设计相应的软件才能实现数据边界安全监控。软件的主要功能是：一方面通过控制网络通信芯片 W5500 实现与客户端和服务器的数据包收发，另一方面对客户端发来的数据包进行过滤；另外还要对控制器中的寄存器进行读写操作，实现对串口、存储器、指示灯和定时器等功能模块的控制。软件功能组成如图 2-13 所示。

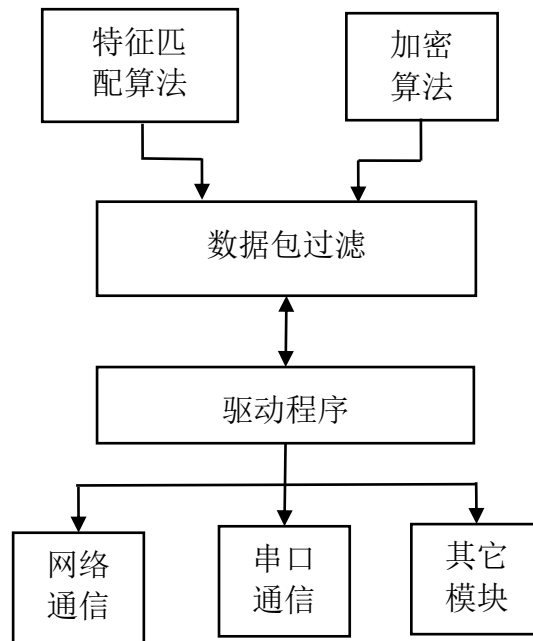


图 2-13 软件功能组成

根据数据边界安全监控方案，数据包过滤的步骤如下：

- （1）接收数据包并识别该数据包的 IP 源地址，若在短时间内（如一分钟）收到同一个 IP 地址发送的数据包多次（三次以上），则丢弃该数据包；
- （2）对文件名进行解密，判断解密后的文件中是否含有由特定算法生成的信息，将不符合条件的数据包抛弃；
- （3）识别 DB 文件中是否含有特征数据或特殊的字段名（例如数据库中特定字段名）；
- （4）识别 JPG 图像文件中的内容是否含有由文件名经过加密生成的特征数据，将不满足条件的数据包丢弃；
- （5）将上述步骤中符合所有条件的数据包上传给服务器；
- （6）如服务器有响应数据包，则将响应数据包回传给源 IP 地址（客户端）。

软件运行于 STM32H743 控制器平台，其流程如图 2-14 所示。数据包过滤技术的核心是特征数据匹配算法和数据加解算法的设计和实现，因此下文将重点研究深度包过滤技术和加密算法。



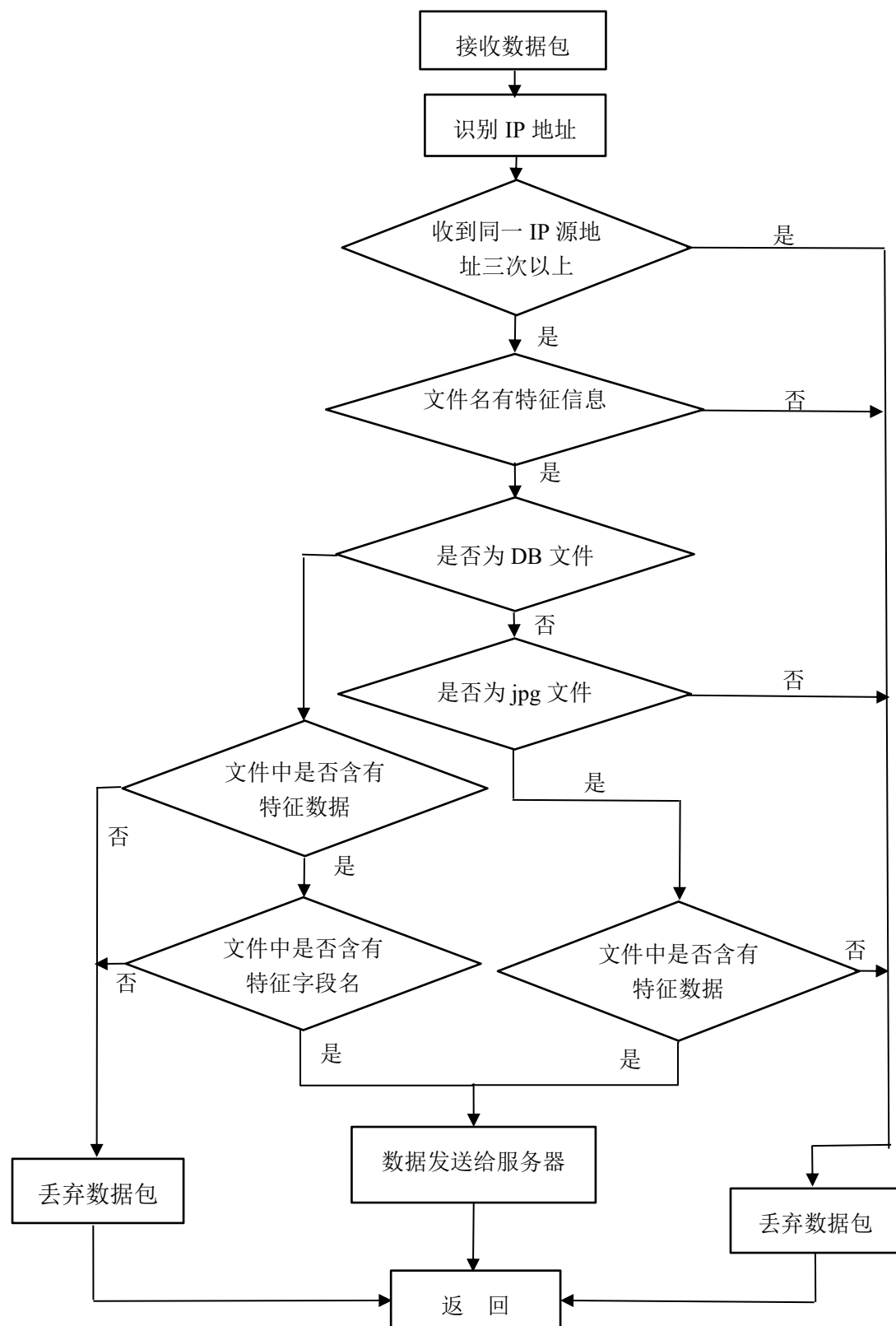


图 2-14 数据包过滤流程

## 2.4 本章小结

本章主要介绍了智慧燃气数据边界安全监控系统整体实现方案，通过在服务器前端设置数据包过滤达到安全监控的目的。数据包过滤采用三层过滤方案，使数据包过滤更准确、更有效。硬件设计中采用基于 STM32H743 的嵌入式系统，并采用具有硬件 TCP/IP 协议栈的专用网络接口芯片作为客户端和服务端进行数据包的收发，使系统可靠性好，性价比高。根据系统功能说明了软件的组成，同时根据数据边界安全监控方案给出了数据包过滤的流程。

## 第三章 深度包过滤技术研究

### 3.1 引言

在数据边界安全监控方案中，使用了特殊字段的匹配来作为其中的一层内容过滤。这一过程主要用到深度包过滤技术。传统的包过滤技术只是对数据包的包头部分（源 IP 地址、源端口、目的 IP 地址、目的端口等）进行分析来决定数据包是否能通过防火墙，而对隐藏在数据包内部的潜在危险却无能为力。相比于传统包过滤解析，深度包过滤技术不仅能解析 IP 数据包的四层以下内容，还可以对捕获的 IP 数据包进行基于应用协议的解析，详细的了解到数据包内部信息的含义，执行基于应用层的分析与控制。该技术具有强大的数据解析能力，能够根据流量分析结果来制定相应的策略。深度包过滤技术的使用场景主要有流量的识别监控、网络行为日志管理、作为防火墙抵御网络攻击、用户行为分析等等。该技术的核心是识别模块，利用匹配算法完成数据和特征信息的比较，该技术应用到系统上的性能优劣是由匹配算法的效率决定的。

匹配算法主要分为两大类：单模式匹配和多模式匹配。模式匹配是基于文本串的一种串运算。定义一个文本串为  $T_1 \dots T_n$ ，定义一个模式串为  $P_1 \dots P_m$ ，在文本串中找到某个字串  $T_x \dots T_{x+m-1} = P_1 \dots P_m$ ，如图 3-1 所示。若模式匹配成功并返回文本串中  $x$  的位置（即对应模式串第一个字符的文本串位置），若文本串中没有子串与模式串相匹配，则匹配失败。

$$\begin{array}{c}
 T_1 \cdots T_x \cdots T_{x+m-1} \cdots T_n \\
 \vdots \quad \quad \quad \vdots \\
 P_1 \cdots P_m
 \end{array}$$

图 3-1 文本串模式串对比图

最早的模式匹配算法有 KMP 算法、BM 算法等，随着算法的不断改进和更新，衍生出了 BMH，BMHS 以及多模式匹配算法中的 AC，AC\_BM 算法等各类算法，旨在应对不同场景下的需求尽可能提升模式匹配的效率，降低算法的

时间复杂度。模式匹配算法不仅具有科研价值，其实用价值也非常高，应用匹配算法的场景很多，例如各类搜索引擎的关键词查找，各类文本工具的关键词定位，各类去重系统以及计算生物学等等。因此现有很多相关文献针对各类匹配算法做了不同的研究和改进，旨在优化算法效率，寻求最优算法来解决字符串匹配的问题，即在保证文本串和模式串大小不变的前提下，算法的平均效率最高。本章在分析研究现有的各类模式匹配算法的基础上，提出一种高效改进算法用于本系统的深度包过滤。

## 3.2 单模式匹配

所有的单模式匹配算法都有一个共同的特点，就是使用一个长度等于模式串长度的窗口，随着窗口的移动来进行字符串的匹配。根据窗口的移动规则按照不同的方式将单模式匹配算法分为基于前缀的匹配方法、基于后缀的匹配方法和基于子串的匹配方法三类。

### 3.2.1 基于前缀匹配的方法

最有代表性的基于前缀的方法是 KMP 算法，KMP 算法的主要思想是进行文本预处理并得出失效函数，来计算出每个字符所对应的  $next[i]$  从而通过减少子串不匹配时窗口移动的距离来降低算法的时间复杂度。 $next[i]$  数组的计算公式为公式 (3-1)。

$$next[i] = \begin{cases} 0, i=1 \\ \text{Max}\{k \mid 1 < k < i, P_1 \cdots P_{k-1} = P_{i-k+1} \cdots P_{i-1}\}, & (3-1) \\ 1, \text{others} \end{cases}$$

其中移动的距离是算法的关键，这一指标一般由失效函数来决定。KMP 算法的本质是构建有限自动状态机，其具有有向无环图的特点，可以对自动机进行很高效的改造。与传统的朴素匹配算法（暴力匹配算法）相比，KMP 算法引入了跳转表，摒弃了朴素匹配算法中每次移动一位进行比较的方式，通过找到匹配子串的相似性，来减少与母串的重复匹配。KMP 算法预处理的时间复杂度为  $O(n)$ ，其空间复杂度为  $O(n)$ 。

### 3.2.2 基于子串匹配的方法

当文本串的随机性很强时，基于子串匹配的方法所用的平均时间复杂度最低，以 BDM 算法为例，在窗口函数  $W(k)=T[j..j+m-1]$  从右向左读入文本串中的字符，从其后缀中找到模式串的最大前缀，窗口移动的距离依据该最大前缀来决定。每一个窗口内的平均最大前缀的长度为  $O(\log \sigma m)$ ，每次移动的距离为  $m * O(\log \sigma m)$ ，由此得出该算法的平均时间复杂度为  $O(n * \log \sigma m)$ 。该算法的应用场景比较少，一般在处理大数据集和小字符串时使用，且建立自动机的过程比较繁杂，所以在实际应用时使用的比较少。

### 3.2.3 基于后缀匹配的方法

基于后缀匹配的重点在于窗口的移动规则，BM 算法是一种典型的基于后缀匹配的方法，也是在实际中应用非常广泛的一种基础匹配算法。Boyer 和 Mooer 根据 KMP 算法的原理提出的更高效的算法——Boyer-Mooer 算法，该算法与 KMP 算法上有本质的不同，虽然模式串的移动方向都是从左至右，但 BM 算法的比较顺序是从模式串的尾部到首部进行比较，核心在于用窗口跳跃式的比较方式来减少匹配次数以提高效率。BM 算法的两个重要规则是坏字符和好后缀(图 3-2)，这两个规则的合理使用可以最有效的加大窗口的移动距离，移动距离的大小一般等于坏字符和好后缀规则计算出的距离中较大的那一个，并且这两个规则的后移距离只与模式串有关，与文本串无关。

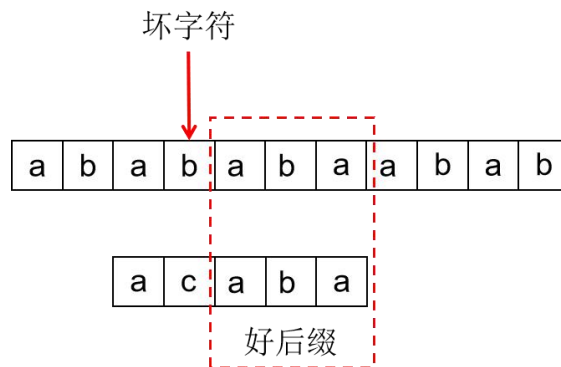


图 3-2 坏字符和好后缀

坏字符规则:若坏字符在文本串中，则将窗口移动到坏字符的下一位再进行

比较；若坏字符在模式串中，则将文本串中不匹配的字符在模式串中找到并对齐。跳转函数  $skip(x)$  的计算规则为公式 (3-2)，其中  $x$  为失配时文本串中对应的失配字符。

$$skip(x) = \begin{cases} m, x \neq P[j](1 \leq j \leq m), x \text{ 不在 } P \\ m - \max(x, \{k \mid P[k] = x, 1 \leq k \leq m\}), x \text{ 在 } P \text{ 中} \end{cases} \quad (3-2)$$

好后缀规则:发现字符不匹配时，若恰好模式串中有一个子串与好后缀相匹配，则向右移动窗口至模式串中的子串与文本串中的好后缀位置对齐；若模式串中并没有其他子串与好后缀相匹配，则从模式串中找到一个能够与好后缀相匹配的最长前缀，并将模式串移动至最长前缀与好后缀子串对齐；若模式串中不存在其他子串与好后缀及其子串相匹配，即以上两种情况都不存在时，直接将该字符串移动到好后缀的下一个字符的位置。 $shift(x)$  的计算公式为式(3-3)。

$$shift(j) = \min\{s \mid (P[j+1...m] = P[j-s+1...m-s]) \& \& (P[j] \neq P[j-s])(j > s), P[s+1...m] = P[1...m](j \leq s)\} \quad (3-3)$$

其中  $j$  为好后缀的位置， $s$  为失配时好后缀的位置与下一次出现好后缀的位置之间的距离。

通过以上规则可以看出 BM 算法的比较次数相比于 AC 算法有显著的减少，加快了字符匹配进程，但是 BM 算法会丢失每次比较之前的所得到的匹配信息，因此每次匹配时都需要重新对模式串的每个字符进行匹配。BM 算法的最坏时间复杂度是  $O(m*n)$ ，其改进算法的时间复杂度大都呈现线性趋势。

### 3.3 多模式匹配

单模式匹配算法是最为简单的一类匹配算法，但它的局限性在于一次文本的遍历只能匹配一个模式串，自然地可以想到一次性匹配一组字符串。因此产生了多模式匹配算法，即遍历一次本文可以匹配多个字符串的算法。本节主要介绍 AC 算法和其改进算法 AC\_BM 的特点及优缺点。

#### 3.3.1 AC 算法

经典的 AC 匹配算法是由 Alfred V.Aho 和 Margaret J.Corasick 提出的，在

实际应用中非常广泛，是基于有限自动状态机（FSA）的一种匹配算法。自动机的构成只与给定的模式串相关，自动机的每个节点都有 256 个指针，分别对应 256 个 ASCII 码值，但是由于模式串中的字符不可能全部包含 256 个 ASCII 码，因此会有很多的节点为空指针，这就意味着当模式串越多时，所需要的内存空间也就更大。AC 算法主要分为预处理阶段和匹配过程。

1、预处理阶段：包括自动状态机的构建和转移函数以及输出函数的计算，对于自动状态机构建的一般过程是，对于给定的模式集  $P=\{P_1, P_2, \dots, P_m\}$ ，从 0 状态开始，依次读取  $P_1$  的每一个字符，不断生成新状态， $P_1$  构建完后构建  $P_2$ ，直至  $P_m$ 。现以模式集  $P=\{so, son, is, aso\}$  为例构建自动状态机如图 3-3 所示，

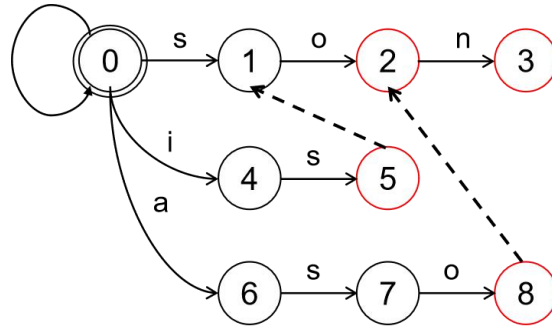


图 3-3 有限自动状态机的构造

当自动状态机构建完成以后，需要计算其转移函数，即 *goto* 函数和 *failure* 函数，函数用来指向该状态下读入某一字符的下一个状态，以图 3-3 中的状态机为例，*goto* 函数可表示为式(3-4)至式(3-11)。

$$goto(0, s) = 1 \quad (3-4)$$

$$goto(1, o) = 2 \quad (3-5)$$

$$goto(2, n) = 3 \quad (3-6)$$

$$goto(1, i) = 4 \quad (3-7)$$

$$goto(4, s) = 5 \quad (3-8)$$

$$goto(0, a) = 6 \quad (3-9)$$

$$goto(6, s) = 7 \quad (3-10)$$

$$goto(7, o) = 8 \quad (3-11)$$

若输入字符没有对应的状态时,  $goto(0,l)=fail$ 。失效函数  $failure$  函数用来表示当字符发生失配时, 转移到的下一个状态。若某状态的上一状态为 0, 则该状态的失效函数为 0。

在图 3-3 所示的自动机中转移函数表达式可以写成公式(3-12), (3-13), (3-14)。

$$failure(6) = 0 \quad (3-12)$$

$$failure(7) = goto(failure(6), s) = 1 \quad (3-13)$$

$$failure(8) = goto(failure(7), o) = 2 \quad (3-14)$$

$$output(x) = output(x) \cup output(y) \quad (3-15)$$

自动机每次构建完一个模式串, 模式串结尾的状态都要添加到输出函数中, 输出函数的计算公式为式 (3-15), 其中  $y$  状态为  $x$  的转移函数, 图 3-3 中的  $output(8)=\{aso,so\}$ 。从上述过程可以看出, 这一过程的核心思想就是  $goto$  函数,  $failure$  函数和  $output$  函数的建立。

2、匹配阶段: 由于匹配过程是从文本串首部到尾部, 因此文本串的指针  $P$  首先指向首字符, 并依次读取指针所指向的字符直到指针  $P$  的指向为 NULL, 计算该状态下字符的  $goto$  函数, 若  $goto$  函数不等于  $fail$ , 且下一状态输出函数  $output$  为空, 则指针加一继续上述步骤; 若  $goto$  函数为  $fail$ , 则计算其失调函数  $failure$ , 并转移现在的状态为失调函数得出的值再重复上述步骤。

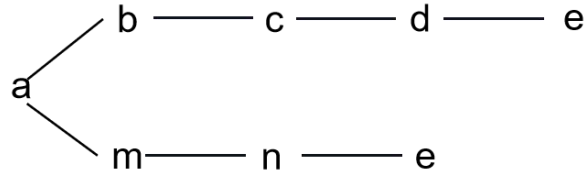
AC 算法的特点是匹配稳定, 速度快, 并且该算法的性能不受模式串的影响, 但是该算法的空间复杂度会呈指数式递增, 虽然算法性能不受模式串影响, 但是在模式串规模过大时, 所消耗的内存空间将会成为算法的劣势。

### 3.3.2 AC 算法的改进算法

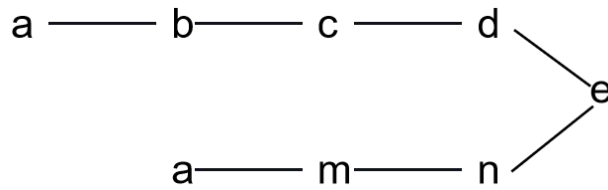
基于 AC 算法的特点, 许多改进算法应运而生, 例如 AC\_BM 算法, AC\_BM 算法<sup>[26]</sup>是 AC 算法与 BM 算法结合的一种改进算法, 它构建的是一种反向状态机。以模式串为  $\{abcd, amne\}$  为例与 AC 算法构建状态机比较如图 3-3, AC\_BM 算法加快了匹配速度, 结合了 BM 算法中坏字符好后缀的特点, 计算跳跃距离



从而减少跳跃次数，失调时调用 *shift()* 函数跳过无用的字符。该算法虽然提高了整个匹配过程的效率，但随着自动状态机越来越大，复杂度会相应的越来越大，那么构建该自动状态机的时间可能会大于字符匹配的时间。



(a) AC 算法构建自动机



(b) AC\_BM 算法构建自动机

图 3-3 AC 算法和 AC\_BM 算法构建自动机对比

另外的一些改进算法如 AC\_BMH 算法<sup>[27]</sup>跳跃的最大移动距离为最短模式串长度，AC\_QSS<sup>[28]</sup>也为最短模式串长度，也都有可能造成漏检。

### 3.4 AC\_HA 算法

综合现有的多种改进算法，本文提出一种 AC\_BM 算法结合 RK 算法<sup>[29]</sup>中 HASH 函数思想的优化算法—AC\_HA 算法。该算法在预处理阶段可以利用 hash 函数的特点首先过滤掉不需要进行匹配的字符文本。该算法的主要思想是：将模式串切割成长度为  $m$  的若干个字符串，若切割后存在重复的字符串则只保留一个存入集合  $S1$ ，然后计算集合  $S1$  中每个字符串的 hash 值，计算出来的结果存入集合  $P$ 。将目标串同样分割成长度为  $m$  的若干子串，计算其 hash 值并与集合  $P$  中的值进行比较，不匹配的值直接删除，匹配的字符组成新的集合  $S2$ ，再比较具有相同 hash 值的子串。

该算法最主要的特点是借鉴优化了 BM 算法中坏字符规则，让跳跃距离最大程度的逼近最短模式串长度。优化后的坏字符跳跃规则是：设最短模式串长度为  $L_{min}$ ，设失配时失配字符处于模式树的第  $N$  层（所处的深度），定义字符

串  $Str$  为失配时后续还未匹配的  $N$  个字符组成的字符串。

(1) 当第  $N$  层发生失配时, 判断  $N$  与  $L_{min}$  的大小关系, 若  $1 \leq N \leq L_{min}$ , 则判断  $Str$  是否存在前  $L_{min}$  层模式树中 (见 (2)); 若  $N > L_{min}$  则直接跳跃  $L_{min}$ 。

(2) 若  $Str$  在前  $L_{min}$  层模式树中, 则跳跃距离为失配字符处于模式树中的最小深度减  $N$ 。若  $Str$  不在前  $L_{min}$  层模式树中, 转到 (3)

(3) 若  $N \leq (L_{min} + 1)/2$ , 则跳跃距离为  $(L_{min} - N + 1)/2$ ; 若  $N \geq (L_{min} + 1)/2$ , 则跳跃距离为  $N$ 。

### 3.4.1 算法流程

AC\_HA 算法的流程主要分为以下两个阶段:

(1) 预处理阶段: (1) hash 值的计算, 分配好相应的内存空间, 遍历完整的模式串并进行切分放入集合  $S$ , 相应的 hash 值存入集合  $P$ ; (2) 构建 FSA, 按照 AC 算法的方式, 生成  $goto()$ ,  $skip()$  和  $output()$  函数。

(2) 匹配阶段, 用相同的方法切割目标串, 计算其子串的 hash 值与模式子串的 hash 值作比较, 保留成功匹配的子串组成集合  $T$ , 比较  $T$  中的子串长度与最短模式串的长度, 删除小于最短模式串长度的子串再进行匹配, 若发生失配, 则调用  $skip()$  函数跳跃,  $skip()$  函数与 AC 算法的  $failure()$  函数不同: 移动距离的计算规则如下: 设当前失配情况下的深度为  $N$ , 将文本串中该深度位置的字符以及该字符后面的  $N$  个字符组成字符串  $Str$  并比较该字符串与模式串中深度为  $N$  的各个子串, 若匹配则调用  $goto()$  函数到下一个状态最后由  $output()$  函数输出结果。基算法流程如图 3-4 所示。

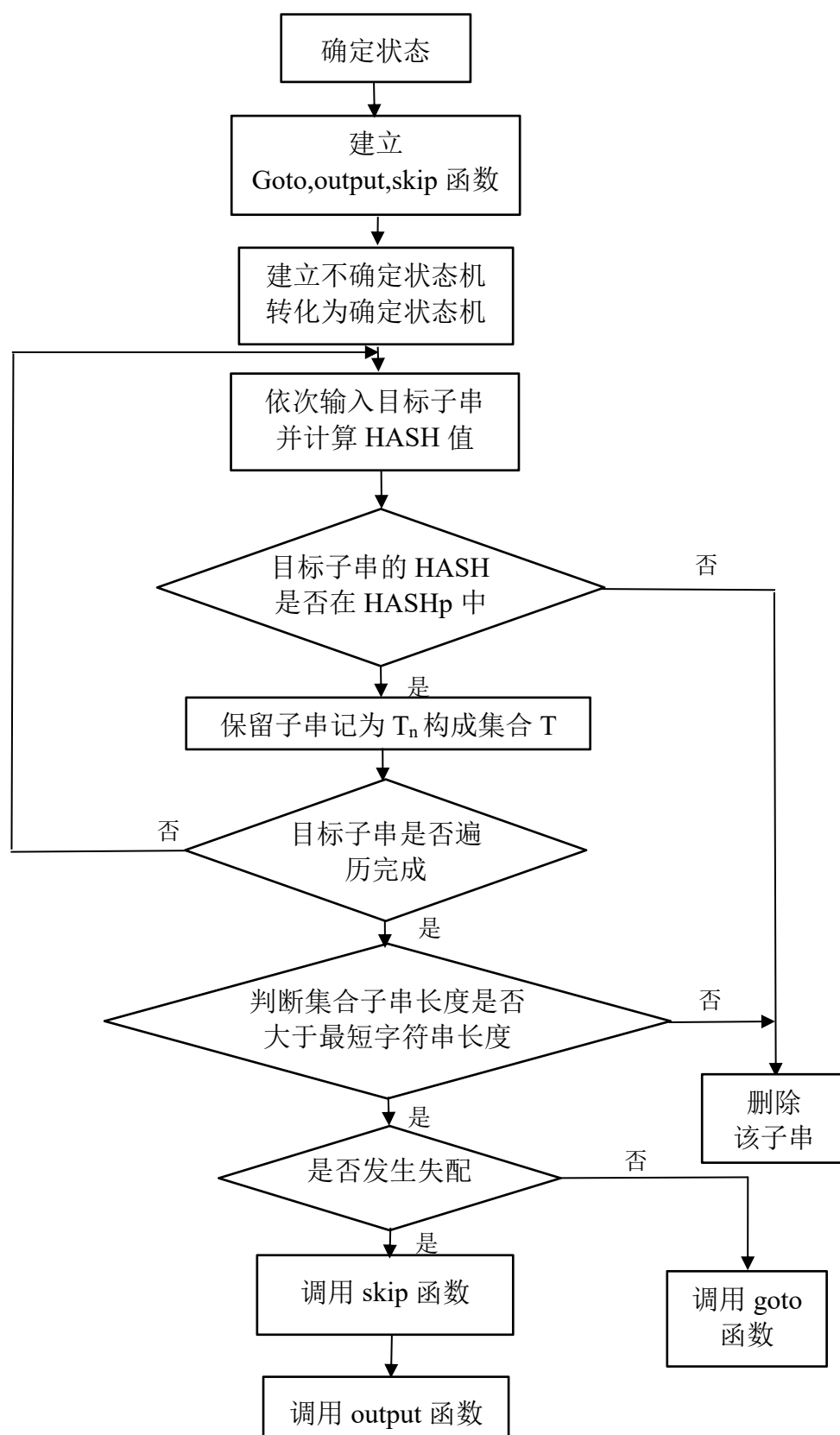


图 3-4 AC\_HA 算法流程

### 3.4.2 算法举例

以文本串“*abcgddffadfehcdecabbf*”,模式串集{“*adf*”, “*abbf*”, “*adfe*”, “*cde*”}为例, 设切割字符的长度为 1。

步骤一: 得到模式串集经过字符切割之后得到的最小集  $P'=\{a,b,c,d,e,f\}$  并申请一段大小为 256B (包含 256 个 ASCII 码值) 的内存 HASHp。通过 hash 函数的计算, 将 P' 中的字符映射到 HASHp 上, 得到如图 3-5 的地址映射。

模式串 子字符	对应 hash地址
a	E8B7BE43
b	71BEEFF9
c	6B9DF6F
d	98DD4ACC
e	EFDA7A5A
f	76D32BE0

图 3-5 模式子串 hash 地址映射图

步骤二: 构建反向有限自动状态机如图 3-6 所示, 并且生成 *goto()* 函数, *skip()* 函数和 *output()* 函数。

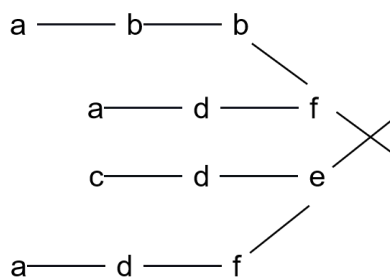


图 3-6 反向有限状态自动机

步骤三: 切割文本串为长度为 1 的字符集, 用 hash 函数计算每个字符的 hash 地址, 与步骤一中的 hash 地址相比较, 例如文本串中的字符“g”所对应的 hash 地址不存在模式串字符所映射的 hash 地址中, 所以删除该字符并将该字符之前的字符作为一个子串  $T_1=\text{“abc”}$  存储在新的集合 T 中, 以此类推。遍历完整个文本串后所得到的集合  $T=\{\text{“abc”}, \text{“ddffadfe”}, \text{“cdecabbf”}\}$ 。

步骤四: 比较集合 T 中各子串的长度与最短模式串长度的大小 (此时最短

模式串长度为 3)，删除小于最短模式串长度的子串，因为 T 中没有长度小于 3 的子串，所以不删除子串。将步骤二中构建的有限反向状态机的终节点与集合 T 中的首字符对齐，依次匹配，匹配过程以文本子串“*ddffadfe*”为例如下图 3-7 所示。

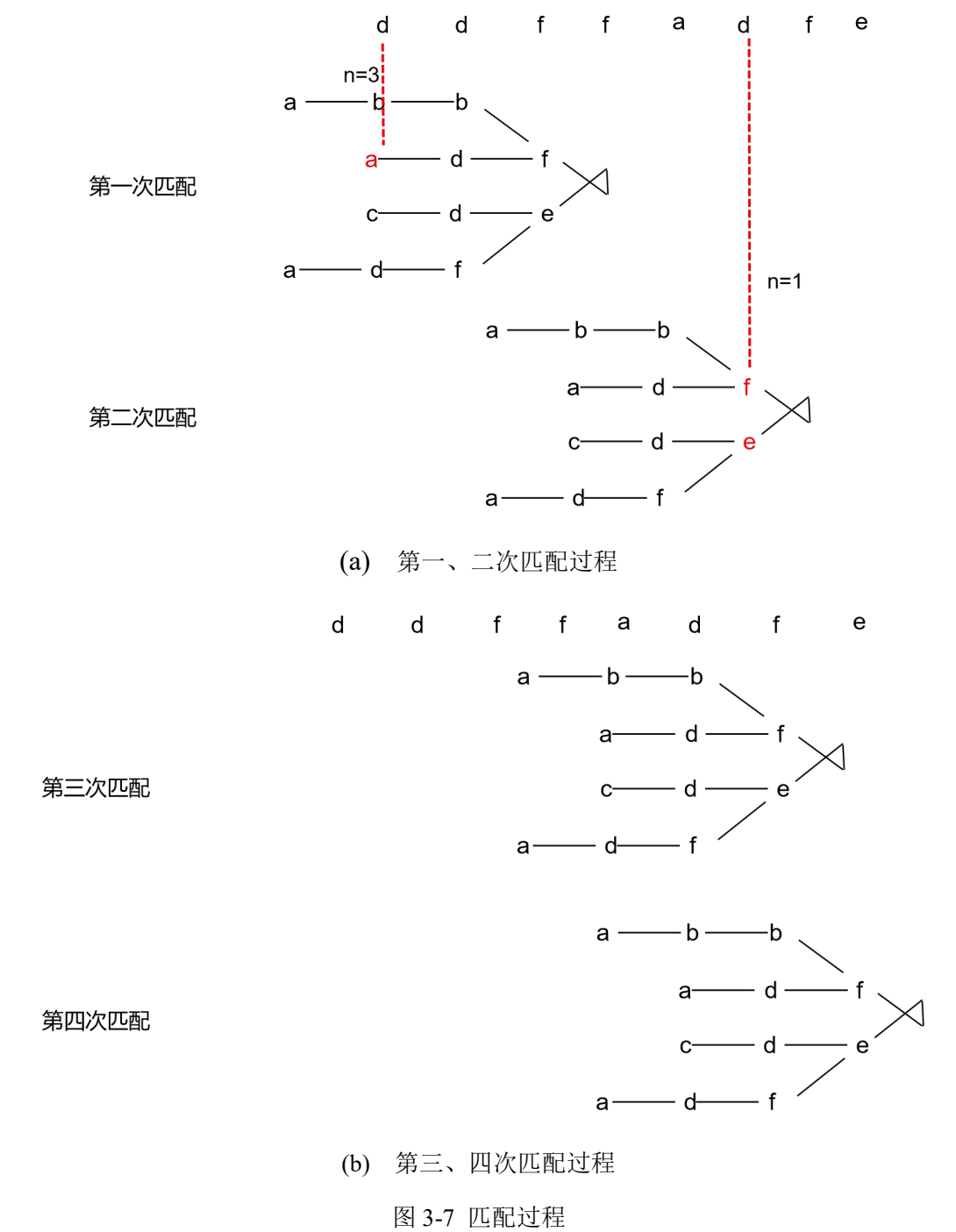


图 3-7 匹配过程

### 3.5 实验结果与分析

对前面所提到的 AC 算法,AC\_BM 算法以及改进的 AC\_HA 算法进行实验分析,对比其算法性能。本文使用的实验平台如下:计算机 CPU 配置为: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz; 操作系统为: Windows 8 系统; 软件编程语言: visual c++6。

首先对 AC\_HA 算法进行功能性测试,设置模式串大小为 4 个,文本串为“sdmfhsgnshejfgnihaofhsrnihaojslxs”,运行程序结果如图 3-8 所示。

```
STR:sdmfhsgnshejfgnihaofhsrnihaojslxs
pattern:nihao hao hs hsr

=====AC_HA-SEARCH=====
RESULT:
LOCAT  No      pattern
4       2       hs
14      0       nihao
17      1       hao
20      2       hs
20      3       hsr
23      0       nihao
26      1       hao
TOTALTIME:2/1000 (s)
Press any key to continue_
```

图 3-8 AC\_HA 算法功能测试

运行结果表明算法可以正常运行,且对模式串在文本串中出现的位置能被正确定位,算法功能正常。

#### 3.5.1 实验一: 模式串数量对算法性能的影响

由于燃气数据的文件大小不超过 1M,大多数为 100K-200K。测试模式串数量对算法性能的影响时,设置文本串大小为 124K(一篇英文文章),模式串集合写入 txt 文件中,调用不同数量的模式串测试,记录 AC 算法和 AC\_HA 算法以及 AC\_BM 算法的运行耗时。统计并绘制曲线图如图 3-9,图 3-10 所示。

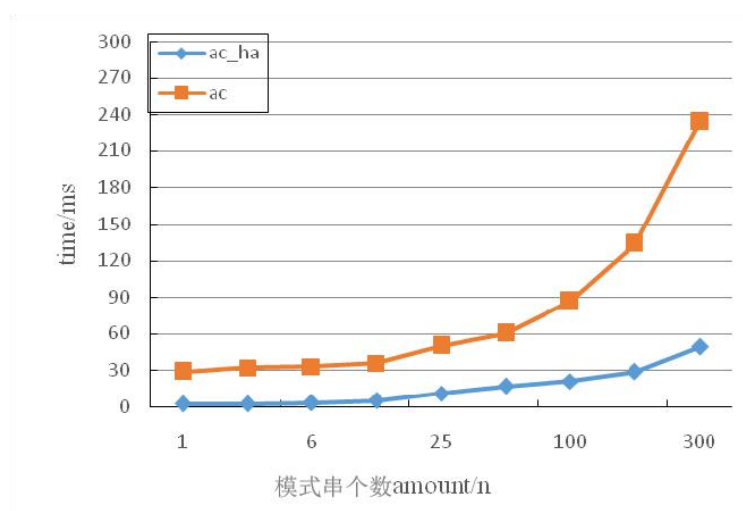


图 3-9 AC 算法与 AC\_HA 算法比较

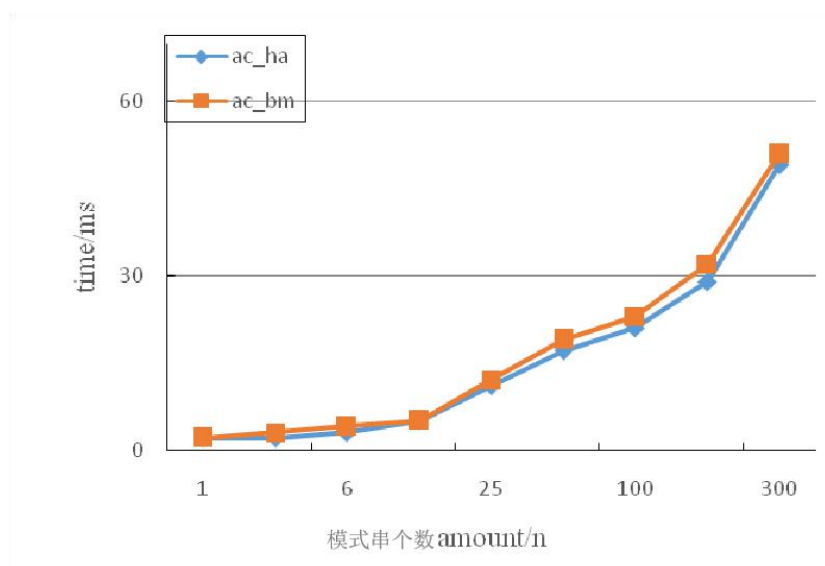


图 3-10 AC\_BM 算法与 AC\_HA 算法比较

从结果可以看出，在文本串大小和内容固定时，AC\_HA 算法的时间性能要明显优于 AC 算法，与 AC\_BM 算法相比性能上有一定提升。

### 3.5.2 实验二：文本大小对算法性能的影响

设置模式匹配字符串为 10 个，设置文本大小为 10k,20k...300k 来测试文本串大小对于算法时间性能的影响。测试结果统计如图 3-11 所示。

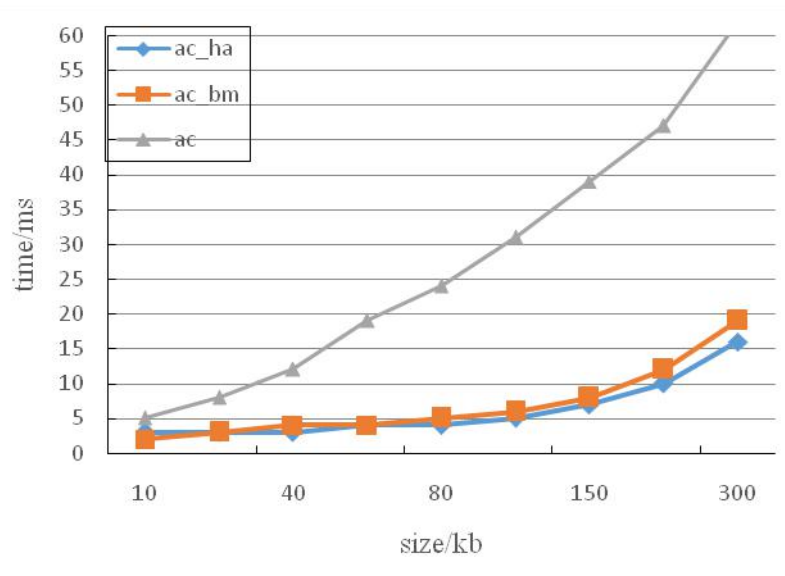


图 3-11 AC 算法，AC\_BM 算法，AC\_HA 算法的比较

从结果可以看出，在模式串大小和内容固定时，AC\_HA 算法的时间性能要明显优于 AC 算法，与 AC\_BM 算法相比在数据稍大时效果好一些。

### 3.6 本章小结

本章通过对深度包过滤技术中的匹配算法进行研究，将 AC 算法的模式串有限状态自动机的原理和 BM 算法中坏字符的规则延申到多模式匹配算法中，并算法应用到智慧燃气抄表系统上。为了燃气服务器的安全性更高，一般需要多个模式串进行匹配。在考虑算法性能时，在基于 AC 算法的前提下进行分析改进，针对 AC\_BM 算法在匹配时跳跃距离的不足，提出了 AC\_HA 算法。该算法在继承 BM 算法的坏字符规则的前提下对匹配阶段的跳跃距离进行改进，并且对文本串利用 hash 表进行缩短。该算法在仿真软件上通过实验验证可行，并且测试表时其算法时间性能优于 AC 算法及 AC\_BM 算法。



## 第四章 加解密技术研究

### 4.1 引言

加解密技术是现代各个应用中提升安全性的一种重要方式，本文的数据边界安全监控方案中也需要利用加解密技术来保证信息的传输安全以及信息验证。

加解密技术是基于密码学的一门技术，密码学的发展由古典密码学到近代密码学再到现代密码学。密码学主要分为密码编码和密码分析，就是通常所说的加密解密算法。一般的加解密算法由一套密钥进行控制。现如今是一个信息化的时代，每分每秒都在进行依赖于互联网的信息交互，但是互联网的组成繁多冗杂，难免会有漏洞，导致一系列的安全问题包括个人隐私安全，企业信息安全，乃至国家信息安全。因此越来越多的领域应用加解密技术，其中常见的用于用户身份识别、数字签名及验证，用户口令的设定及验证等等。

本章主要研究适用于对数据信息进行加密解密的算法，并将其应用到智慧燃气数据边界监控系统中。用户掌握密钥信息，上传文件名经过加密处理的数据包，由数据边界监控系统对上传的数据包文件名进行解密，比对其数据格式是否符合统一规定，由此拦截掉非法用户发送的无效数据来提升系统的安全性。目前最常见的加解密方式分为两种，对称加解密和非对称加解密，下文对比常见的加密解算法的特点及优劣，选择最符合本系统的算法。

### 4.2 对称加解密算法

对称加密算法也可以称为单密钥加密算法，加密解密可以使用同一个密钥，其解密算法为加密算法的逆过程，主要通过数字化、加密算法、加密逆算法、逆数字化完成整个加密过程。具体过程如图 4-1 所示。

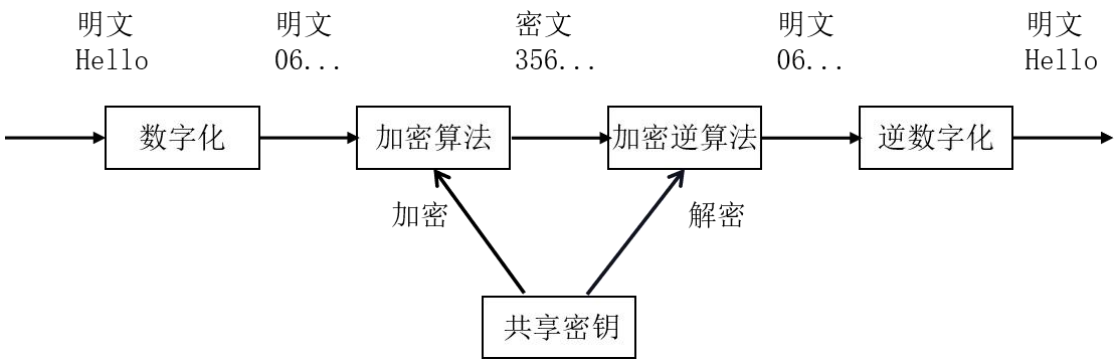


图 4-1 对称加解密过程

所以只需要知道加密算法的原理就可以反推出解密算法。密钥的长度直接决定算法的性能，长度越长越难被破解。对称加密算法运算速度快，方便简单，性能良好，应用场景广。加解密过程如公式(4-1)所示。

$$D_K(E_K(M)) = M \tag{4-1}$$

式中  $M$  代表明文， $K$  代表密钥， $E, D$  分别代表加解密过程。

4.2.1DES 算法

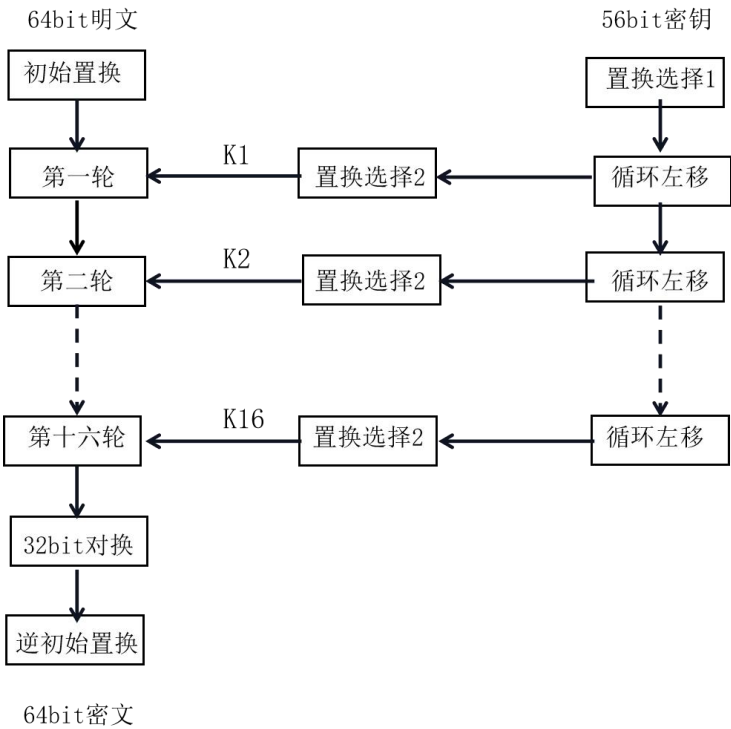


图 4-2 加密流程图解

DES 算法是一种公认的对称加密算法，它的前身是 IBM 公司提出的 Lucifer

算法，在 20 世纪 70 年代被美国联邦政府授权为标准的加密算法。DES 算法的输入明文与输出密文长度相等，都是由 8 位奇偶校验位（第 8，16，24，32，40，48，56，64 位）和 56 位密钥组成。其加密流程图如图 4-2 所示。

算法步骤如下：

（1）将输入的明文初始 IP 置换后重新排列，输出的前 32 位记为  $L_0$ ，后 32 位记为  $R_0$ 。

（2）56 位的密钥经过置换、循环移位、再置换等操作得到  $K_1-K_{16}$  个子密钥。

（3）进行 16 轮迭代运算得到  $L_1, R_1$ 。

（4）将  $L_1, R_1$  进行逆初始置换得到密文的表达式为  $IP^{-1}L_1R_1$

虽然 DES 算法原理简单，易于操作，但是其缺点是安全性不高，易于被攻破。一台计算机用穷举法对其进行算法破译，其明文，密文，密钥的互补特性会使得原本的工作量减少一半，其次算法产生的内部密钥的循环左移次数具有对称性，攻击者能利用穷举法攻击中的并行原理进一步加快搜索速度。

#### 4.2.2 AES 算法

AES 是一种分组密码，是美国国家标准技术研究所 NIST 旨在取代 DES 的 21 世纪的加密标准，现已被广泛使用。虽然 DES 的大量生产不会让其暂停使用，但是被 AES 完全取代是必然趋势。

表 4-1 分组长度  $N_b$ 、密钥长度  $N_k$  和迭代轮数  $N_r$  之间的关系

$N_r$	$N_b=4$
$N_k=128$	10
$N_k=192$	12
$N_k=256$	14

AES 算法采用明文密文都是 128 位的固定分组长度，支持 128，192，256 位的密钥，分为 AES-128, AES-192, AES-256 三种形式<sup>[30]</sup>，对应的分组长度，密

钥长度与迭代轮数的关系如表 4-1<sup>[31]</sup>。因此 AES 算法比较灵活，可以适用于更多软硬件。该算法可用  $4 \times 4$  的矩阵表示，共 16 个字节，每个字节 8 比特，如图 4-3 所示。

$$\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{bmatrix}$$

图 4-3  $4 \times 4$  状态矩阵形式

加密（解密）算法的步骤说明（以密钥长度为 128 位，迭代 10 次为例）：

（1）将  $N_b$  个字的初始密钥扩展为  $(N_b(N_r+1))$  个字的子密钥，使得每一轮迭代的子密钥都不同。

（2）将明文（密文）按图 4-3 所示用  $4 \times 4$  的矩阵表示，记为  $W$ ，将对应的轮密钥与  $W$  矩阵做异或运算，从而得到一个新矩阵  $W'$ 。

（3）第 1-9 轮的迭代运算中，每轮会使用字节替代变换，行位移变换，列混淆变换和轮密钥加变换依次对新矩阵  $W'$  操作<sup>[32]</sup>。

（4）第 10 轮运算时，省去步骤三中的列混淆变换对矩阵进行操作。最后的输出方式按照先列后行的顺序生成密文（明文）。

### 4.3 非对称加解密算法

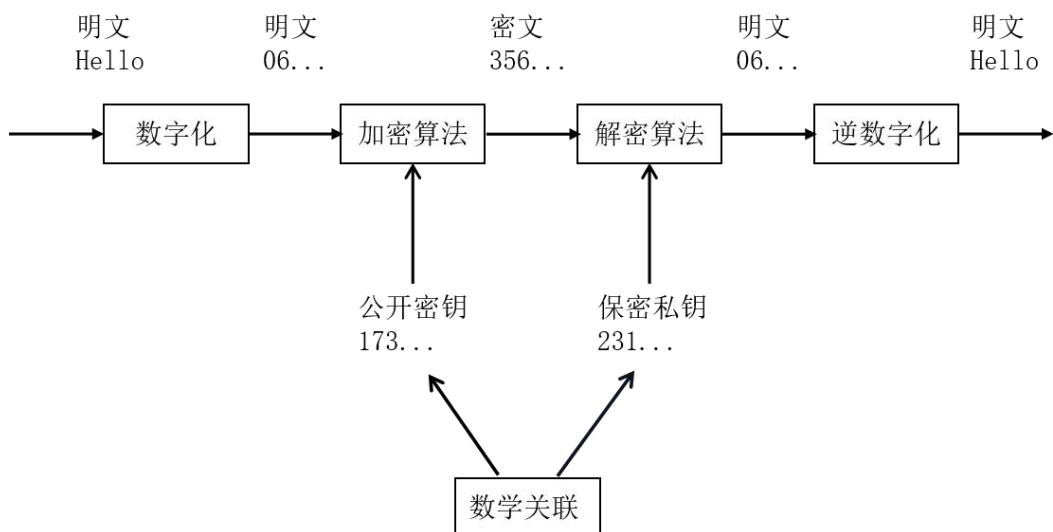


图 4-4 非对称加解密过程

非对称加密算法不同于对称加密算法的是其拥有两个密钥——公钥和私钥，公钥私钥成对生成，公钥加密则私钥解密，反之私钥加密则公钥解密。过程如图 4-4 所示。

非对称加密算法中私钥不能由公钥推导出来，公钥私钥之间有数学关联，由内部人员保管。算法过程不用保密，该算法的安全性在于密钥的保密程度，一旦泄露丢失，就可以被很轻易的破解。加解密过程用公式表示为公式(4-2)。

$$D_{k_s}(E_{k_p}(M)) = M \quad (4-2)$$

式中  $M$  代表明文， $K_s$  代表公钥， $K_p$  代表密钥， $D, E$  分别代表加解密过程。

#### 4.3.1 RAS 算法

RAS 算法是 1978 年由三位发明者 Ronald Rivest、Adi Shamir 和 Leonard Adleman 一起发明的，是一种可以同时适用于数据加密和数字签名的算法，被 ISO 推荐为公钥数据加密标准，可以抵制绝大部分密码的攻击，成为目前影响力最大的公钥加密算法。

RAS 算法的原理来源于大数分解，即两个大素数相乘，再对相乘得到的积进行因式分解就尤为困难。所以乘积可以公开作为密钥。RAS 算法属于典型的非对称加密算法，算法原理也决定算法的公开密钥密码体制：加解密密钥不同且无法在知道加密密钥的前提下推导计算出解密密钥。因此加密密钥和加解密的算法流程可以被公开，虽然解密密钥是由加密密钥得来的，但其中的数学关系只有掌握解密密钥的人知晓。但是在密钥非常短的前提下有可能被强力破解，因此 RAS 密钥长度至少为 500，以此来确保其安全性。加解密流程如图 4-5 所示。

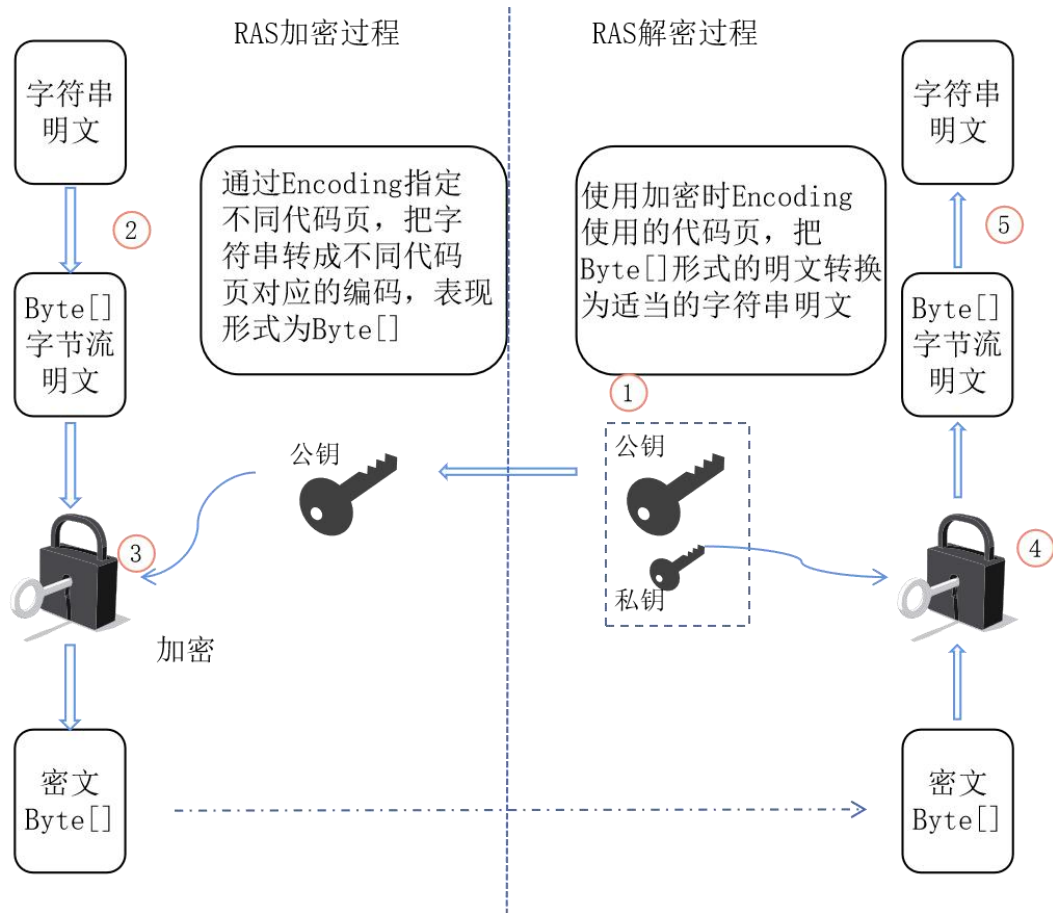


图 4-5 RAS 算法加解密图解

加密流程的原理步骤说明:

(1) 取两个素数  $p, q$  取  $n, t$  使得

$$n = p \times q \quad (4-3)$$

$$t = (p-1)(q-1) \quad (4-4)$$

取一个数  $e$  使得  $d \times e \% t = 1$  (在 c 语言中,  $d$  可以由  $\text{pow}(e, -1) \% t$  求得)。此步骤下,  $n, d, e$  的值已知, 并且  $e$  和  $d$  的值可以互换。

(2) 加密时, 设需要加密的消息为  $M$ , 加密后的消息为  $c$ , 则  $c = M^d \% n$ 。

(3) 解密时, 令  $m = c^e \% n$ ,  $m = M$ , 解密成功。

在上述步骤中, 公钥为  $\{n, d\}$ , 私钥为  $\{n, e\}$ 。其中  $\{d, e\}$  是成对出现的。

RAS 算法的一些变种算法的安全性被证明等价于大数分解<sup>[33]</sup>, 但其本身的安全性还未被理论证明, 攻击者若分解出  $n$ , 就可以轻而易举地窃取信息。目前 140 多个十进制位的素数已经能够被分解出来, 所以  $n$  必须尽可能大, 来抵御

破译。

RAS 算法的优点在于算法简单易于实现,双密钥的特点较单密钥而言不易泄露。缺点在于速度太慢,要在保证安全性的前提下尽可能使用长度较长的密钥,这就意味着系统的运算处理代价很大,相比于前一节的对称算法而言,运算量要大几个数量级。就本文的燃气数据边界监控系统而言,需要兼顾安全性和效率,因此没有将该算法作为备选方案。

#### 4.3.2 其它加密算法

1976 年发明的 D-H 算法是第一种非对称加密算法,与 RAS 算法的区别在于 D-H 生成一种用来作为对称密钥的数值,发送方和接收方需要分别生成一个随机数并推导出公开值,双方使用自己的随机数并与对方的公开值组成对称密钥。该算法的优势在于每次都能使用一组新值来作为对称密钥,RAS 算法的私钥一旦被别人掌握,就会窃取所有使用这一对密钥发送的加密信息。它的缺点在于计算密集性,如果有人恶意请求大量密钥,那么被攻击方就会消耗大量资源。

1985 年发明的 Elgamal 算法的基础是 D-H 算法,其与 RAS 算法功能类似,一般都能用于数据加密和数字签名。Elgamal 算法与 RAS 算法的区别在于,前者是一个单向的过程,公钥加密,私钥解密。A 公钥加密给 B 发送信息,B 可以用私钥解密,但是 B 无法给 A 发送加密信息让 A 解密。该算法的缺点在于其密文成倍数扩大。

### 4.4 加密方案

#### 4.4.1 加密算法比较

对以上的几种加解密算法经过对比分析后,总结如表 4-2 所示。从表 4-2 可以看出,相比于 DES,RAS 的密钥长度,AES 的密钥长度适中,在保证安全性的同时,AES 的运行速度较快,消耗的资源较少,综合性能高于 DES 和 RAS 算法。

表 4-2 AES,DES 和 RAS 的性能比较

算法	密钥长度	运行速度	安全性	消耗资源
DES	56 位	慢	低	多
AES	128, 192, 256 位	快	高	少
RAS	大于 500 位	慢	高	多

因此，在保证系统整体安全性和效率的前提下，选用 AES 算法来进行数据以及文件名的加密是最适合的一种加密方案。

4.4.2AES 算法的实现

该算法的原理过程主要包含  $N-1$  次字节替换变换，行移位变换，列混淆变换和轮密钥加变换，其中列混淆变换在最后一轮不出现。具体的执行流程如下：

(1) 字节替代 *SubBytes()*

```
static const uint8_t sbox[256] = {
    //0    1    2    3    4    5    6    7
    0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5,
    0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab, 0x76,
    0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0,
    0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0,
    0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc,
    0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31, 0x15,
    0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a,
    0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75,
    0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0,
    0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84,
    0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b,
    0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf,
    0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85,
    0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8,
    0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5,
    0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2,
    0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17,
    0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73,
    0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88,
    0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb,
    0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c,
    0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79,
    0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9,
    0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08,
    0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6,
    0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a,
    0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e,
    0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e,
    0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94,
    0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf,
    0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68,
    0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16 };
```

图 4-6 S 盒的构造



这一过程的核心是构造 S-Box(图 4-6),将原始状态矩阵  $W$  中的每一个字节都经过 S 盒变换成新的字节(图 4-7)。

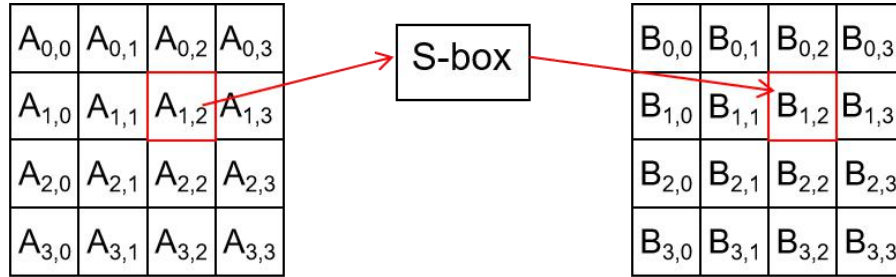


图 4-7 字节替代示意图

S 盒是由 256 个字节组成的  $16 \times 16$  的矩阵，（在代码中构造时不以二维数组的方式构造，直接构造一维数组，在后续找对应行列的值进行替换时利用行列关系找到 256 个值中的对应值），直接通过乘法逆变换和仿射变换操作，使得输入输出之间没有相关性，用公式表示为式(4-5)。

$$S(b) = f(g(a)) \quad (4-5)$$

其中  $g()$  代表乘法逆变换，具有非线性特性。 $f()$  表示仿射变换，即与一个矩阵先相乘再相加。

最后通过 S 盒查表的方式完成最终操作，其中原始状态矩阵中字节的先后四位分别作为 S 盒中的行列号，经过查表置换成新的矩阵。

## (2) 行移位 $ShiftRows()$

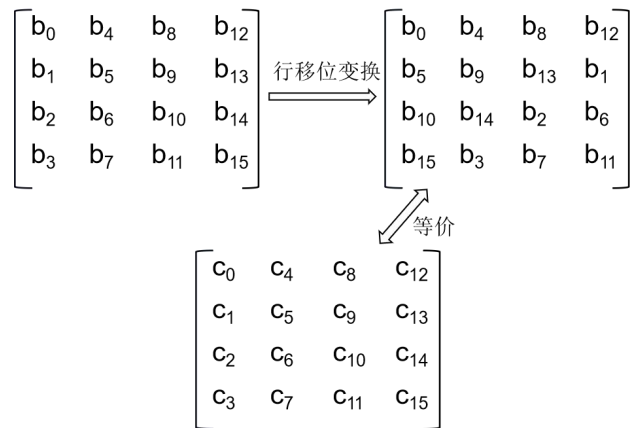


图 4-8 行移位变换

该变换是对状态矩阵的行进行移位操作,一般设  $N_b=4$ 。状态矩阵中第一行

的元素不变，第二行左移 1 个字节，第三行左移 2 个字节，第四行左移 3 个字节。这一过程是线性变换，使状态矩阵的列全部打乱，得到新矩阵 C，具体变换过程如图 4-8 所示。

### (3) 列混淆 *MixColumns()*

对行变换后得到矩阵的每一列通过映射关系得到新的列。该过程可以看作是由列构成的多项式与最高次数小于 4 的另一个多项式（任意且固定）相乘，最后进行取模运算来得到新的矩阵(图 4-9)。这一映射关系表示为式(4-6)。

$$d(x) = c(x) * a(x) \bmod(x^4 + 1) \quad (4-6)$$

其中  $c(x)$  为行移位得到的新矩阵对应列， $d(x)$  为列混淆后得到新矩阵对应列， $a(x)$  为一个任意但固定的次数小于 4 的多项式。具体过程如图 4-10 所示。

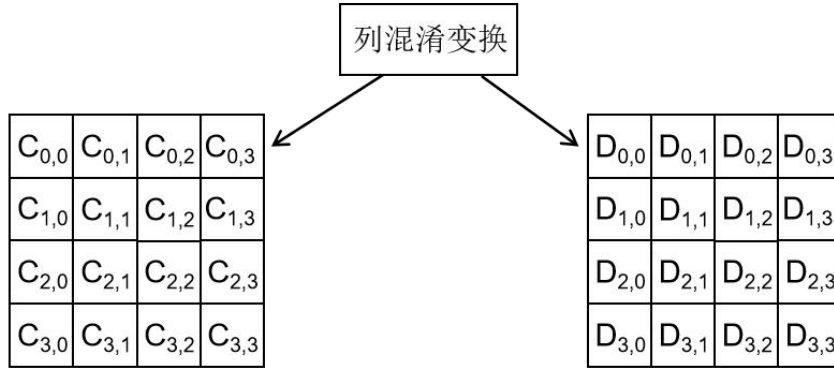


图 4-9 列混淆变换

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

图 4-10 列混淆过程举例

### (4) 轮密钥加 *AddRoundKey()*

$$\begin{bmatrix} d_0 & d_4 & d_8 & d_{12} \\ d_1 & d_5 & d_9 & d_{13} \\ d_2 & d_6 & d_{10} & d_{14} \\ d_3 & d_7 & d_{11} & d_{15} \end{bmatrix} \oplus \begin{bmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{bmatrix} = \begin{bmatrix} d_0 \oplus k_0 & d_4 \oplus k_4 & d_8 \oplus k_8 & d_{12} \oplus k_{12} \\ d_1 \oplus k_1 & d_5 \oplus k_5 & d_9 \oplus k_9 & d_{13} \oplus k_{13} \\ d_2 \oplus k_2 & d_6 \oplus k_6 & d_{10} \oplus k_{10} & d_{14} \oplus k_{14} \\ d_3 \oplus k_3 & d_7 \oplus k_7 & d_{11} \oplus k_{11} & d_{15} \oplus k_{15} \end{bmatrix}$$

图 4-11 异或运算过程

将列混淆变换后得到的矩阵中列与扩展后的子密钥做异或（相加）运算，

得到的新矩阵作为下一轮迭代的输入矩阵。该过程如图 4-11 所示。

综上，加密算法的构成完成，只需进行相应次数的迭代即可。其中列混淆变换在最后一轮变换中不出现。解密算法原理参照加密算法。

## 4.5 加解密算法的软件实现

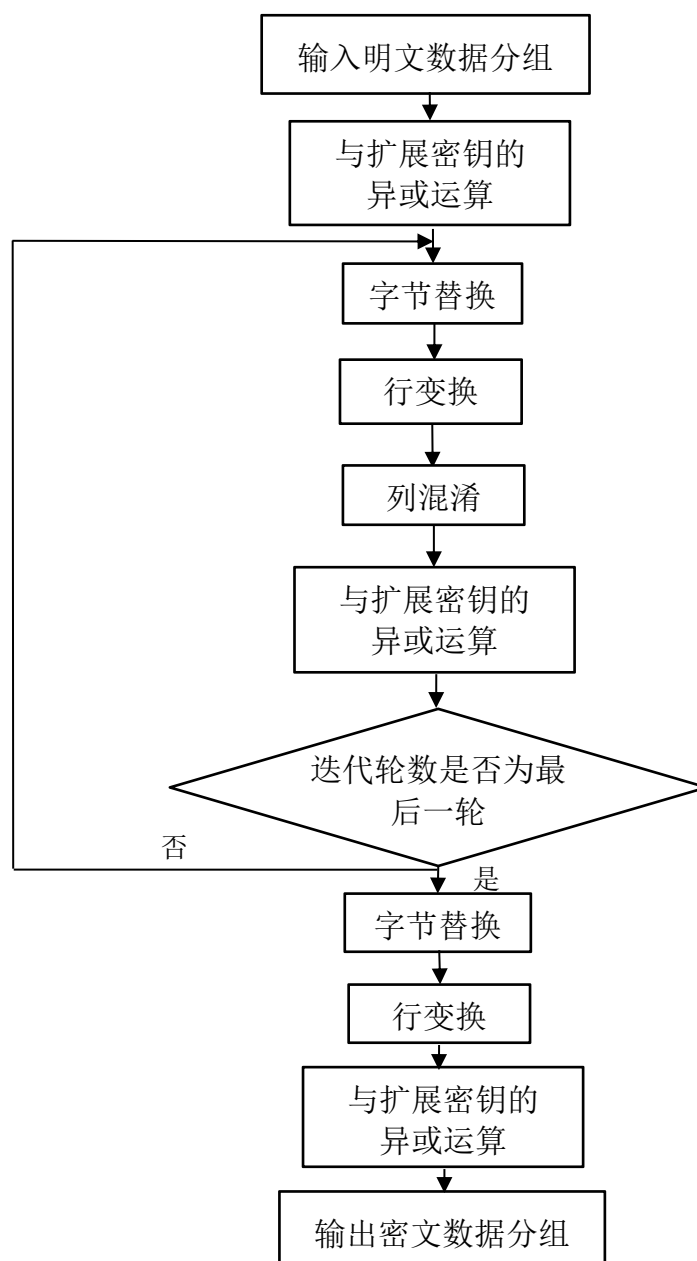
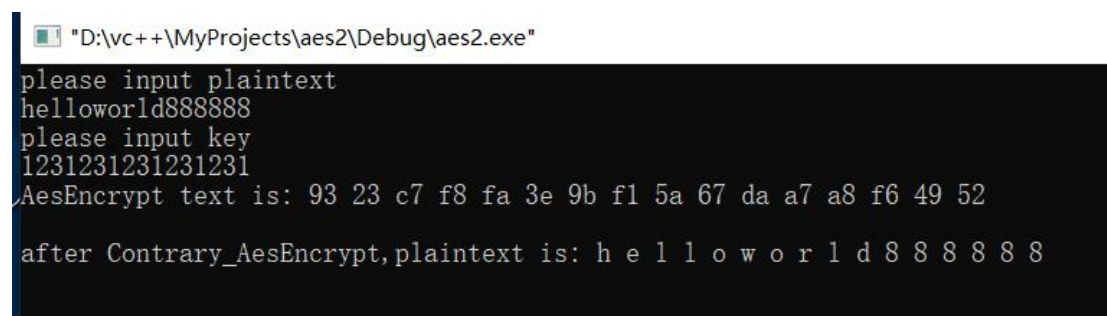


图 4-12 AES 算法流程图

根据上述原理，按照图 4-12 所示的算法流程进行软件编程。由于采取的是

128 位密钥加密，因此一共需要进行 10 次轮变换，且最后一次轮变换中不包含列混淆变换。在 VC++6.0 上进行软件编程编译通过后运用仿真软件测试 AES 算法的正确性，结果如图 4-13。



```
"D:\vc++\MyProjects\aes2\Debug\aes2.exe"
please input plaintext
helloworld888888
please input key
1231231231231231
AesEncrypt text is: 93 23 c7 f8 fa 3e 9b f1 5a 67 da a7 a8 f6 49 52
after Contrary_AesEncrypt,plaintext is: h e l l o w o r l d 8 8 8 8 8 8
```

图 4-13AES 加解密功能实验

图 4-13 中可以看出代码编译后加解密功能正常，之后将 AES.C 和 AES.H 文件移植到嵌入式系统总工程文件中并调试，在第五章中进行总体的实验测试。

## 4.6 本章小结

本章在目前广泛使用的几种加解密算法中进行分析比对，由于 AES 算法具有密钥长度合适、安全性高、运行速度快、消耗资源少等优点，因此选用了 AES 算法作为系统的数据加密算法。根据 AES 加密原理，利用主要的四种变换，通过 c 代码的编写使其适用于系统并验证了该算法的功能。

## 第五章 系统测试及实验结果

本章完成了数据边界安全监控装置的硬件制作和调试，设计了相关硬件驱动程序，将改进的 AC\_HA 模式匹配算法和 AES 数据加密算法移植到嵌入式系统平台上，搭建好的嵌入式系统实物如图 5-1 所示。软件在 KEIL5 环境下进行编程，调试通过后用 JTAG 接口将程序的可执行代码下载到板上运行。由于数据边界安全监控装置本身不具有显示模块，因此通过查看串口输出，以及手机客户端和服务器上的显示来进行系统测试。

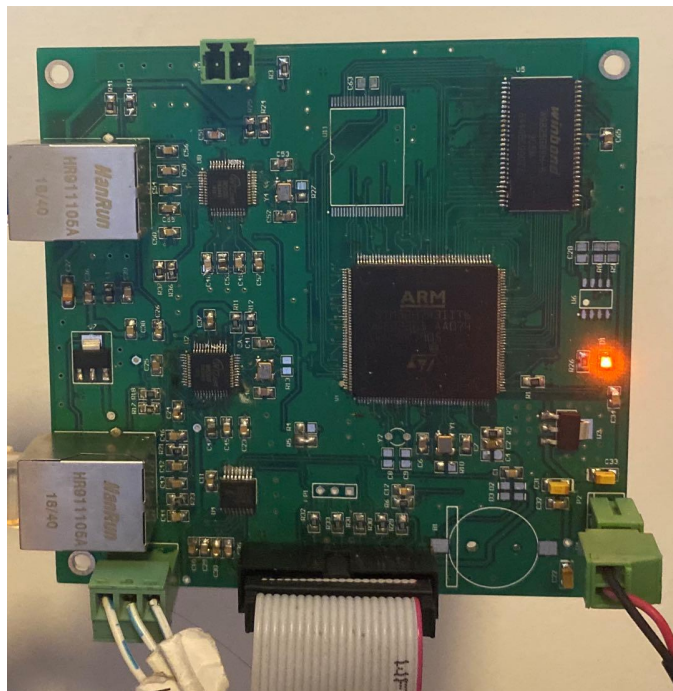


图 5-1 系统组成实物图

数据边界安全监控系统需要与客户端和服务端通信，本实验使用内网进行测试，将服务器端和边界系统连入同一路由，网关为 192.168.1.1，子网掩码为 255.255.255.0。使用手机 APP 客户端作为发送信息方，与监控装置在同一网段中，手机 app 客户端 IP 地址为 192.168.1.100。服务器端更改设置 IPV4 地址为 192.168.1.90，由册本解析服务器和图片解析服务器来接收图片文件和册本文件，其中服务器接收册本文件的端口号为 3009，接收图片文件的端口号为 3008。数据边界安全监控系统的客户端 IP 地址设为 192.168.1.31，服务器的 IP 地址设

为 192.168.1.33。册本文件的收发端口为 8091，图片文件的收发端口为 8090。  
本章所有实验都基于以上实验设置进行。

## 5.1 网络通信测试

本节实验目的是：测试网络接口能正常连接并通信，相对应的手机端与边界系统的服务器，以及边界系统的客户端和服务端都要保持正常连接。

本文使用的实验平台如下：硬件设备：数据边界安全监控装置，无线路由器，若干网线；客户端设备：安卓手机及 APP；服务器接收：服务器、图片和册本协议解析服务软件；网络调试助手，串口调试助手。

### 5.1.1 实验一：手机与边界设备通信

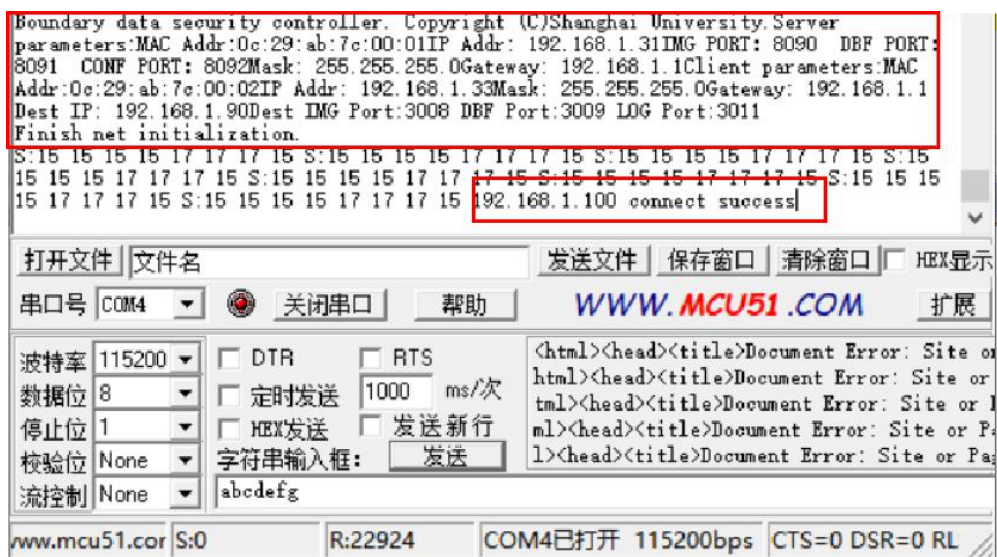
由于数据边界安全监控不能显示输出，借助网络调试助手作为辅助，测试手机端的发送和边界系统的接收功能是否正常。

实验测试结果如图 5-2 所示。



(a) 手机端连入





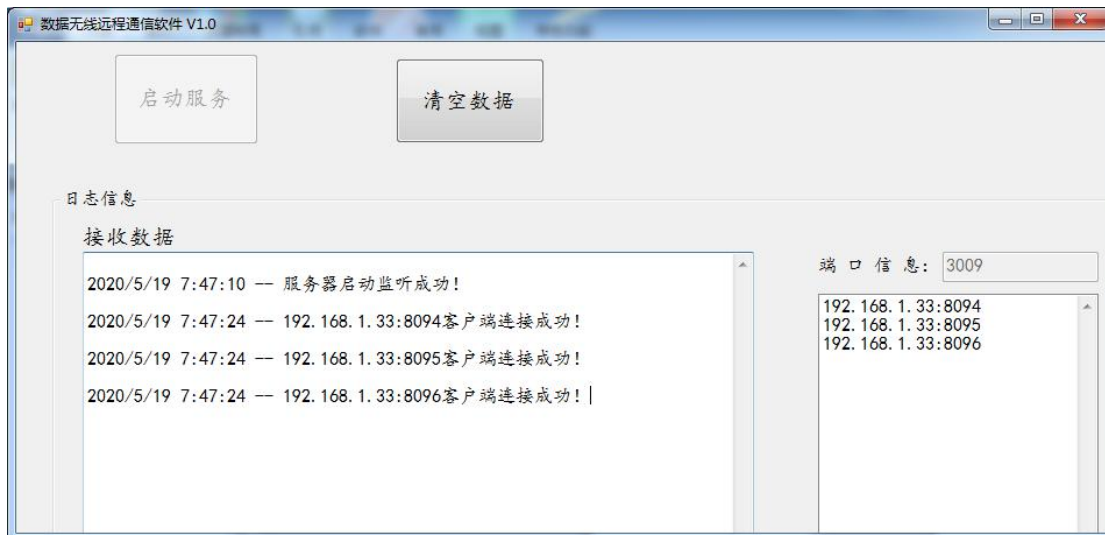
(b) 边界系统连接手机端成功

图 5-2 连接显示界面

由图 5-2 可以看出,手机客户端和本监控设备的服务器端都可以正常连接。

### 5.1.2 实验二：边界设备与服务器通信

使用服务器端的册本文件协议解析服务软件和图像文件协议解析服务软件分别与边界设备客户端进行测试连接。结果如图 5-3 所示。



(a) 边界设备连入册本服务器



(b) 边界设备连入图像服务器

图 5-3 服务器解析器界面

图 5-3 是册本解析服务软件和图像解析服务软件的界面，可以看出，边界系统可以与电脑端服务器进行正常连接通信，图中 8094，8095，8096 为册本文件客户端口号，8090，8091，8092，8093 为图像文件客户端口号。

## 5.2 数据包加密测试

本节实验目的是对经过加密的数据包使用密钥进行解密，对比解密后的图片特征信息是否与原始图片的文件名相符，并对比解密后的册本文件中的数据内容是否正确，比较图片文件的文件名是否为 XXXXXXXXX（8 位户号）+XXXXXXXXXXXXXXXXXX(14 位时间戳)形式。本文使用的实验平台同 5.1。

实验过程及准备：将编写完成的 AES.C, AES.H 文件添加到总工程文件中，修改主函数，设置密钥 aes\_key="SGASDATABOUNDARY"。在手机 APP 端设置好发送时使用的加密密钥，运用第四章中的 AES 加密算法对要上传的文件进行加密，边界设备接收到加密的数据包后使用统一密钥对其进行解密之后与添加的特征信息进行比较，不符合的文件直接过滤掉，不会发送到对应的解析服务器。其中对图像文件中的特征信息是由图片本身的文件名通过 AES 加密算法生成，再插入数据包中，因此接收时需要从相应位置找到对应的密文并进行解



密比对。

数据包上传时，在串口调试助手上显示的实验结果如图 5-4 所示。



(a) 册本文件的特征信息解密



(b) 图像文件的文件名信息解密

图 5-4 数据解密比对

其中图 5-4(a)为册本特征信息提取的加解密结果，图 5-4(b)为图像文件的文件名加解密结果，图 5-5 为图像文件解析服务器的接收结果。

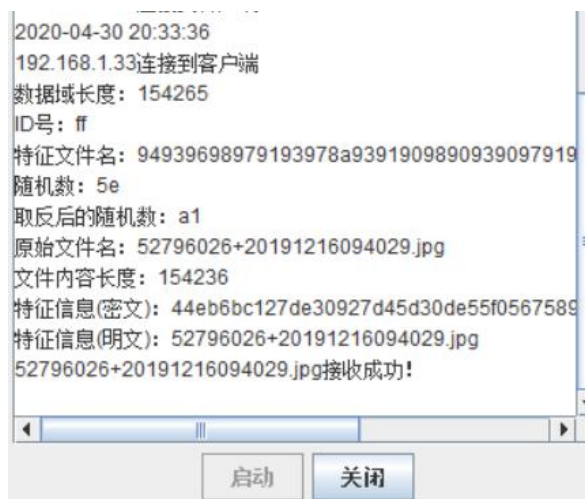


图 5-5 图像解析服务器收到数据

将文件名改成不合法的格式再发送,数据包上传失败时的实验结果如图 5-6 所示。



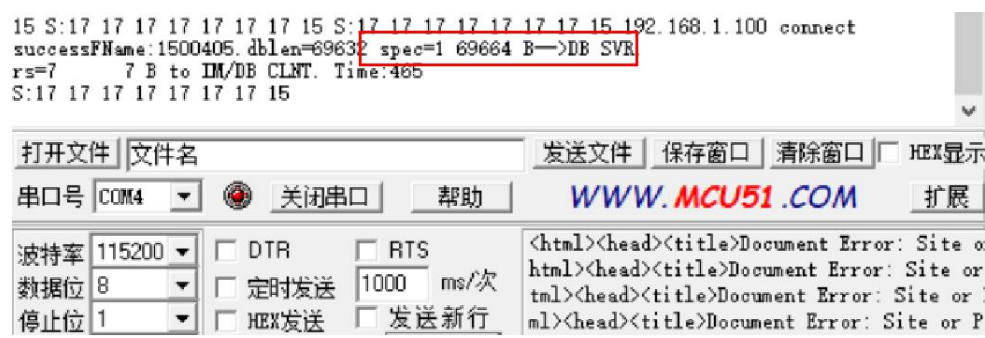
图 5-6 数据上传失败客户端显示

由此可见,加密算法在系统中得到了验证,可以从串口调试助手和协议解析服务软件看到加解密的密文明文以及特征信息。

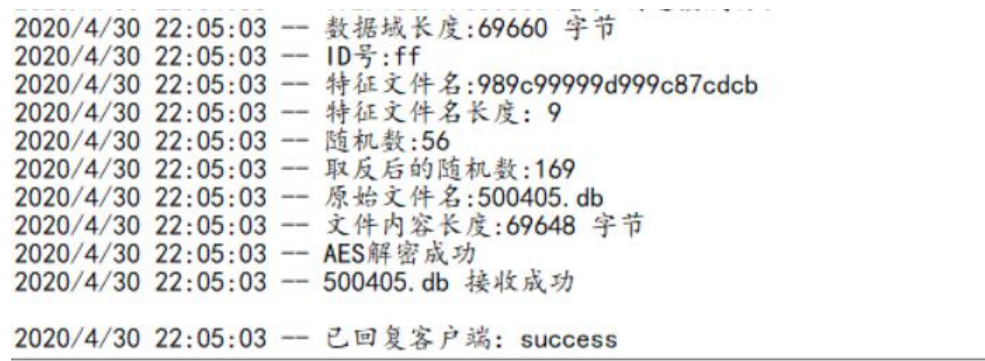
### 5.3 特征字匹配测试

本节实验目的是:设置 DB 文件中含有的多个标志性字段作为文本匹配字段,验证改进后的 AC\_HA 算法的内容过滤功能是否成功。本文使用的实验平台与 5.1 相同。

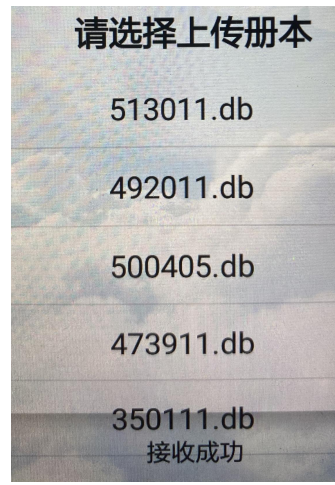
实验准备及过程：设置字段为“TABLE”，“SQLite”（含有大小写要求）并发送文件名符合规范的正确册本。设置障碍字段如“TALLE”测试系统能否将前面发送的正确册本文件数据包过滤掉。设置变量 spec 为匹配标志，等于将所有字符的匹配结果相与的值，若 spec 为 1 则说明匹配成功，若 spec 为 0 则匹配失败。设置正确字段得到的实验结果如图 5-7(a)(b)(c)所示。



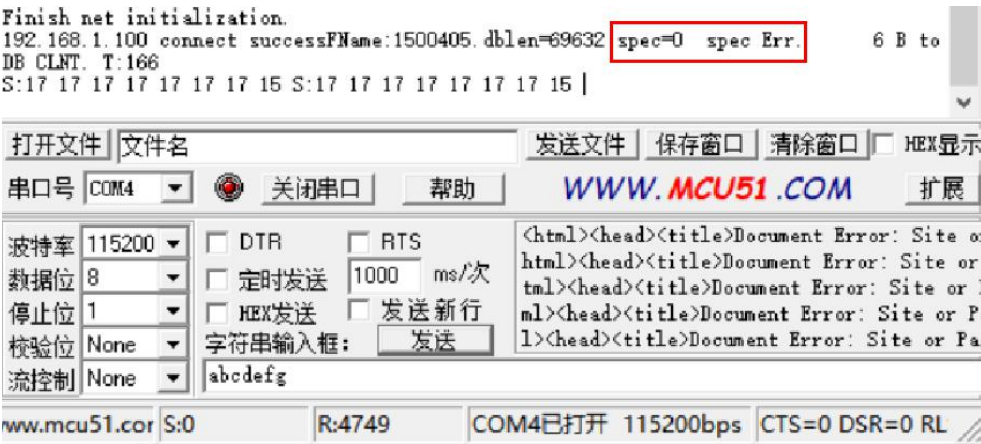
(a) 串口输出信息



(b) 册本解析服务器输出信息



(c) 数据接收成功客户端显示



(d) 设置障碍字段串口输出

图 5-7 输出显示界面

设置障碍字段得到的结果见图 5-7(d)，解析器上没有接收到文件，手机端显示服务器接收失败（见图 5-6）。实验结果表明，匹配算法在系统中运行正常，能够成功过滤掉不含特征字段的数据包。

## 5.4 IP 地址过滤实验

本节实验目的是：验证某一 IP 短时间内发送不合规的数据包达到三次以上时，系统是否能够自动断开连接，由于手机端没有设置显示连接情况，调试时使用网络调试助手查看连接。本文使用的实验平台同 5.1 节。

实验过程及准备：打开网络调试助手，发送多次不合规的文件至边界设备，验证系统是否会自动断开与该 IP 地址的连接。

实验结果如图 5-8，5-9 所示。

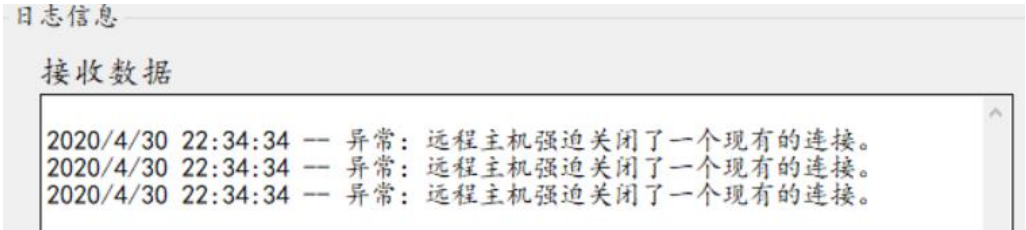


图 5-8 服务器强制断开连接





图 5-9 边界系统与客户端断开连接

从实验结果可以看出，该系统可以断开发送多次不符合要求数据包的 IP 连接，有效防止某些设备恶意占用端口发送大量问题文件使得系统资源耗尽。

5.5 实验结果分析

对系统进行总体功能测试，测试结果如表 5-1 所示。由表可以看出，系统可以过滤掉不符合要求的问题文件，例如不含特殊字段或文件名格式错误的文件都会被过滤掉，只有内容和文件名合规的文件才能被服务器接收。

表 5-1 系统功能测试示例

文件类型	文件名	特殊字段	IP 地址	端口	是否接收
DB	1513011.db	TABLE,SQLite	192.168.1.100	8091	是
	52796026+201				
JPG	91216094029.jpg	文件名加密	192.168.1.100	8090	是
DB	1350111.db	TABLE,SQLite	192.168.1.100	8091	否（不含特殊字段）
	527960262019				否（文件名格式错误）
JPG	1216094029.jpg	文件名加密	192.168.1.100	8090	否（文件名格式错误）
	g				

## 5.6 本章小结

本章主要进行了四种实验——网络通信实验、数据包加密测试、匹配测试和 IP 地址过滤测试。其中网络通信测试验证了边界系统和手机客户端以及电脑服务器端之间的正常通信，保证了后面实验的正常进行。数据包加密测试，匹配测试以及 IP 地址过滤测试验证了第二章中的三层过滤都是有效可行的，可以抵御相应的攻击。总的看来，相比于没有三层过滤的系统而言，系统整体的安全性有了很大的提升。

## 第六章 总结与展望

### 6.1 总结

燃气数据的安全关系到燃气供应网络的安全和社会经济的稳定运行, 本文围绕如何提高自动抄表系统中燃气数据服务器的安全性展开研究。在研究分析当前燃气自动抄表技术和网络安全问题的基础上, 研究燃气数据边界安全监控系统的实现方案, 设计并实现了智慧燃气数据边界安全监控系统。本文的研究工作总结如下:

(1) 提出了智慧燃气数据边界安全监控系统的实现方案。方案在燃气服务器前端部署一套数据边界安全监控设备, 用于对客户端上传的数据包进行过滤。设备作为中间服务器负责接收客户端发出的数据包, 也可以作为中间客户端负责发送通过过滤的数据包。过滤方法采用三层过滤方案, 即使用 IP 地址过滤、文件名过滤和内容过滤来防止一系列的非法数据包和网络攻击, 在客户端和服务端之间铸成一道过滤网, 筛除不合法的数据包, 使燃气服务器的安全性得到保障。

(2) 对深度包过滤技术进行研究, 提出了 AC 算法的改进算法—AC\_HA 算法。在研究多种单模式匹配算法和多模式匹配算法的原理和特点的基础上, 对单模式匹配算法的 BM 算法和多模式匹配算法的 AC 算法的结合改进算法做了进一步的改进, 加入 hash 表来进行字符删减, 过滤掉不需要进行匹配的字符文本, 将原文本字符缩短之后再行匹配, 并且在匹配阶段使跳跃距离尽可能逼近最短字符串长度, 从而减少匹配次数, 提升了算法的效率。将该算法用于对上传文件内容过滤中, 将 DB 文件中包含的关键字字段设置为模式串, 将 DB 文件的整体内容作为文本串进行比对, 以确定文件是否需要被过滤。仿真实验表明, 提出的 AC\_HA 算法时间性能优于 AC 算法及 AC\_BM 算法。

(3) 研究了数据加密方法, 在研究分析 AES, DES, RAS 等加解密算法基础上, 采用最适合本系统的 AES 算法对数据内容进行加密。通过对 128-AES

算法核心的字节替代、行移位变换、列混淆变换、轮密钥加变换等原理的分析，设计了加密算法的软件，并进行功能测试，验证了算法的有效性。

(4) 设计制作了基于嵌入式系统的智慧燃气数据边界安全监控设备的硬件，将模式匹配算法和数据加密算法移植到嵌入式系统平台，开发了燃气数据边界安全监控软件。对系统的总体性能进行了测试，结果表明，本文研究的燃气数据边界安全系统能够有效过滤客户端的上传数据，将合法数据包转发到服务器，而将非法数据阻挡在服务器之外，证明了本文设计的燃气数据边界安全监控方法的有效性。

随着无线通信和信息技术的快速发展，燃气自动抄表系统也逐步普及，随着智慧燃气的数据上传量越来越多，其所面临的安全问题也不容小觑，所以提升燃气系统的安全与效率也就成了被重点关注问题。传统的燃气系统对客户端上传的数据采用全部接收的方式，这无疑对系统处理数据的能力是一种挑战，加之一些短时间大量的恶意无效信息的上传很容易导致系统崩溃，影响燃气系统的正常功能。本论文的主要研究为解决燃气服务器及数据安全提供了一种有效的解决途径，可以在很大程度上解决燃气数据边界的安全问题。同时，本文研究的数据边界安全监控方法也可推广到其它领域，如电表、水表和其它系统的数据安全防护系统中，为维护信息系统安全提供技术手段。

## 6.2 展望

本文研究并实现了智慧燃气数据边界安全监控系统，使燃气系统的整体安全性得到了很好的提升。但是由于时间不足和能力有限，系统还存在不足的地方，未来还可从以下几个方面进行更加深入的研究和改进：

(1) 本文设计的系统虽然采用高性能的嵌入式控制器，但其处理能力还不是很强，能够连接的客户并发数较少。未来可以考虑采用功能更强的运行平台，以解决大量用户并发时的数据瓶颈问题。

(2) 对系统的实验和验证都只处于实验室搭建的内网环境模拟测试，暂时未进行真实网络的模拟，现实的网络环境繁多复杂，对互联网环境下的安全性



还未得到验证。未来准备将系统小范围投入使用，检验其系统能力，特别是在面对各种攻击时能否及时处理好信息并保证正常的上传。

（3）本文涉及到的深度包解析算法中的改进算法 AC\_HA 算法还有待改进，实验对比发现其相比于 AC\_BM 的时间性能的优势不够明显。又由于其删减文本串的操作方式使得系统需要开辟更大的空间去存储新的删减后的文本串，这样一来系统需要存储原文件到缓存，增加了缓存压力。如果大量用户上传数据，可能会使系统内存紧张，后续会改进对系统的内存清理设置更好的机制来解决这一问题。

## 参考文献

- [1] 高顺利,吴荣,吴波,李彦爽.智慧燃气研究现状及发展方向[J].煤气与热力,2019,39(02):23-28+46.
- [2] 杨建华,冯海鹏.远传抄表技术的应用现状和发展趋势[J].煤气与热力,2008,28(11):70-73.
- [3] 屈奋雄,郭晓峰,王键,屈建雄.我国远传抄表现状与存在问题[J].中国住宅设施,2004(11):28-30.
- [4] 李中阳,黄君委,卢智颖,陈海棠.基于 NB-IoT 技术无线计费系统的设计研究[J].自动化与仪表,2019,34(03):105-108
- [5] 何进,仲元昌,孙利利,姚博文.物联网燃气表系统的设计与实现[J].电讯技术,2019,59(06):719-723.
- [6] 乔珩,沈晓东,申粤.不同抄表模式下民用智能燃气表管理及应用[J].煤气与热力,2019,39(03):39-41+46.
- [7] 杨泽渠. 智能电网中用户数据安全及隐私保护研究[D].浙江大学,2017
- [8] 石英春.基于 RFID 燃气控制器装置的设计[J].仪表技术,2016(04):30-31+40.
- [9] 吴吉义,李文娟,黄剑平,章剑林,陈德人.移动互联网研究综述[J].中国科学:信息科学,2015,45(01):45-69.
- [10] 秦丰林,段海新,郭汝廷.ARP 欺骗的监测与防范技术综述[J].计算机应用研究,2009,26(01):30-33.
- [11] Moon D, Lee J D, Jeong Y S, et al. RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks[J]. The Journal of Supercomputing, 2016, 72(5): 1740-1756
- [12] Prabadevi B, Jeyanthi N. A framework to mitigate ARP sniffing attacks by cache poisoning[J]. International Journal of Advanced Intelligence Paradigms, 2018, 10(1-2): 146-159.
- [13] Kaur G, Malhotra J. An integrated approach to ARP poisoning and its mitigation using empirical paradigm[J]. International Journal of Future Generation Communication and Networking, 2015, 8(5): 51-60.

- [14]Jeong Y S, Lee S H. Personal Information Leakage Prevention Scheme of Smartphone Users in the Mobile Office Environment[J]. Journal of Digital Convergence, 2015, 13(5): 205-211.
- [15]程艳艳.地址解析协议攻击后网络数据防泄漏方法[J].科学技术与工程,2017,17(34):273-277.
- [16]Chae C J, Shin Y J, Choi K, et al. A privacy data leakage prevention method in P2P networks[J]. Peer-to-Peer Networking and Applications, 2016, 9(3): 508-519.
- [17]Zhonghua Z L C, Xiaoming W. Research on detection methods of CC attack[J]. Telecomm. Science,, 2009: 62-65.
- [18]李硕,张权.基于蜜罐的 CC 攻击防护体系 [J].信息安全与通信保密,2015(09):99-102.
- [19]Chun-Tao X, Xue-Hui D, Li-Feng C, et al. An algorithm of detecting and defending CC attack in real time[C]//2012 International Conference on Industrial Control and Electronics Engineering. IEEE, 2012: 1804-1806.
- [20]景泓斐,张琨,蔡冰,余龙华.基于 BP 神经网络的应用层 DDoS 检测方法[J].计算机工程与应用,2019,55(20):73-79.
- [21]Zargar S T, Joshi J, Tipper D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks[J]. IEEE communications surveys & tutorials, 2013, 15(4): 2046-2069.
- [22]董哲,唐湘滢,程杰仁,张晨,林福生.基于 HMM 时间序列预测和混沌模型的 DDoS 攻击检测方法[J].计算机工程与科学,2018,40(12):2164-2172.
- [23]程杰仁,罗逸涵,唐湘滢,欧明望.基于 LSTM 流量预测的 DDoS 攻击检测方法[J].华中科技大学学报(自然科学版),2019,47(04):32-36.
- [24]Idhammad M, Afdel K, Belouch M. Semi-supervised machine learning approach for DDoS detection[J]. Applied Intelligence, 2018, 48(10): 3193-3208.
- [25]Chen Y, Pei J, Li D. DETPro: A High-Efficiency and Low-Latency System Against DDoS Attacks in SDN Based on Decision Tree[C]//ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019: 1-6.
- [26]刘春晖,黄宇,宋琦.一种改进的 AC 多模式匹配算法 [J].计算机工程,2015,41(10):280-285.

- [27]陆琳琳,田野.基于确定有限状态自动机的改进多模式匹配算法研究[J].计算机应用与软件,2013,30(07):321-323+330.
- [28]舒银东. 基于有限状态自动机的多模式匹配算法研究[D].合肥工业大学,2011.
- [29]赵远,秦拯,张大方,武年华.一种面向高速网络的模式匹配算法的设计与实现[J].微计算机信息,2010,26(12):167-168.
- [30]Srinivas N S S, Akramuddin M D. FPGA based hardware implementation of AES Rijndael algorithm for Encryption and Decryption[C]//2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). IEEE, 2016: 1769-1776.
- [31]刘兰,姚行中,王振宇,杨晓非.适用于 CCSDS 的“一帧一密”加/解密方案的 FPGA 实现[J].计算机工程与应用,2015,51(10):84-88.
- [32]赵君.基于 64 位处理器的 Android 平台优化 AES 加密算法[J].哈尔滨理工大学学报,2017,22(03):8-12.
- [33]宋维平.RAS 算法与安全性的探讨[J].工业技术经济,2009,28(03):137-138.

## 作者在攻读硕士学位期间的科研成果

【1】 冯玉田，高萌. 一种数据安全边界监控方法. 中国发明专利 CN  
202010400440.1

## 致 谢

岁月不居，时节如流，回首间，一年延长一年宝山的学习生活就要拉下帷幕了，仿佛昨天还是刚刚来学校报到时与实验室老师师兄师姐初见的日子，那时对研究生生活的好奇写满在了天真懵懂的脸上。这两年的时光，让我不仅在年龄上有了增长，在学习见识，与人沟通交流方面也有了很大的提升，最重要的是明确了自我定位，也让我对迈入社会的生活有了更大的勇气和更坚定的信息。这一切的成长都离不开我身边老师，家人，朋友，同学，实验室师兄师弟的帮助，在这里对他们表示最诚挚的感谢。

首先最感谢的是我的导师冯玉田老师，从第一次见面老师就给我留下了深刻的印象，总对人笑嘻嘻的非常慈祥。在工作学习中，每次遇到问题时，老师总是悉心的帮我解答；代码调试不通时，老师会坐到我身边帮助我调试；实习和项目有些冲突时，老师给予了很大的理解。在日常生活中，老师把我们当家人一般，会在快放假时请我们一起吃年夜饭，会细心的关心每个人。读研期间有个师兄结婚，整个实验室包括老师都前往苏州为师兄祝贺。每每聊到我们实验室的和谐氛围都觉得自己非常幸运。相信即使再过很多年，我依旧不会忘记冯老师穿梭在办公室和各个实验室的矫健身影，在此衷心的对冯老师表示感谢！

其次要感谢的是实验室 405 的所有兄弟姐妹们，是他们的一路陪伴，在我遇到困难时热心的帮助，不断给予我勇气。其中最要感谢罗涛师兄和涂云轩师兄，不论是科研还是工作都给了我很大的帮助，他们总结的经验方法也让我少走了不少弯路，他们认真科研的样子也为实验室树立了良好的榜样。还要感谢我的同桌杨洁，每次我们都相约一起前往实验室，在我偶尔想要偷懒的时候会给我敲警钟激励我，也会提醒我合理饮食和运动和健康的作息。还要感谢新来的师弟们，为实验室又增添了朝气和活力。

还要感谢我的室友，使我两年的寝室生活丰富多彩。最后要感谢我的父母，一直以来对我都给予很大的支持，是我最坚强的后盾。

写到这里，才发觉在学校校园里已经生活了 18 年，转眼间就要迈入社会的

大门，希望自己不负所托，不忘初心，用最好的姿态迎接全新的未知的未来。

最后对百忙之中评阅本论文的老师们表示衷心的感谢，你们辛苦了！