清 华 大 学

综 合 论 文 训 练

题目：针对罗吉斯回归模型优化算法并行加速的研究

系 别：计算机科学与技术系

专 业：计算机科学与技术

姓 名：彭昊若

指导教师：赵颖 教授

2013 年 6 月 8 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名： 导师签名： 日 期：

中文摘要

罗吉斯回归模型是一个被广泛应用的有监督学习模型。本文着眼于大数据背景下，解决针对罗吉斯回归模型优化算法的计算效率问题。我们的研究主要是在算法层面。我们研究了三种不同的并行计算系统以及三种不同的计算算法来有效地提升算法的可扩展性。Hadoop，Mahout以及Spark是当前的三个为人所熟知的系统。它们都能支持大规模机器学习算法在大数据及上的运算。并行梯度下降和随机梯度下降是两个当前非常经典的优化算法，并且可以专门用于解决罗吉斯回归模型的学习问题。另外，我们还开发出一种新的并行次线性算法。该并行算法是建立在线性次线性算法基础上的。我们会比较这些算法在不同并行系统上的运行结果。实验结果显示，在不同的数据情况以及不同的系统资源下，我们可以选择有针对性的算法更为高效地解决罗吉斯回归模型的学习问题。同时，更进一步的研究表明，对于此种需要较长时间执行的分布式并行计算而言，容错能力既可以体现在系统层面，也可以体现在算法层面，我们可以根据解决问题的需要进行平衡。

关键词：罗吉斯回归模型；大数据；并行计算；次线性方法

ABSTRACT

Penalized logistic regression (PLR) is a widely used supervised learning model. This paper addresses the issue of computational efficiency for solving PLR in big data scenario. We focus on algorithm level and employ three different parallel computation systems with three different types of parallel algorithms to efficiently improve scalability. Hadoop, Mahout and Spark are three existing well-known parallel systems that can be implemented with large scale machine learning algorithms. Parallel gradient Descent and stochastic gradient descent are two existing state-of-art algorithms for parameter optimization in PLR. We also present a novel parallel sublinear method based on its sequential version. We then make a comparison between these algorithms implemented on different systems. As the outcome shows, we can select different algorithms for the purpose of efficiency in different situations of datasets and machine resources. Moreover, fault tolerance can be provided for those lengthy distributed computations both on algorithm level and system level.

Keywords：Penalized Logistic Regression Model; Big Data; Parallel Computing; Sublinear Method

目 录

第1章 引 言 1

1.1 研究背景 1

1.2 论文 1

1.3 论文结构 3

第2章 已有工作 4

2.1 黑洞攻击与防御 4

2.1.1 黑洞攻击原理 4

2.1.2 黑洞攻击的防御机制 4

2.1.3 黑洞攻击的防御算法 4

2.2 数据污染攻击与防御 6

2.2.1 数据污染攻击 6

2.2.2 数据污染攻击的防御机制 6

2.2.3 数据污染攻击的防御算法 6

第3章 WSN TMOTE平台 8

3.1 WSN TMOTE平台简介 8

3.1.1 硬件环境 8

3.1.2 编译环境 8

3.2 WSN TMOTE平台搭建 9

3.2.1 基础路由协议 9

3.2.2 网络监控及调试方法 11

3.2.3 WSN TMOTE平台实现功能 12

第4章 黑洞攻击与防御 13

4.1 黑洞攻击的实现 13

4.1.1 基于WSN TMOTE平台的攻击实现 13

4.1.2 进行黑洞攻击后的效果 13

4.2 黑洞攻击的防御算法分析 14

4.3 算法实现 15

4.3.1 算法简化 15

4.3.2 基于WSN TMOTE平台的防御实现 15

4.4 实测防御效果及分析 16

4.5 黑洞攻击与防御小结 18

第5章 数据污染攻击与防御 19

5.1 数据污染攻击的实现 19

5.1.1 基于WSN TMOTE平台上的攻击实现 19

5.1.2 进行数据污染攻击后的效果 19

5.2 数据污染攻击的防御算法分析 19

5.3 算法实现 20

5.3.1 算法简化 20

5.3.2 基于WSN TMOTE平台的防御实现 20

5.4 实测防御效果及分析 21

5.5 数据污染攻击与防御小结 23

第6章 总结 24

6.1 论文主要工作的总结 24

6.1.1 工作成果 24

6.1.2 工作难点 24

6.2 进一步的研究工作 25

插图索引 26

表格索引 27

参考文献 28

致 谢 1

声明

附录A

附录B

# 引 言

## 研究背景

罗吉斯回归模型（Logistic Regression Model）[14]在机器学习和数据挖掘领域的作用越来越大。这个模型主要针对分类问题，并且已经有大量的理论和算法基础。罗吉斯回归模型相对于另一种十分流行的支撑向量模型（Support Vector Machines）[25]，同样是很有竞争力的。其优势是，罗吉斯回归模型不仅往往具有很高的拟合精度，而且还有很强的可解释性。这一点体现在罗吉斯回归模型可以直接估计一个条件类的概率。并且，此优势使得一个可以被罗吉斯回归模型所解决的二分类问题同样可以很接单的扩展为一个可以被罗吉斯回归模型所解决的多分类问题。在本文中，我们将集中于二分类问题。

在机器学习领域中，解决分类问题的模型有很多。我们之所以选择罗吉斯回归模型，是因为它是一个在数据挖掘领域中常用的较为快速为成熟的分类器。它的统计基础使得在罗吉斯回归模型上的研究往往可以给其他分类问题的模型研究带来启发和指导。

近些年来，大规模数据集越来越普遍，从中演化产生的大规模数据应用也越来越多。这些应用的一个核心特征是，训练数据的样本数量非常大，并且数据唯独非常高。例如，在医疗诊断应用中，医生和患者都希望能从上百万条的医疗诊断记录中获得经验，提高诊断正确率，并且一条医疗诊断记录往往有上百维特征可供查询[24]。另一个更明显的例子是现代搜索引擎。这些搜索引擎需要处理十亿级别以上的文本数据以及多媒体数据，并且每条数据的特征空间都在上千维以上[11]。可以想见，如此大规模的数据和高维度数据空间会对机器学习算法在计算能力上提出很高的要求，以致产生很大的挑战。

而罗吉斯回归模型作为其中的典型代表，由于其在PageRank[21]以及Anti-Spam Filtering[1]方面的广泛应用而使其计算效率问题尤为突出。

## 论文工作

本文将从算法层面分析和解决罗吉斯回归模型优化算法的计算效率问题。我们将主要采用并行方式来进行算法加速。

在开发针对罗吉斯回归模型优化算法的并行模式的过程中，我们无法回避这样一个问题：选择什么并行系统，以及选择怎样的并行计算框架。在经过深入调研之后，我们选择了三个独特而流行的系统来进行测试，它们是Hadoop[27]，Mahout[20]以及Spark[29]。

其中Hadoop支持HDFS[3]和MapReduce[7]。而Mahout更像是一个代码库，既可以建立在Hadoop上运行，也可以独立运行。Mahout不能在严格意义上称作为一个系统，但由于其在罗吉斯回归模型上所支持的算法运行框架是独立于Hadoop的，我们在此表述为非严格意义上的系统。同时请读者注意，这并不表明我们认为Mahout和Hadoop完全在一个层次上，因为大部分Mahout所支持的机器学习算法还是依赖Hadoop的。

Spark系统最大的特色在于其对含迭代过程算法的良好支持。它采用了更好的内存和缓存控制策略，并且支持HDFS。我们将会在相关工作部分分别介绍它们的特点。

在针对罗吉斯回归模型的优化算法中，有过很多经典的串行算法。一种的一个代表性例子是随机近似方法。随机近似方法，具体而言又有随机梯度下降（Stochastic Gradient Descent）方法[30]以及随机对偶平均（Stochastic Dual Averaging）方法[28]。它们都可以通过极少量的数据迭代，来趋近于最优优化结果。因而，在训练过程中，当控制预期优化趋近率时，这些算法的运行时间都是正比于数据总规模的，即呈线性。其中最著名的当属随机梯度下降方法。此算法以在线（online）方式运行，我们可以把随机抽取的数据点看作不断引入的数据流。针对此算法进行并行化处理较为困难，但即使是串行，它的运行也十分高效。

如果我们退一步来考察用普通梯度下降算法来解决罗吉斯回归模型的优化问题，尽管作为串行算法它不够高效，但它极易进行并行化处理，从而在大数据背景下大幅提高运行效率，不失为一个潜在的良好算法。我们可以在一次迭代中载入全部数据，以MapReduce的形式计算它们的平均梯度下井方向，从而实现并行。更进一步，我们可以使用次线性方法[22]，更进一步的提高运行效率。由于次线性方法可以充分利用在随机近似方法上，所以解决罗吉斯回归模型的优化问题，同样可以利用次线性方法来降低算法复杂度，提高运算效率。次线性方法在每次迭代中只利用训练数据的一维来代替所有维度进行处理，从而实现加速。我们将在后文的算法部分提出基于该算法的一个并行版本。它可以在保证近似率的进出上更为快速的收敛。

## 论文结构

第1章 引言，介绍了本文的研究背景，论文的相关作，同时简要概括了一下论文结构。

第2章 相关工作，详细介绍了黑洞攻击和数据污染攻击的原理以及防御的方法以及具体算法，在后面将会用到这些算法，并且在后面介绍的实验平台上实现它们。

第3章 WSN TMOTE平台，详细介绍了本次搭建的实验平台的软硬件环境以及基础的路由协议，网络监控，以及上面实现的一些具体功能。其中将具体说明平台提供的一些接口以及如何去使用以及调试。

第4章 黑洞攻击与防御，介绍了黑洞攻击以及前面所说的防御算法在WSN TMOTE平台上具体的实现方法，再对其效果进行分析。

第5章 数据污染攻击与防御，介绍了数据污染攻击以及前面所说的防御算法在WSN TMOTE平台上具体的实现方法，将会对算法进行一定的简化，来保证其可以在平台上正常工作。

第6章 总结，着重介绍了本次研究的一些成果，研究的难点以及如何去克服这些难点。提出下一步可能的研究方向以及该平台的推广。

# 相关工作

## 大数据

### 黑洞攻击原理

在网络中，黑洞是一个空的地址，所有发送到该地址的包都会消失。而黑洞攻击则是人为制造了一个黑洞，阻断封包达到阻断服务的目的[20]。在无线传感器网络中尤为重要，攻击节点会声明自己是正常节点，并且发送正常路由包，声明自己是最优的路由，从而达到吸引流量的目的，等到接收到数据包后，继而再对其进行处理，可以丢弃，也可以篡改，这里将先考虑系统可用性，仅将包丢弃，不篡改数据。

### 黑洞攻击的防御机制

假设有攻击节点声明是最近的节点，那么数据包大多会经过该节点，从而被其截下无法传输，我这里的目的是要让其能够继续工作，大部分消息依然能够到达正常节点。从而我想到了数据冗余，将数据发送多份到不同节点。虽然这个解决办法会产生大量冗余数据，但是可以解决少量攻击节点的黑洞攻击，使得数据可以正常地被接收。这个方法还会产生丢包，但是和原来比起来会好很多。

但是这样会产生一个问题，攻击节点可以获得原始数据，于是我便想到了一种冗余码，将一个数据分成n块，其中每一块都包含了1/k互不相同的信息量，也就是说，只要任意取得其中k份都能还原出原来的数据，这样只要其中k份能够到达正常的节点，那么原始数据就可以被还原。只要攻击节点不截下k份数据就无法获得原始数据。

### 黑洞攻击的防御算法

算法中用到了伽罗瓦域[21]，以为例：

设特征多项式，本质元素满足。

伽罗瓦域中加法和减法是一样的，故而中四个元素可表示为：

表 .1 伽罗华域的三种表示形式

|  |  |  |
| --- | --- | --- |
| 指数形式 | 多项式形式 | 向量形式 |
| 0 | 0 |  |
| 1 | 1 |  |
|  |  |  |
|  |  |  |

加法即为异或，因此有，其他依次类推。

表 .2 伽罗华域GF(4)上的加法表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| + | 0 | 1 |  |  |
| 0 | 0 | 1 |  |  |
| 1 | 1 | 0 |  |  |
|  |  |  | 0 | 1 |
|  |  |  | 1 | 0 |

乘法需要模p(x)，因此有。

表 .3 伽罗华域GF(4)上的乘法表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 |  |  |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 |  |  |
|  | 0 |  |  | 1 |
|  | 0 |  | 1 |  |

首先从应用层获取原始的数据，将数据作如下处理。

首先，原始数据为S，那么将S分成等长的m块。

构造一个多项式。

求得，其中。

这就是里德所罗门码[22]，即为RS码。当时，RS码有纠错功能。

举例：

，分为2块，

有，得到



可得



由此就能得到原始数据，而若是需要能够纠少部分错，需要。但是此处由于设备存储限制，将采用最简单的RS码，即。

通过这种方法，即使攻击者再怎么伪装，都能起到一定的作用。但是若数据被篡改，并且继续被发送到Sink节点，这里无法将其识别出来，需要其他机制，以确保其是原始数据，没有被人篡改过的数据，而下面要说的就是数据污染攻击，会着重讨论一下如何识别出被篡改过的数据。

## 数据污染攻击与防御

### 数据污染攻击

数据污染攻击[23]实际上是一种数据篡改，数据污染顾名思义就是让数据被污染，即不再是原来的数据。攻击者通过各种方法进入网络，并且添加、修改或者删除其中部分的数据，以达到污染数据的目的，经过这样的攻击以后，Sink节点依然能够得到数据，但是得到的是被污染过的数据，这就需要一些验证方法将其识别出来，否则若是将错误的数据当成了正确的数据，则会导致一系列严重的后果，这里将主要讨论一下如何将其识别出来。

### 数据污染攻击的防御机制

首先这里的算法是基于前面针对黑洞攻击的防御算法，而在该算法中，数据被分发到了多个节点，而这里的问题就是如何去检测这多个数据的完整性呢？因此，我们想到了可以采用发送动态验证数据的方法，各个节点收到动态验证数据请求后，将回应一个数据，通过一种算法来验证这些数据的完整性，来识别其是否被污染。该验证请求中将包含一个动态地验证码，回应的信息将根据该验证码来确定，当数据被篡改时，回应的验证码也将改变，此时发起验证的节点将获得这些验证码，并且可以检测出被篡改，从而放弃这些数据。

### 数据污染攻击的防御算法

在这里将用到AS(Algebraic Signature)[24]，假设X是l个由字符组成的串，

，

，运算都是基于上的。

前面是将发送到n个节点上，现在就为其加上AS。

假设可以表示为，先任意从中生成n个元素。



（2-1）

然后随机找n个节点将都发送出去。当这些节点收到验证请求以后，将会返回一个验证信息，这个验证信息经过如下计算得到：



由于



所以只需验证



（2-2）

若不满足，则表明数据被污染，该数据不可用。反之，则为可用，这就是对于数据污染攻击防御的主要算法。

# WSN TMOTE平台

## WSN TMOTE平台简介

### 硬件环境

本次实验平台的硬件部分是伯克利大学开发的Tmote Sky[25]，通过串口可以对其进行烧录，并且可以调用下层的一些接口，进行数据收发，时钟等操作。而由于Tmote Sky与电脑连接时通过USB接口，所以需要安装USB转虚拟化串口的驱动程序USB SERIAL DRIVER以及USB SERIAL PORT。这样系统会虚拟出一个串口，后面的一些烧录操作都是在这个虚拟化的串口上进行的。

面板上有几个用于指示以及调试的LED灯，其中有两个灯是用于指示USB口与电脑传输情况的，另外三个灯是可以在程序中调用的。这就是本次实验中主要用到的硬件环境。

### 编译环境

本次实验使用的语言是基于TinyOS[26]下的nesC[27]语言。nesC语言是一种模块式的语言，首先是一个一个独立的模块，然后通过配线将各个组件连接起来，这样就构成了一个完整的nesC程序。TinyOS中集成了tmote sky的模块，可以直接通过命令进行编译，生成中间程序，并且通过虚拟串口进行烧录。

编译过程中，首先编译各个模块，然后将这些模块通过配线连接起来，进行编译，生成中间的C程序，再对C程序进行编译生成可以在Tmote Sky上执行的可执行程序。

通过eclipse的插件[NESCDT](http://docs.tinyos.net/index.php/NESCDT-_An_editor_for_nesC_in_Eclipse)[28]可以搭建一个环境，不过由于本次实验使用的是TinyOS-1.x，eclipse的插件对此支持不是很全面，只能做到语法高亮。在以后可以考虑利用TinyOS-2.x搭建环境，使得eclipse对其支持更好，可以对其直接在eclipse下进行编译。

## WSN TMOTE平台搭建

### 基础路由协议

Tmote Sky最底层使用的是802.15.4协议[29]，对上面有许多协议，基础路由协议就是建立在这之上的。

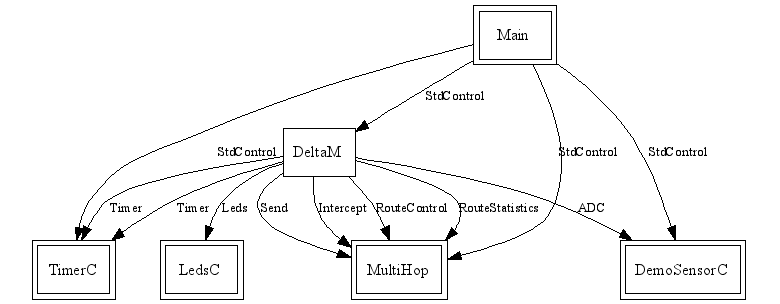
这里首先要介绍一下基础路由协议，即初始路由协议，这一协议比较基本，其中有一些对于下层已有模块的调用，该协议的程序主要可分为两层，上面一层为应用层，主要功能是从ADC(Analog to digital converter)通道，即数模转换通道获取数据，并且将该数据传到下层，这一层比较简单，主要是可以在其上建立一些应用，调用下面的接口进行一些应用层面的数据处理。

图 .1 Delta程序结构图

Delta程序的结构如图3.1。

其中矩形是一些模块，双框的矩形是属于其他程序的，其内部还有一些深层次的结构，可以展开，而单框的矩形主要是本程序中实现的。其中带箭头连线表示配线，将其通过一定接口连接起来，连线中间的字符既是具体接口名称。

首先说一下Demo SensorC这个接口，这个接口就是从ADC通道获取数据，其中实现了一个函数，每当请求的数据准备好了就会触发这个函数。其中还调用了一个TimerC的接口，这个接口主要是用来触发一些事件，当触发这些事件时执行特定的工作。比如在这里每当TimerC触发了一个事件，就让其向Demo SenserC通道请求数据过，并且让其过一定时间后再次触发TimerC，在一定时间后就会再次触发TimerC的触发函数，形成了一个时钟，其中进行一定的操作以后调用下面的接口MultiHop将数据发送出去。另外还有个LedsC接口，这个接口主要是对于Led灯的调用。这里一共有三个灯可以调用，分别是红，黄，绿三个灯，分别有不同的函数调用它们，主要是用来指示当前的状态。这里当发送队列里有数据要发送时红灯会亮，当接收到数据时，黄灯会亮（黄灯在本次实验的硬件上表现为蓝色，但是在接口中是yellow）。

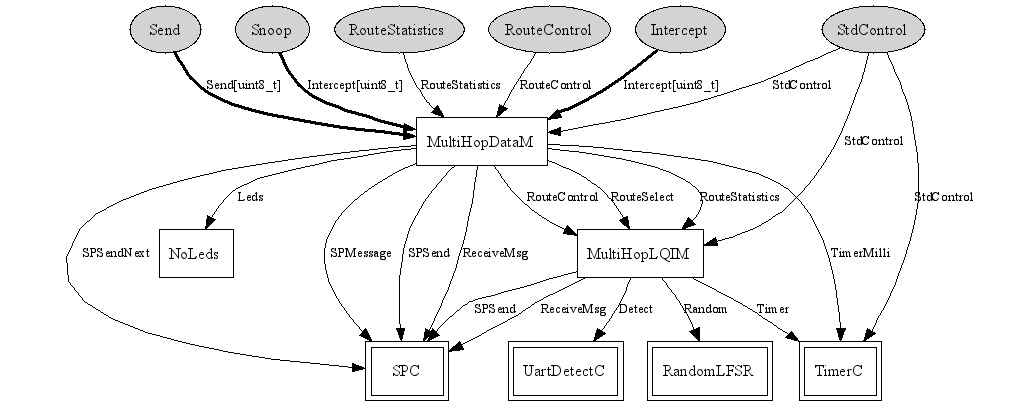
下面着重介绍一下MultiHop层，我们后面的一些修改也是基于这个层修改的，这里基础的路由协议是安装时自带的一个程序经过一小部分修改得到的。这里的修改没有修改其结构，但是后面的修改会对其结构进行修改，增加了许多模块。先来看一下基础路由协议，如图3.2。

图 .2 MultiHop程序结构图

这一模块明显要复杂一些。其中阴影椭圆表示向上提供的接口，而带箭头的粗线则表示向上提供了一个接口数组，可以看到接口中有一个参数是uint8\_t类型的，带了一个8位无符号整型的参数。

首先介绍下本层的各个模块：MultiHop DataM模块主要功能是进行数据的收发，并向上提供接口。MultiHop LQIM模块主要功能是进行一些路由的控制，可以看到一些关于路由的接口都是由该模块向上提供的，比如Route Select接口可以进行路由的选择。

接下来是下层的模块：SPC接口实际上是一个收发数据的模块，可以收发一些数据，我们看到MultiHop DataM模块和MultiHop LQIM模块都调用了该模块的接口，其中MultiHop DataM模块调用该接口进行数据的收发，而MultiHop LQIM模块主要是调用其进行路由广播包的收发。另外该模块也可以对Uart端口发送数据。Uart DetectC主要是检测和电脑是否有连接，板上是通过Uart连接再连接到USB口再虚拟成串口连接到电脑上的，该原始程序中，检测若与电脑建立了连接则将这个节点视作Sink节点，反之则视作一般节点。RandomLFSR主要是生成随机数，TimerC是时钟，这两部分就不展开了。

原始程序中两层是分开的，这样不容易进行扩展，所以将其改动了一下将两层合二为一，文件仍然是分开的，但是放于同一个目录下。原程序是通过检测Uart连接来判断是否是Sink节点，而我将固定一个节点是否是Sink节点，有利于后面的研究。其中需要改一些MultiHop LQIM中的函数，改动地方也比较多，这里只是简单地提一下。

具体的收发过程比较复杂，这里也就不再详细描述了。

### 网络监控及调试方法

安装完Tmote Sky的程序后其中有一个监控程序，用java编写，首先与Uart端口[30]建立连接，通过监听Uart端口，获取Tmote Sky向Uart端口发送的数据包，程序中可以将这些数据包输出到文件以供调试，后面的一些展示也以这些数据包为主。原程序中只记录了一些MultiHop层的数据包，我将其进行了一些改动，使其记录所有发送到Uart端口的数据包以方便调试。

监控程序是在TinyOS中利用java编译，并且在TinyOS中通过一个变量来指定监听的端口，再利用java来运行该程序。

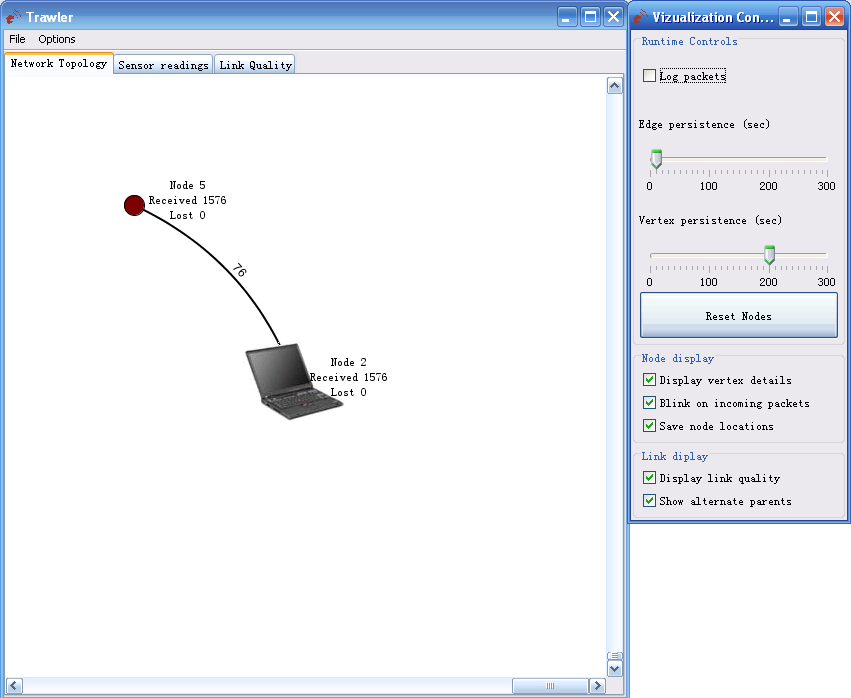
监控程序的界面如图3.3。

图 .3 监控程序界面图

右边是一个控制面板，可以调一些参数，还有记录接收到包的功能。左边面板主要是看一些连接状态以及应用层接收到的数据。

### WSN TMOTE平台实现功能

搭建WSN TMOTE平台主要是为了后面进行一些研究的测试。让一些理论上的研究结论在实际平台上运行，并且分析其效果。

平台主要实现了节点间相互通信以及节点与USB口的通信，并且接收其发送给电脑的数据包，构建出一个具有一定调试功能的平台，同时可以检测实际效果。其中包括了一些可编辑的模块，用以实现这些理论。不同的功能封装在不同的模块内，这样后面在实现算法以及修改的时候将更加容易。

接下来对平台作了一些简单的测试。主要用于测试节点间连接的功能。首先让两个节点相距8米，此时无法接收到数据，而在他们之间放置一个节点，可以看到原来两个不能通信的节点可以相互通信了，可以看到该网络可以实现多跳传输。接下来手动更改程序，让其传输一个数据，从接收到的包来看可以正确接收到数据。实验平台此时已经成功搭建。

# 黑洞攻击与防御

## 黑洞攻击的实现

### 基于WSN TMOTE平台的攻击实现

黑洞攻击的实现主要是修改了前面所说的路由模块，即MutiHop LQIM。这里首先说一下原来基础路由的一些机制。

通过广播一个数据为BeaconMsg类型的包来更新路由信息，其中包含了到最近Sink节点的延迟以及最近Sink节点的地址等信息。而实际上的延迟是通过接收到BeaconMsg包的频率来计算的，每接收到一个BeaconMsg包，其他节点的网络质量就会有一个衰减，该节点的网络质量会上升，这样越早收到的包对其网络质量的提升就越小，实际上就是跟接收到包的频率有关系。

黑洞攻击的实现就可以转变为提升发送BeaconMsg包的频率，这样自然而然地所有节点认为这个节点的网络质量最好，那么就会优先经过这个节点，从而达到黑洞攻击吸引流量的目的。这里比较巧妙地利用了其路由选择的机制进行了攻击，利用提高发送路由包的频率来提升该节点在其他节点路由中的优先级。

### 进行黑洞攻击后的效果

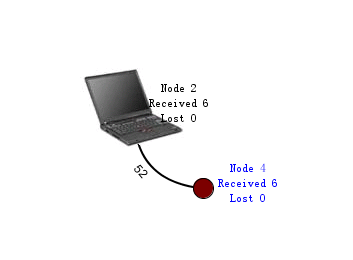
进行黑洞攻击之前的状态如图，我们可以看到节点4正常地连接到节点2，如图4.1。

图 .1 黑洞攻击之前的状态图

然后我们加入一个节点5，通过修改程序中的计时器来修改其发送路由包的频率，并且观察节点2和节点5收到的数据。

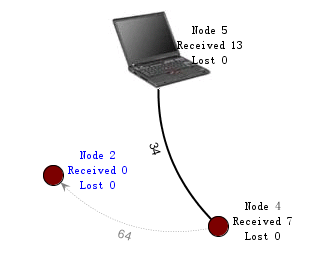
节点5即黑洞攻击节点的状态如图4.2。

图 .2 黑洞攻击后攻击节点的状态图

节点2即原正常节点的状态如图4.3

可以看到节点2已经无法正常收到数据了，数据都发送到了节点5，节点2受到了黑洞攻击。每隔5分钟监测其收到的数据，如图4.4。未收到一个包，这是一个完全丢弃包的黑洞攻击。攻击的效果非常好，将所有包拦截下来并且丢弃。

图 .3 黑洞攻击后原正常节点的状态图

## 黑洞攻击的防御算法分析

该算法占用资源与需要传输的数据和伽罗华域有关，由于WSN TMOTE平台硬件受限，若采用q>2的伽罗华域，则会占用较大的资源，使其可扩展性大大降低，很难基于它进行其他功能的开发，所以采用q=2的伽罗华域。这样实现较容易，并且也可以实现基本功能，又有利于进行后面的扩展。按照前面算法所说，只要接收到三个数据包中的两个就可以将数据还原出来，可以增强其数据的可用性。

## 算法实现

### 算法简化

首先，基于前面的分析，将使用基于上的运算，使算法实现简单。接下来，在进行封包之前为了简单确保数据的正确性，需要对原数据进行哈希算法并且将结果置于原数据末尾。但是依然由于其硬件的限制，将哈希算法简化成一个校验和，将数据按字节相加得出结果置于末尾，再将其分包并且传输，在得到数据并解码后进行简单的校验，以初步确定数据的正确性。

### 基于WSN TMOTE平台的防御实现

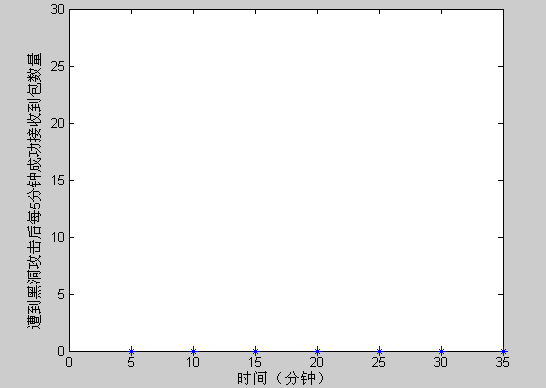
首先是普通节点的更改。将路由模块更改，将选择路由改为随机从有效列表中选择，并且整个数据发送模块的架构也发生了变化，原本是可以多个包一同发送，但是这样所有的包就会发送到同一个目标，因此改变了发包方式。由于通过ADC通道获取数据时5秒一次，那么我们在这5秒内要发送3个包，采用1秒为间隔发包。整个架构发生了较大的变化，因此这里的改动涉及的函数也比较多。需要增加一个时钟模块用以控制其将3个数据包每隔1秒发送，并且需要将原来发送多个包改为发送一个包，并且要将原来直接将三个包放入队列，而现在需要一个一个放入队列，并且更新消息池。另外算法实现上按照上述理论算法一步一步实现即可，首先计算校验和放于最后一个字节，然后将其按照4位来分割，每4位将分成三份每份4位，并且将序号放于最后一个字节，以确定其顺序用以还原数据。

图 .4 遭受黑洞攻击后接受到数据

接下来是Sink节点的更改，Sink节点此处将不作大的改动，只需要将接收到的数据发送给Uart端口，使其能够通过USB发送到计算机，以便进行后续的操作。若要还原则需要收集这些节点上至少两个数据即可还原。

## 实测防御效果及分析

经过如上改进以后，将改进过的普通节点程序烧入节点4，将Sink节点程序烧入节点2，节点5依然是黑洞攻击的节点，成功烧入后监控节点2，可以看到如图4.5。

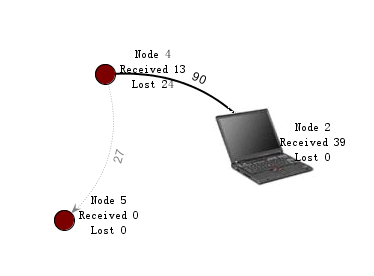


图 .5 实现防御算法后的效果图

虽然有攻击节点，但是依然能够成功建立连接并且接收到数据。能够使原来完全无法工作到现在可以接收到数据，已经进步了一大截。

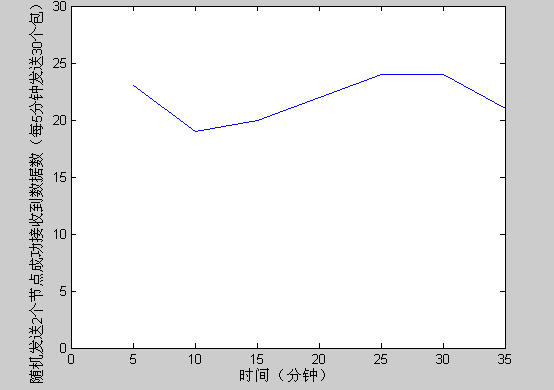


图 .6 采用随机发送2个节点效果图

首先看一下随机发送数据的结果，如图4.6所示，随机发送给2个节点时，约有75%的包能够发送给正常节点。而采用RS码后成功接收到数据也在75%左右，如图4.7。但是我们来看一下两者的通信开销。经过数据包分析采用RS码后不算包头则通信量增大了约50%，而随机发送通信量增大了100%。但是若算上包头，则随机发送只发了2个包头，而RS码方案则是3个包头。这里就需要权衡一下。如果单个数据的量较大，那么采用RS码方案将是比较好的，可以减少通信开销。而如果单个数据的量较小，那么采用随机发送的方案将比较好。

表 .1 针对黑洞攻击的两种冗余码效果比较

|  |  |  |
| --- | --- | --- |
|  | 基于GF(4)的RS码 | 重复发送2次 |
| 数据成功接收率 | 73.9% | 75.2% |
| 数据传输开销 | 增大50% | 增大100% |
| 发送数据包数 | 3 | 2 |
| 效果比较 | 发送数据较大时更好 | 发送数据较小时更好 |

实验仅仅只有1个Sink节点。当Sink节点增多时，将会有更多的包发送往别的节点。而且此处用的是基于上的运算，只能将包分为三份，若采用基于上的运算，则能将包分成更多块，允许丢失的包的数量将增多，其完整性将会更好。

## 黑洞攻击与防御小结

图 .7 采用RS码后成功接收到数据效果图

在本章中着重分析了黑洞攻击及其防御算法，成功地从原来完全无法接收到后来可以接收到数据。首先是采用了随机发送给几个质量较好的节点来减缓黑洞攻击带来的影响，但是这样只能接收到较少的包。后来提出了利用RS码进行分包并且传输，最后取得了比较好的效果，大约有百分之七十以上的数据被成功接收。

然而，回到了最初的一个问题，如果数据由于黑洞攻击被截获，截获之后被篡改以后再发回Sink节点，虽然前面有简单的校验和来验证，或者是复杂一点的哈希函数，如何更方便地验证其正确性。另外，由于数据时分布存储在各个节点上的，如何在还未获得数据的情况下就对其正确性进行一些验证，这将是下一章要讨论的问题，如何应对数据污染攻击。

# 数据污染攻击与防御

## 数据污染攻击的实现

### 基于WSN TMOTE平台上的攻击实现

数据污染攻击的实现我们有两个方案。

第一个方案，数据污染的攻击实现可以基于黑洞攻击来完成，将接收到的数据进行修改再发送到正常的Sink节点。然而由于实验的限制，无法同时观测更多的节点。所以这里将进行一些改动，提出了第二个方案，将直接将从原始节点发送的数据包进行改动，直接将其改成错误的数据包，这样的效果是一样的，并且将其发送到Sink节点，这样Sink节点将接收到错误的数据包。这就是数据污染攻击实现的原理，这实际上是模拟了传输中的误码，也就是在传输过程中某些数据出现传输错误导致最后结果的错误，由于只有简单的检错机制，所以要将数据完全还原后才能知道数据错误，或者是由于凑巧校验位未能检测出错误，这时可能就会引起严重的后果。这就是数据污染攻击的实现方法。

### 进行数据污染攻击后的效果

经过数据污染攻击后，Sink节点依然能够接收到数据，但是接收到的数据经过还原以后不再是原始数据，而是错误的数据，虽然一些数据依然能够通过校验码检测出错误，但是这样浪费了许多资源，而且要将数据还原后才能看出来。经过这种攻击后，往往效果不是很明显，但是危害却十分大。

## 数据污染攻击的防御算法分析

之前谈防御算法时说到了动态验证的机制，这个算法的主体思想是由Qian Wang提出的，然而这个算法应用在这个平台上却有问题。

最主要的问题是原算法是将数据发送到周围的节点上，而这里实现起来不太方便，由于节点容量限制，这样的话需要在所有Sink节点上和普通节点上同时实现该机制，并且需要将这些数据存储下来才能进行动态验证，而这里的节点在防御黑洞攻击时在普通节点上已经加了几个模块，可能导致空间不足。这就需要对其进行一些简化以及修改，否则可能将无法实现其基本思想。

## 算法实现

### 算法简化

首先，鉴于前面讨论的问题，我们可以对其进行一些修改，既然普通节点无法容纳更多的内容，那么我们就让Sink节点来进行这一环节，普通节点只负责进行一些预处理，发送的包中首先是经过之前防御黑洞攻击时的一些处理，得到了一些包，这些包包含这个包属于数据的序号，黑洞攻击中形成的RS码。在这基础上添加了AS签名，以及前面算法中所说的生成中的任意元素。

而现在只有Sink节点负责接收这些数据，并且将这些数据保存起来。验证数据完整性时，首先需要发送完整性验证请求，这个请求中首先需要需要包含的是序号，即你需要验证哪个数据的完整性，这是必然要存在的。其次需要一个验证的特征码，是从中随机选出的一个基数。另外还需要一个，这是需要获取的数据数量。接收到验证信息后，Sink节点将返回一个自己计算的AS签名，根据前面理论部分分析的机制，验证是否成立即可。若该式成立则验证通过，说明数据时完整的，可以被还原。

### 基于WSN TMOTE平台的防御实现

起初，我试验了在所有节点加上该机制，发现普通节点无法容下，所以才有了上面的想法，只对Sink节点实现该机制，普通节点仅作一些数据的处理上的工作。

在Sink节点上，增加了一个接收和发送数据的模块，定义了一个常量AM\_TESTPACKET，用以区分其他正常包，这个数据包就是接收和发送验证信息的包类型。由于AM\_TESTPACKET又分为两种，但是这两种包很容易区分开来。验证请求由于是通过广播的形式发送出去的，而验证请求的返回则是发送到具体一个地址。所以只要判断其目的，若是广播地址则为请求，此时将作回应，相反，若不是广播地址，则进行接收并且按照前面所述的算法进行验证。验证以后若通过则该包为完整的包，若不通过则该包不完整，将被丢弃。

这就是总的实现思路。然而具体的实现过程依然还是有许多问题的。最大的问题依然是空间问题，硬件上本身是无法保存较多包的，只能保存一小部分，那么这个数组开多大则是需要详细考虑的。另外，由于一些数据结构上的问题，还要对实际实现作一个简化，由于是以队列来保存接收到的包，然而对于队列来说丢弃一个中间包是比较困难的，而且调试起来较困难。所以此处将最后校验的过程移出程序，让其将收到的包都发送到Uart端口，另监控软件接收包并且记录下来，我们最后再通过包的分析来看结果。通过这样的简化，就可以进行调试以及分析。

## 实测防御效果及分析

首先我们发送正常数据。接收到的所有数据都是正确的，可以通过AS的检测。通过抓到的包可以验证其结果是正确的。

接下来将节点4替换为攻击的程序，将第三个包的第一位进行更改，然后将其发送，此时抓包得到其结果。

每10秒发送一个数据，按照前面理论，Node 5将发送三个数据包到三个不同的Sink节点Node 2,Node 3,Node 4。每隔5分钟记录一次并且对结果进行验证。可以得知每5分钟发送30个数据，首先取单个节点Node 2的验证结果，测试结果如图5.1。可以看到单个节点验证成功率约为50%。这实际上是因为采用该伽罗华域时，AS签名实际上只有一位，所以签名只有两种可能，故理论上单个节点只有50%的成功率。

而接下来看看综合的验证结果，三个节点的验证结果若有一个为失败则结果不正确。测试结果如图5.2。这里的成功率大大提升。达到约90%的成功率。理论上分析得知，3个节点碰撞概率约为12.5%。

这里仅仅使用了GF(4)若采用更高次的伽罗华域，碰撞概率将会非常小。比如GF(8)上的验证采用7个节点，验证成功率将超过99.99%。

关于通信的开销，可以看到该方案对于通信的开销是非常小的，取决于采用的伽罗华域，本文的实验中采用的GF(4)在通信上的开销，实测得到去掉包头是约为20个字节，包括包头则是40个字节，该值不随数据大小而变，因此是比较稳定的。若采用更高次的伽罗华域，开销将会增大，但是依然是可控范围内的。因为它不随数据位数的变化而变化。若是采用GF(8)其开销也在每个包500B以内，测试得到的结果还是比较好的。

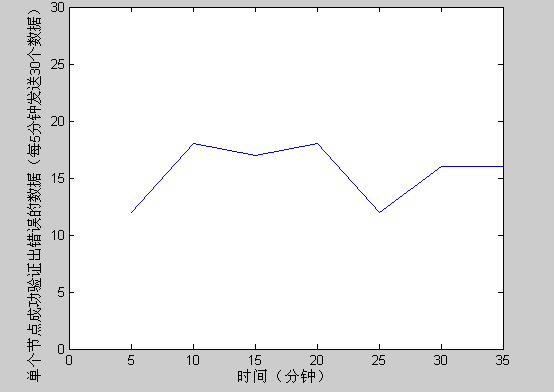
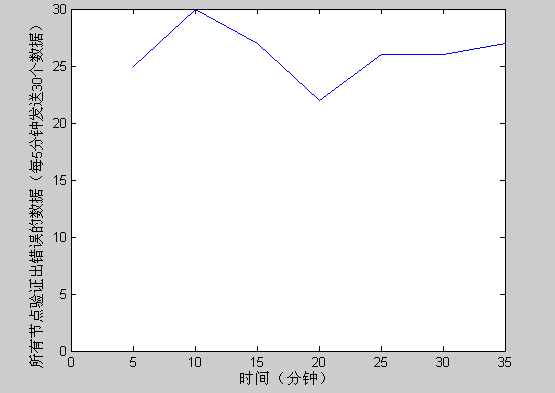


图 .2 多个节点检测数据污染攻击效果图

图 .1 单个节点检测数据污染攻击效果图

## 数据污染攻击与防御小结

本章主要介绍了数据污染攻击与防御的实现，可以成功地检测出大部分被污染的数据，以提高数据的可用性，可以减少一些误码或者是人为篡改数据造成的影响。但是，实际上这种基于AS签名的验证还是有碰撞的，前面所述的GF(4)上的算法碰撞率约为30%。尽管如此，其安全性已经得到了明显的提升。

# 总结

## 论文主要工作的总结

### 工作成果

建立了一个WSN TMOTE平台，基于该平台可以进行一些无线传感器网络的研究，并且在上面实现了黑洞攻击以及数据污染攻击，并且利用了一些防御机制一定程度地减弱了黑洞攻击和数据污染攻击的影响。

针对黑洞攻击主要采用分散发包的思想，抑制了黑洞攻击的效果，使得接收包的成功率得到了显著地提升，并且在WSN TMOTE平台上通过了测试，验证其可行性。

而对于数据污染攻击则进行了一些验证，采用AS签名进行验证，能够识别出大多数被污染的数据，以将其丢弃，保留正常的节点。在实现时将算法简化，以让其能够在平台上成功运行。最后依然有比较好的效果。

### 工作难点

首先是理论部分，理论部分涉及到一些比较深层次的数学理论，其中查阅了许多资料用以理解该理论，主要是对于伽罗华域的理解，这是一个比较抽象的内容，因此花了不少时间。

其次是关于实验平台，本平台依然有着许多缺陷亟待改进。首先是采用tinyos-1.x，无法与eclipse比较好的整合，这样就没有一个好的IDE，编写起来十分困难。并且编译一次要耗费较长的时间，经常若想要其语法通过就需要大约6-7次的编译，而编译一次需要将近10分钟。这与其编译机制有关，首先对各个模块进行编译，由于模块比较多，所以要等待很长时间才会得知具体出错是在哪个部分。编译通过以后，后面的调试就更是困难。由于其没有很好的模拟机制，只能在硬件上直接运行，通过发包来进行调试，因此修改一次要花费至少15分钟时间才能抓到足够的包进行分析，这些都是平台的一些问题。

最大的问题依然是实验平台的空间限制，曾经想将算法完整实现，但是后来发现完全不可行，其可编程容量太小，花了大量时间将其简化至可以容下，并且依然可以保持原来的效果，文中提到的几个是比较大的修改，程序中更有无数细节上的修改。

## 进一步的研究工作

首先是需要升级平台，若采用tinyOS-2.x将会与eclipse更好地结合起来，这将减少大量的工作量。

其次WSN TMOTE平台虽然容量小，但是依然可以作一些基本的研究，只是只能进行小规模的测试。但这对于一些研究来说已经足够。在这上面可以进行一些无线传感器网络协议的研究，无线传感器网络安全的研究等等。应该说还是有一定的可扩展性。通过这个平台可以创造一个新的研究思路，首先理论分析，然后通过软件模拟，再通过该平台进行小规模的测试，可以改善现在大多数研究仅仅停留在理论分析或者模拟的层面。

插图索引

[图 3.1 Delta程序结构图 10](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775836)

[图 3.2 MultiHop程序结构图 11](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775837)

[图 3.3 监控程序界面图 12](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775838)

[图 4.1 黑洞攻击之前的状态图 14](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775839)

[图 4.2 黑洞攻击后攻击节点的状态图 15](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775840)

[图 4.3 黑洞攻击后原正常节点的状态图 15](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775841)

[图 4.4 遭受黑洞攻击后接受到数据 16](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775842)

[图 4.5 实现防御算法后的效果图 17](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775843)

[图 4.6 采用随机发送2个节点效果图 18](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775844)

[图 4.7 采用RS码后成功接收到数据效果图 19](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775845)

[图 5.2 多个节点检测数据污染攻击效果图 23](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775846)

[图 5.1 单个节点检测数据污染攻击效果图 23](file:///D:\TDDOWNLOAD\综合论文训练论文写作指南\综合论文训练-无翻译.docx#_Toc265775847)

表格索引

[表 2.1 伽罗华域的三种表示形式 6](#_Toc265775848)

[表 2.2 伽罗华域GF(4)上的加法表 6](#_Toc265775849)

[表 2.3 伽罗华域GF(4)上的乘法表 6](#_Toc265775850)

[表 4.1 针对黑洞攻击的两种冗余码效果比较 18](#_Toc265775851)

参考文献

1. C. Karlof, D. Wagner, Secure routing in sensor networks: attacks and countermeasures, in: Proceedings of the 1st IEEE Workshop on Sensor Network Protocols and Applications, May 2003, pp. 1–15.
2. A.D. Wood and J.A. Stankovic, Denial of service in sensor networks, *IEEE Computer* **35** (2002), pp. 54–62.
3. C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: Proceedings of MobiCom ’00, Aug 2000, pp. 56–67.
4. F. Ye, A. Chen, S. Lu, L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks, in: Proceedings of ICCCN ’01, Oct 2001, pp. 304–309.
5. D. Braginsky, D. Estrin, Rumour routing algorithm for sensor networks, in: Proceedings of the WSNA ’02, September 2002, pp. 22–31.
6. Masateru MINAMI, Takashi MORITO, Hiroyuki MORIKAWA, Tomonori AOYAMA, Solar Biscuit: A Battery-less Wireless Sensor Network System, The 2nd International Workshop on Networked Sensing Systems, 2005.
7. Dawei Xia, Natalija Vlajic, Near-Optimal Node Clustering in Wireless sensor Networks for Environment Monitoring, 21st International Conference on Advanced Networking and Applications (AINA '07)
8. S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In IPSN ’07: Proceedings of the 6th international conference on Information processing in sensor networks, 2007.
9. J. A. Stankovic, et al, “Wireless Sensor Networks for In-Home Health J. A. Stankovic, et al, “Wireless Sensor Networks for In-Home Health J. A. Stankovic, et al, “Wireless Sensor Networks for In-Home Health 2-3, 2005.
10. Bekmezci, Ilker Alagoz, Fatih, "Energy Efficient, Delay Sensitive, Fault Tolerant Wireless Sensor Network for Military Monitoring," Sensors Applications Symposium, SAS 2008. IEEE, pp. 172-177, 2008.
11. M Hamdi, N Boudriga, M S.Obaidat, An optimized broadband sensor network for military vehicle tracking, Volume 21, Number 3, pages 277-300, 2008
12. M. Al-Shurman, S-M. Yoo, and S. Park, “Black Hole Attack in Mobile Ad Hoc Networks,” ACM Southeast Regional Conf. 2004.
13. DENG Hongmei，LI Wei，AGRAWAL D P．Routing Security in Wireless Ad Hoc Networks，[J]．IEEE Communication Magazine，2002，40(10)：70～75
14. SONG Joo—han．Secure Routing with Tamper Resistant Module for Mobile Ad Hoc Networks[EB/OL]．
15. Yang-De Ming, PanJin, Zhao Shuang , Instrusion detection method for mobile ad-hoc networks based on machine learning, Computer Applications,2005-11
16. SJ Lee, M Gerla, [AODV-BR: Backup routing in ad hoc networks](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.3261&rep=rep1&type=pdf), Proceedings of IEEE WCNC, 2000
17. K Walsh, EG Sirer ,Fighting peer-to-peer spam and decoys with object reputation, Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems
18. BD Klein, DF Rossin,Data quality in linear regression models: effect of errors in test data and errors in training data on predictive accuracy, Informing Science, 1999
19. Q. Wang, K. Ren, W. Lou, and Y. Zhang, “Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance,” Proc. of IEEE INFOCOM, 2009.
20. 彭志楠，叶丹霞，范明钰 [移动Ad hoc 网络的黑洞攻击研究](http://www.cqvip.com/qk/93231x/2009011/31896089.html)计算机应用研究, 2009
21. L.E.Dickson, Linear Groups with an Exposition of the Galois Field Theory,Teubner,Leipzig, 1901.:3-14
22. S. Lin and J. J. Costello, Error Control Coding. Englewood Cliffs, NJ: Prentice-Hall, 1983.:236-237
23. J. Dong, R. Curtmola, and C. Nita-Rotaru, “Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks,” in WiSec ’09: Proceedings of the second ACM conference on Wireless network security. NewYork, NY, USA: ACM, 2009, pp. 111–122
24. D Sannella, A Tarlecki [Toward formal development of ML programs: foundations and methodology](http://www.springerlink.com/index/P833856246K0273P.pdf) TAPSOFT'89, 1989
25. C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov, “System architecture of a wireless body area sensor network for ubiquitous health monitoring,” Journal of Mobile Multimedia, vol. 1, no. 4, pp. 307–326, 2006.
26. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An operating system for wireless sensor networks. In Ambient Intelligence. Springer-Verlag, 2004.
27. GAY, D.,LEVIS, P., VON BEHREN, R.,WELSH, M., BREWER, E., AND CULLER, D. 2003. The nesC language: A holistic approach to networked embedded systems. In Proceedings of Programming Language Design and Implementation (PLDI 2003).
28. S Nellen Eclipse Plugin for TinyOS Debugging 2009
29. Ed Callaway, Paul Gorday, and Lance Hester, Motorola Laboratories Jose A. Gutierrez and Marco Naeve, Eaton Corporation Bob Heile, Appairent Technologies Venkat Bahl, Philips Semiconductors
30. CP Lin, CK Hsieh, USB/UART converter and its control method, US Patent 6,151,653, 2000

致 谢

衷心感谢林闯教授给予我如此好的实验环境。

感谢实验室的师兄们。由衷感谢曾荣飞师兄，姚敏师兄，孟坤师兄没有你们的悉心指导，与你们的讨论使我受益良多。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：\_\_\_\_\_\_\_\_\_\_\_\_\_ 日 期：\_\_\_\_\_\_\_\_\_\_\_\_