**2017年全国大学生信息安全竞赛**

**作品报告**

|  |  |
| --- | --- |
| **作品名称：** | 基于关联环签名的安全电子投票系统 |
| **参赛学校：** | 四川大学 |
| **学院/系：** | 软件学院/软件工程 |
| **指导教师：** | 梁刚 |
| **组 长：** | 韩镓维 |
| **组 员：** | 许勤昆，凌雪，唐钰葆 |
| **通信地址：** | 四川省成都市四川大学江安校区 |
| **电 话：** | 18980406207 |
| **电子邮箱：** | 1014746390@qq.com |
| **提交日期：** |  |

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。
2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。
3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)
4. 作品报告模板里已经列的内容仅供参考，作者也可以多加内容。

目录

[摘要 3](#_Toc483487537)

[0.1背景 3](#_Toc483487538)

[0.2 现状问题 6](#_Toc483487539)

[0.3作品简介 9](#_Toc483487540)

[第一章 作品介绍 11](#_Toc483487541)

[1.1 项目背景 11](#_Toc483487542)

[1.1.1 背景综述 11](#_Toc483487543)

[1.1.2 研究动态 12](#_Toc483487544)

[1.2 需求分析 14](#_Toc483487545)

[1.3 项目介绍 14](#_Toc483487546)

[1.3.1 项目概述 14](#_Toc483487547)

[1.3.2 功能描述 17](#_Toc483487548)

[1.4 项目特色 23](#_Toc483487549)

[1.4.1项目创新 23](#_Toc483487550)

[1.5可行性分析 26](#_Toc483487551)

[第二章 实现方案 33](#_Toc483487552)

[2.1系统架构设计 33](#_Toc483487553)

[2.1.1系统架构总体设计 33](#_Toc483487554)

[2.1.2系统模块划分 34](#_Toc483487555)

[2.2系统概要设计 35](#_Toc483487556)

[2.2.1系统用例图设计 35](#_Toc483487557)

[2.2.2系统流程设计 36](#_Toc483487558)

[2.2.3网站数据处理模块设计 40](#_Toc483487559)

[2.2.4网站设计 40](#_Toc483487560)

[2.3系统详细实现 42](#_Toc483487561)

[2.3.1网站数据处理模块实现 42](#_Toc483487562)

[2.3.2网站实现 53](#_Toc483487563)

[第三章 系统测试 74](#_Toc483487564)

[3.1 测试目的 74](#_Toc483487565)

[3.2 测试环境 74](#_Toc483487566)

[3.2.1 硬件环境 74](#_Toc483487567)

[3.2.2 软件环境 74](#_Toc483487568)

[3.2.3 VS应用开发环境配置 74](#_Toc483487569)

[3.2.4 Web开发环境搭建 76](#_Toc483487570)

[3.3 功能测试 78](#_Toc483487571)

[3.3.1 公私钥对产生测试 78](#_Toc483487572)

[3.3.2 关联环签名正确签名验证测试 80](#_Toc483487573)

[3.3.3 普通RSA数字签名正确签名验证测试 86](#_Toc483487574)

[3.3.4 盲签名正确签名验证测试 89](#_Toc483487575)

[3.3.5 各阶段Socket数据传输测试 91](#_Toc483487576)

[3.3.6 Web后台管理中心测试 92](#_Toc483487577)

[第四章 创新性 99](#_Toc483487578)

[第五章 总结 102](#_Toc483487579)

**摘要**

## 0.1背景

当今时代，以信息技术为核心的新一轮科技革命正在孕育兴起，互联网日益成为创新驱动发展的先导力量，深刻改变着人们的生产生活，有力推动着社会发展，电子投票也逐渐开始取代传统的手工投票。电子选票具有电子化和网络化、低成本化、快速性和便捷性、高度匿名性等优点，能够发挥保证民主的广泛参与性。选举、投票中层出不穷的**各种亟待解决的问题**，说明了**对安全的、公正的电子选票系统的需求与日俱增**。

**【人工投票繁琐，纸张浪费】**人工投票常需要将所有的选民聚集起来，同时还会要求选民进行繁琐的填表手续。人工投票通常都是通过纸张作为媒介，造成大量的纸张浪费。并且人工投票常常都是人工进行计票，效率低且人为因素的过多，引入易造成失误和违规行为．造成较大的人力开销，投票结果统计的周期较长，时间局限。由于十分繁琐导致很多选民直接放弃投票。

**【海外工作选民难以亲临现场投票】**像总统或者主席选举，这样大规模的投票事件，选民中有很大部分选民为海外工作选民，并且很多意外状况的发生会导致部分选民无法到达投票地点进行投票，或者是错过投票截止时间，这样的意外严重影响了投票的公平性。

**【亮票影响选民投票选择】**高雄市议会2010年正，副议长选举，投票是议员疑似技术性亮票。亮票是违法行为，但是在很多纸质投票的现场，常常面临的状况是公众人物或者是政治人物公然以身试法，大做违法、违宪、反民主的亮票行为，影响选民的投票选择。

**【自然灾害导致的投票受阻】**2012年美国总统大选正式投票即将开始，因超级风暴“桑迪”受灾严重的新泽西州地区却无法进行正常的投票，因为许多投票站在风暴中损毁严重，新泽西州只好决定派遣军用卡车代替受损投票站。并且超级风暴“桑迪”离境之后，重灾区新泽西州满目疮痍，灾后重建工作正在紧锣密鼓地进行当中，对于那些投入紧张救援工作而无法抽身的人员来说，大选投票变成了负担。并且由于灾害，导致整个投票过程十分不规范，并且缺少监督，选票舞弊的现象时常发生。电子投票系统便可以很好解决由于灾难而导致的投票受阻以及选票舞弊的现象。

**【可能发生恐怖袭击事件】**2012年佛州投票站外引爆可疑包裹，选民们耐心排队等候投出自己神圣的一票时，眼尖的民众发现了两个可疑的包裹，分别是一个插有U盘的散热器和一个黑色的垃圾袋。奥兰治县防爆小组接到举报后火速赶往现场。经过一番分析调查，为了保险起见，防爆小组决定疏散民众、引爆这两个可疑包裹。而投票站的正常运行也因此中断了几个小时。选票站选票，或者是选民集中投票，可能会引来极端分子的恐怖袭击事件，导致选票作废的事件发生。

**综上可知，电子选票系统已经变得必不可少。**

电子投票系统在如今已经变得不可或缺，我们知道传统的投票方式需要去指定地点投票并且采用人工计票，效率低且人为因素的过多引入易造成失误和违规行为．与传统人工投票相比，电子投票可以节约大量的人力物力，且具有公平、高效等明显优点，但电子投票的虚拟化带来便利的同时也带来了**许多安全问题**。

**【数据库错误导致选票作废】**2000年的总统大选，从选民登记到最后的选票统计方式都出现了误差。据估计，当年的选举中，大约400万到600万的选票没有纳入统计结果或投进票箱，占整个选民的2%以上。布什最后以微弱得票优势当选为总统，被认为是极不光彩的。首先，选民登记数据库出现错误，使150万至300万选民没有登记上，仅佛罗里达州，就有8万登记选民因计算机算法程序问题而被电脑删除。另外，由于投票系统的设计故障，大约150万至200万选民的选票失效。最后还有100万已登记的选民抱怨因投票点队伍太长，使得他们在选举结束时还没有来得及投票，从而影响了最后结果。

**【电子投票凭空捏造选票】**电子计票机似乎还能凭空捏造出选票：2003年，在一场艾奥瓦州布恩县举行的地方性选举活动中，注册投票选民仅有19000名，然而电子计票机汇总出的选票却有144000张。

**【电子投票操作复杂导致投票作假】**2007年在肯塔基州举行的一场地方选举活动，一个法官、一个书记员、一个竞选活动主管以及一群其他身份的被告人供认他们勾结在一起操控了6个地方职位的选举结果，他们使用的手段就包括篡改电子计票机的数据。根据起诉书列举的证据，这帮阴谋家故意在介绍使用电子计票机的方法时误导选民。他们告诉选民，只需按下电子计票机上标有“投票”的按键，投票结果就被记录下来了。事实上，还有一个步骤——让选民查看选票的当前内容，并且可以自愿修改。最终，当选民按下标有“确定”的按键后，选票内容才最终正式确定。选民按下标有“投票”的按键就离开了那里，而这帮阴谋家则偷偷溜到电子计票机前面，更改选票内容并按下标有“确定”的按键，让电子计票机认为是选民自己“更正”了选票内容。

**【通过植入微处理器便能操纵选票】**2012年奥巴马和罗姆尼的选举开始之后，一位名叫罗杰·约翰斯顿的工程师表示，美国现有的电子投票系统其实有很大漏洞，隐患重重，只要一点基础知识就能操纵选举。罗杰·约翰斯顿是阿尔贡国家实验室的漏洞评估团队负责人。他对美国现有的电子投票机终端进行了模拟攻击实验，结果是美国现有的电子投票机确实有漏洞，而且机器在全美范围内都可能存在。Diebold Accuvote-TSX触摸屏投票终端被遍布20个州的2千6百万选民使用，而4个州的9百万选民会用到Sequoia AVC投票终端。操作不需要太复杂，不涉及任何太过专业的领域，只要在投票机中植入一个微处理器，就能随心所欲操纵选票。原因很简单——大多数投票机使用开放标准的通信模式，没有加密。

**【更改投票结果，缺少凭证无法证明投票作假】**2015年9月初，长春市一个微信公众号“家长掌上课堂”发布了一个为国足征集小球童活动，王女士给自己的孩子报了名，并发动亲友团拉票，眼看着孩子的票数到了690票，名列前茅，她才放心。可是第二日上午，她突然发现自己孩子的投票只剩9票了！仔细一看，还有一些孩子的票数也明显由百位数减少到十位数，然而个别孩子的票数却没减少。有关专家认为，该投票系统被黑客篡改了票数，但是由于缺少凭证无法证明投票被篡改。

**【别国干涉本国选举来干涉政治】**2016年12月16日，美国联邦调查局（FBI）局长James B. Comey 和国家情报局（National Intelligence）局长达成一致，认同了中央情报局（CIA）关于俄罗斯干预了美国2016年总统大选投票，并帮助特朗普成功竞选的报告。美国中央情报局一份秘密评估报告指出，2016年美国总统选举期间，美国两个州的投票数据库遭到黑客攻击，窃取了20万选民的个人数据，篡改选举结果。

**由上可知**，电子投票系统的不安全现象不仅发生在新闻上，也开始出现在我们的日常生活中。黑客操纵美国总统大选的风波虽然已经渐渐平息，却对其他国家的选举产生了巨大的影响，各国政府猛然意识到：网络安全不仅存在于虚拟世界，还能影响一个国家的民主和政治发展，甚至改变世界格局。为了避免任何网络攻击的干扰，荷兰政府干脆一不做二不休，决定将以后选举的所有选票由手工来计算。**因此设计与实现安全的电子选票系统势在必行。**并且一个好的电子投票方案在安全性与实用性方面应该满足以下需求：

1. 匿名性：除了投票者，没有人能够确切地了解选票的内容。
2. 民主性：仅有合法的投票者可以投票，并且仅能投一票。
3. 正确性：所有的选票正确的计数，所有的选票不能被修改、删除或者替换，不能增加非法的选票。
4. 可验证性：计数过程必须接受任何个人、团体的检验。
5. 不可强迫性：投票者不能被强迫或收买，投票者不能向第三方证明自己确切的投票内容(以防贿选)。
6. 稳健性：任何小的团体不能破坏并中断选举。

## 0.2 现状问题

第一个现代意义上的电子投票方案，是由Chaum于1981年提出的，它采用了公钥密码体制，利用数字签名来隐藏投票人的身份，并通过计算机和网络来完成整个过程。1985年，Cohen和Fisher提出了基于同态加密技术的电子投票方案，接着Benaloh,Yung,Iverson,Sako和Kilian等也分别提出了基于同态加密技术的电子投票方案。

近年来，在电子投票方案不断发展的过程中，有些方案过于复杂，不适合大型投票，而有些则是在安全方面存在着较大的漏洞。第一个使用的适合大规模投票的方案，是由Fujioka，Okamoto和Ohta在1992年提出的FOO方案，方案的核心采用了比特承诺技术和盲签名技术。该方案提出后，受到了社会的较大关注，被认为是一个能较好实现安全投票的电子投票方案。至此，电子投票系统适用性方案得到了实质的突破，并在各个非政府部门得到了广泛的应用。之后，该方案不断由许多大学和研究机构进行改进，开发出了相应的电子投票软件系统。其中注明的有MIT的EVOX系统和华盛顿大学的Sensu系统。1999年，台湾大学的Wei-Chiku和Wang-ShengDe提出了一种基于RSA的电子投票方案。

在电子投票系统的研究和发展过程中，研究者们又设计了适合不同场合目的的电子投票系统。

为了解决电子投票方案中的问题，Cramer利用门限同态加密、电子公告牌和零知识指数证明等技术给出了一个效率较高且满足匿名性和广义验证性的投票方案。随后，为了解决投票过程中出现的买卖选票、强迫投票的行为，Benaloh引入了电子投票的“无收据性”的概念，即投票者无法向第三方显示其投票给了哪一位选票人。在此基础上，Martin证明了多投票实体模式下，Benaloh的方案不具备无收据性。Lee在Cramer方案基础上进行改进，引入了一个“可信任的第三方”作为验证，以此实现无收据性，但在实际投票过程中，“可信任的第三方”是难以达到的条件。

在2000年美国的总统选举中，佛罗里达州等少数几个地方对部分选民试用了通过Internet进行投票的选举，这对电子投票系统发展有着重要的意义。

2002年，法国的总统及国会议员选举中使用E-POLL系统进行电子投票。 2004年，美国总统大选，电子投票系统首次启用，标志着无纸化的电子投票选举时代的开端。Diebold公司的电子投票系统被安置在全美75000个投票点用于投票，该公司也因此成为了全国最大的选举工具提供商。

2007年，法国大选的首轮投票，采用了电子投票的方式。约有150万选民以电子投票的方式进行了首轮投票。

虽然，在线投票系统也越来越得到大家的欢迎，并且逐渐成为投票领域的主要发展趋势。在线电子投票系统目前是很多网站采用的网络在线调查用户信息的一种方式。这种方式的优点是反馈信息迅速，统计分析速度快，系统能自动并快速地计算投票的结果并把结果立刻反馈给用户。

然而通过分析，目前国内，许多电子投票系统**仍然存在着以下问题：**

1. **投票人身份的匿名性仍存在缺陷**

陈晓峰等人[[1]](#endnote-0)提出的电子投票方案中，如果注册机构与管理机构或计票机构合谋，就能追踪到投票人的所有信息。赖瑾等人[[2]](#endnote-1)提出的电子投票方案中，如果注册机构与记票中心合谋就可以知道某张选票所对应的投票人，从而泄露投票人身份等相关信息。

1. **无收据性和公开可校验性的矛盾**

即如何既防止选票买卖和选举胁迫，又使选民可以及时发现选举舞弊，并匿名地证明正确的选票结果。目前的方案多数依赖于某些物理假设，如存在选举亭、安全信道、智能卡等，但这样做无疑降低了方案的可行性、灵活性和易用性。另一种方式是通过引入冗余，利用可否认加密、指定校验者重加密证明等技术，使得接收者收到的信息模棱两可。但是如果选举协议同时要求允许写入式的选票形式，将使得该方案的实现难度大为增加。

1. **电子投票的可验证性和秘密性存在矛盾**

一方面，投票系统需要向公众证明所有的选票都已被正确计数。另一方面，这些证据不能损害投票者的匿名性。D. Chaum[[3]](#endnote-2)和K.Ohta[[4]](#endnote-3)利用匿名通讯信道分别给出了一个适合于大群体选举的投票方案，保证了投票者的匿名性，然而这两个方案都没有解决选票的秘密性和公平性。

1. **电子投票系统稳健性不佳**

纸质的选票比数字存储的选票更难造假，人工验票可以减少人们对不公平选举的担忧。在投票机上投票，如果提交前将已选好的选票打印出来，那也是有据可循的。但采用在线电子投票系统而没有任何纸质凭证的投票机的话，一旦面临作弊指控，就很难进行核查了。

1. **基于不同密码技术的投票方案互有利弊**

现有的投票方案，大体上可以被分为3类：基于 Mix．Net的、基于同态加密的和基于盲签名的。不同密码技术支持的投票方案及其延伸方案各有优缺点，比如选票是否存在碰撞，是否满足安全需求，是否存在第三方安全隐患，投票效率高不高等。如何权衡也是一大问题。

## 0.3作品简介

根据以上分析，本项目结合现有安全电子投票理论，实现了一个基于关联环签名的高效安全电子投票系统，旨在排除电子投票中的安全隐患。

电子投票系统以网站的形式搭建。在符合电子投票协议应具备的基本安全要求的前提下，也充分考虑了对现状问题进行改善。系统做到：

1. 服务一场正常的电子投票
2. 无条件匿名性，完成匿名不依赖任何第三方和匿名通信信道
3. 投票各阶段可公开验证
4. 解决电子投票可验证性和秘密性之间的矛盾
5. 避免选票碰撞，任意阶段可弃票，投票过程效率提升

为实现以上功能，系统中使用关联环签名实现投票者的匿名投票，使用盲签名解决投票者与投票中心相互猜疑，使用普通数字签名预防各阶段传递的信息被篡改。系统使用身份序列码解决了选票碰撞问题，同时，通过在投票过程中引入可公开验证的信息公告机制允许投票者在任意阶段中途弃权，并防止了第三方机构的不诚实行为对投票过程的破坏及对选票结果的干扰。此外，将计算量较大的关联环签名用于注册阶段，在后续的投票和计票阶段则利用基于身份序列码的个体签名进行身份验证，无需任何匿名通信信道，有效提高了投票协议效率，适合于大规模选举。

**作品关键字**：关联环签名 盲签名 身份序列码 公开验证 无条件匿名性

**第一章 作品介绍**

**1.1 项目背景**

**1.1.1 背景综述**

电子投票是通过使用计算机与网络通信，利用高速计算的计算机负责整个投票过程，通过互联网能够将身在不同地方的投票人联系起来，在保证合法的投票人身份的匿名性的前提下，每个投票人都能够根据自己的主观意愿投出选票。同时与传统投票相比，电子投票能够避免在各个环节中人工因素对最后投票的真实有效性进行干扰，并且通过计算机统计，可以大大提高最后对选票结果统计的速度，从而能够极大的节约人力物力以及时间，使得每场电子投票的结果能够更加的公正公开，普遍为投票人所相信。

自从电子投票产生以来，在美国等国家已经开始将电子投票全部或部分应用在政府日常选举投票中。比利时，巴西，美国，爱沙尼亚，菲律宾，印度和委内瑞拉，这七个国家是全球拥有电子投票系统的国家。 2004 年美国大选中，美国联邦政府吸取上届大选的失败经验，在全国各地设立电子投票站，使用电子选票的形式完成了整场总统投票，并沿用至今。欧洲的爱沙尼亚在2015年的国会投票中，30%的投票人通过在线方式进行了投票。在过去的十几年间，人们不断对电子投票方案进行完善，并且得到的最新研究成果使得投票方式发了翻天覆地的变化。电子投票的初衷在于，能够得到更加公正的结果从而从吸引投票人参与到日常投票中，得到更为令人信服的结果，同时科技的发展也大大提高了投票和投票效率。对这类投票方案的研究有助于我国不断填补完善我国在电子投票领域的空白，帮助我国公民更好的了解电子投票，对以后电子投票的推广有着积极的推动作用。举行一场安全无争议的电子投票，需要安全的网络环境，以及安全性通过证明的加密算法。对电子投票研究的同时，可以同时带动通信网络、数学数论、密码学等多个研究方向的发展。

**[1.1.2 研究动态](#_Toc358831685)**

**1.1.2.1 电子投票研究现状**

电子投票是通过使用计算机的高速计算，结合密码学中的加密解密方案来实现的。Chaum在1981年首次提出相对意义上的电子投票是由，他提到的mix-net协议来实现匿名信道传递邮件。该协议首先通过使用混合服务器对邮件加密后的密文进行混淆置换操作，使用秘钥对相应层进行解密操作，由混淆服务器输出最后的明文。这样使得攻击者无法通过追踪密文的来源，来确定发送者的身份，从而将发送者隐藏。但是在该方案中也存在一定的问题，使用RSA公钥体制对邮件进行加密时，加密消息的长度和用户计算量与mix-net的服务器个数存在正比关系，所以该方案需要的存储空间会随投票人的数量增加而成倍数的增加，故不适用于大规模的电子投票。在电子投票方案发展的历史中，其中最为重要的研究成果，是由Fujioka等人在1992年提出的，在该研究成果中使用RSA作为加密算法，根据研究者的名字将该研究成果命名为FOO方案。由于该方案使用的RSA作为加密算法，所以需要对选票进行的运算量较小，选票作为数据交换过程中的数据包较小，所占空间较小，是一个比较成功的可以在大范围内中应用的电子投票方案。之后在1996年，Cranor在文献中提出了一个安全的电子投票必须满足的7个特性，并且被沿用至今，作为每个电子投票方案都要基于这7个特性来论证方案的安全性。同年，Cramer首次在文献中提出了多候选人的问题，拓展了电子投票只能单一的对一位候选人表达赞同或反对意见。之后在1997年，Cramer等人在文献中提出了基于加密算法同态性的电子投票方案，该方案使用ElGamal加密算法来构造选票，该方案在计算复杂度上，对计算量的要求很高，并且只能实现投票人在多个候选人中选择一个作为自己的选票。但是在原先的电子投票方案中，基于同态加密算法构造的电子投票方案改变了由于加密算法的限制，电子投票仅仅能实现“二选一”的狭窄局面。在同态加密提出后，极大的丰富了投票的形式，可以实现“多选一”的投票方式，但是同时还面临，加密算法过于复杂，计算难度大，以及无法杜绝某个投票人对同一个候选人多次投票等问题。同时上述的电子投票方案也存在，使用中央领导的方式，使得参与电子投票的某一实体拥有绝对的领导权，一旦中央领导实体滥用权力，会导致内部欺诈，进一步影响电子投票方案的公正可靠性。

**1.1.2.2 电子投票不同的实现方案**

自电子投票方案于1981年由Chaum提出至今，现有的电子投票方案按照实现方法不同可以分为三类：基于混合网络（mix-net）协议的电子投票方案、基于盲签名的电子投票方案和基于同态加密技术的电子投票方案。混合网络从理论上可以实现解密计票的公开可验证性，但因为其算法的复杂性，当选举规模扩大时，需要进行大量的计算以满足零知识证明的要求，方案计算效率低下。基于盲签名的方案需要预设较强的假设，如匿名信道和可信签名方等。该方案拥有较好的计算效率，所以市面上已经实现的电子投票系统以盲签名方案为主。盲签名方案的缺点是要求投票的组织者、监督者、计票人完全可信，对投票参与者的安全要求性高；此外，盲签名方案中只有投票者可以验证自己的选票是否被计入，无法满足公开可验证的安全需求。相较而言，同态加密技术可以对密文本身进行任意计算，适合解决电子投票应用中核心的安全与信任问题。文献介绍了基于ElGamal的电子投票方案，文献提出了基于Paillier的电子投票方案。ElGamal和Paillier都是以幂计算实现同态的功能，不仅算法效率低下，而且选票数据量的规模也会呈指数增长，只能做有限次的加法运算。直到2009年，Gentry基于理想格构造了第一个全同态加密方案，掀起了全同态加密方案的研究热潮。在此基础上，2013年Gentry和Halevi发布了同态加密开源库——HElib。HElib在BGV同态加密方案的基础上，加入了密文更新、自举电路等功能，满足了全同态的加密要求。针对投票应用的特殊性，本文提出了一个基于全同态加密技术的电子投票方案。同时，在HElib的编程接口上，设计了一个高效的同态密文加法器，在该加法器的基础上实现了该投票系统。与盲签名方案相比，此系统即使将计票过程可以交给不可信的第三方进行，也能满足了选票的匿名性、机密性和公开可验证性的安全要求。

## 1.2 需求分析

根据以上分析可知，现有方案存在的选票碰撞、投票者无法中途弃权、投票效率低下，以及由第三方机构不诚实行为导致的安全隐患等问题。因为安全电子投票系统的受众注重投票各阶段的安全，为保证用户的积极使用，我们必须考虑以下几点问题：

1. 电子投票网站在安全、公正的前提下，要延承电子投票快速便捷的优点
2. 投票人完成匿名不依赖任何第三方和匿名通信信道，即无条件匿名
3. 网站能作为投票中心和投票人之间的中枢支持两方互相验证信息
4. 现存的理论化电子投票方案大多无法有效地应用于实际投票，为解决这一问题，需要避免选票碰撞，允许中途弃票，提高投票效率

## 1.3 项目介绍

**1.3.1 项目概述**

本系统提出并实现了一个基于关联环签名的高效安全电子投票方案。使用身份序列码解决了选票碰撞问题，同时，通过在投票过程中引入可公开验证的信息公告机制，允许投票者在任意阶段中途弃权，并防止了第三方机构的不诚实行为对投票过程的破坏及对选票结果的干扰。此外，将计算量较大的关联环签名用于注册阶段，在后续的投票和计票阶段则利用基于身份序列码的个体签名进行身份验证，无需任何匿名通信信道，有效提高了投票协议效率，适合于大规模选举。

本项目提出了基于关联环签名的安全电子投票系统，具有以下特点：

1. **投票人的匿名性**

在投票之前，投票者利用关联环签名进行匿名注册，投票中心只能确认投票者为合法投票者之一，但无法确定投票者真实身份。此匿名方案不依赖任何第三方和匿名通信信道，实现了无条件匿名，很大程度上降低了投票者身份被识破的可能。

在选举的任意阶段都采用独立的信息公告机制，且分别设置了公示时间，供投票者查询自己的信息是否得到正确公布。同时，在所有人的监督下能够及时发现并有效制止第三方机构的不诚实行为。

对投票者进行无条件匿名注册，同时为投票者发放具有唯一性的合法临时身份用于后续的投票及计票过程，任何人只能确定投票者的合法性而无法得知其真实身份，因此无需借助匿名通道。

1. **投票的保密性**

只有投票人自己知道选票的内容，除计票机构外任何机构都无法知晓内容。在投票阶段，结合投票者被分配的用来唯一标识身份的序列码Ni，投票者生成电子选票，随机选择一盲化因子，用盲化算法盲化选票，投票者再对盲化后的选票进行签名。把｛序列码Ni，盲化后的选票，选票的签名｝发送给投票中心，投票中心接收及公布的选票均是盲化后的选票，即使在投票中心中找到选票，但也无法知晓选票具体内容。

因此本系统具有很好的保密性。

1. **公平可验证性**

每个过程都接受监督检验，任何不诚实行为也可以被制止。

从投票者角度，任何不诚实的投票行为会被发现和制止。

首先，保证了投票人的身份合法具有注册资格，进而获得注册机构为其分配的唯一标识身份的序列码Ni。其次，在其后的投票和计票环节中，都使用Ni，且Ni分别公布在List2，List3中，如投票者重复投票，则会在表中出现冗余，从而杜绝投票者重复投票行为。

从整个机构角度，涉及到投票者信息的投票中心会将信息公布到对应的公告信息表中。

在注册阶段，投票中心将 ｛消息，对消息的关联环签名，对消息的普通个体签名，Ni｝保存到注册信息表List1中，公布List1，接受公开验证；在注册公示时间内，未能在注册期及时注册的投票者可以公开注册，公布自己的签名，要求注册中心为其发放身份序列码和对应签名，并公布其注册信息于List1中；同时已注册者查询List1，如果查询不到自己的注册信息，投票者可以公示自己的签名，并公开验证签名。对消息的关联环签名和私钥对消息进行的普通个体签名的正确性，进而要求投票中心在所有人的监督下为其发放身份序列码和对应签名，并公布投票者相应信息于List1上。能够有效防止投票中心的不诚实行为。

在投票阶段，把{选票，盲化选票，盲签名}保存到投票信息表List2中，投票结束后公布List2，接受公开验证。在投票公示时间内，未能在投票期及时投票的投票者可以公开投票，公布自己的盲化选票和签名，要求投票中心为其发送签名并将其投票结果公布于List2中；同时已投票者查询List2，如果查询不到自己的投票信息，投票者可以公布自己的序列码、盲化选票和签名，在所有人的监督下，要求投票中心为其签名并发送选票、选票的签名，并将其投票结果公布到List2中。能够有效防止数据管理机构的不诚实行为。

在计票阶段，投票人将消盲后的选票公布于计票信息表List3中，能够有效防止计票机构的不诚实行为。

因此，每个过程都将接受大家的监督，防止第三方泄露信息。

1. **可在任意阶段弃票**

投票者在任意阶段弃权，体现在各公告信息表中的投票者数目不相等，并不会影响选举的正常进行。

在注册阶段，用户部分信息会公布在于List1上。若公示时间结束后，投票者依然没有注册，则被视为自动弃权。

在投票阶段，投票信息表List2中会公布序列号、选票、选票的签名。投票结束后公布List2，接受公开验证。如果公示期结束，投票者依然没有投票，则被视为中途弃权。

在计票阶段，投票人将消盲后的选票公布于计票信息表List3中。超过计票期限，投票者依然未将选票公布于List3中，则被视为中途弃权。

1. **高效性**

环签名的大计算量成为制约其应用于大规模电子投票的主要因素。与现有方案不同的是，本产品将计算量大的关联环签名应用于注册阶段实现合法投票者的匿名注册，并通过为合法投票者发放一个唯一的临时身份标识Ni，并结合自己产生的公私钥对，用于后续投票和计票阶段的匿名身份认证。由于本产品在投票阶段仅使用公私钥对实现普通的个体签名及验证，故有效保证了投票阶段的高效性和实用性，适合于大规模选举。

**1.3.2 功能描述**

本系统提供了如下功能：发起投票，匿名投票，查看结果，公开验证。

| ***功能*** | ***详细描述*** |
| --- | --- |
| 发起投票 | 投票中心在网站上发起投票，填写投票主题等相关信息，之后添加投票者邮件地址，网站对每位投票者发起邀请。 |
| 匿名投票 | 在投票之前，投票者接受到邮件邀请，之后利用投票者消息进行匿名注册，投票中心只能确认其为合法投票者之一，但无法确定投票者真实身份。  确认身份后投票者获得唯一标识身份的序列码，进入投票页面进行投票。 |
| 查看结果 | 投票结束后，投票者和投票中心可以前往网站投票结果页面查看本次投票结果。 |
| 公开验证 | 在选举的中间过程，选举的中间结果不会被泄漏，且投票的状态变化公布在网站的公告信息表中，投票人和投票中心可凭借手中持有信息对信息表进行公开验证，对各阶段进行有效监督。 |

表1-3-2-1：系统具体功能列表

**1.3.2.1发起投票**

投票中心登录网站创建投票信息，输入主题、内容等信息创建投票。

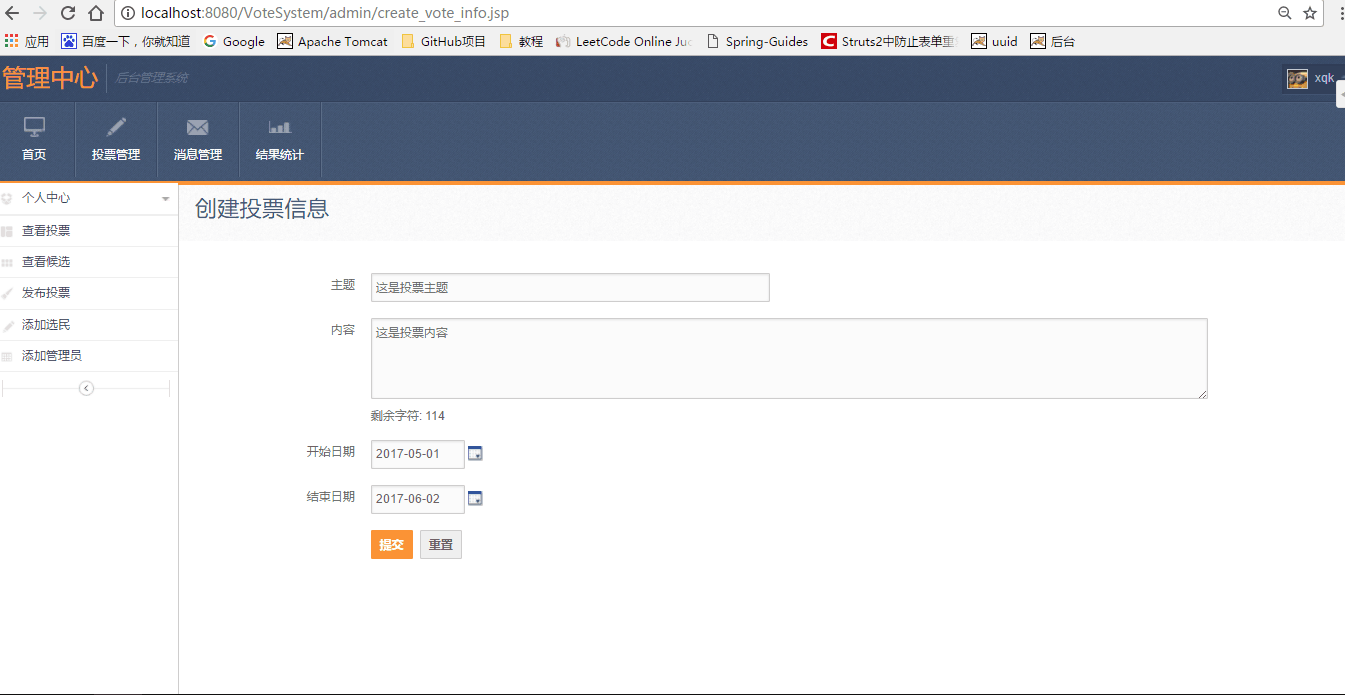


图1-3-2-1-1 创建投票

投票中心在网站上编辑邮件内容，网站根据投票中心提供的邮件地址对每位投票者进行邀请。

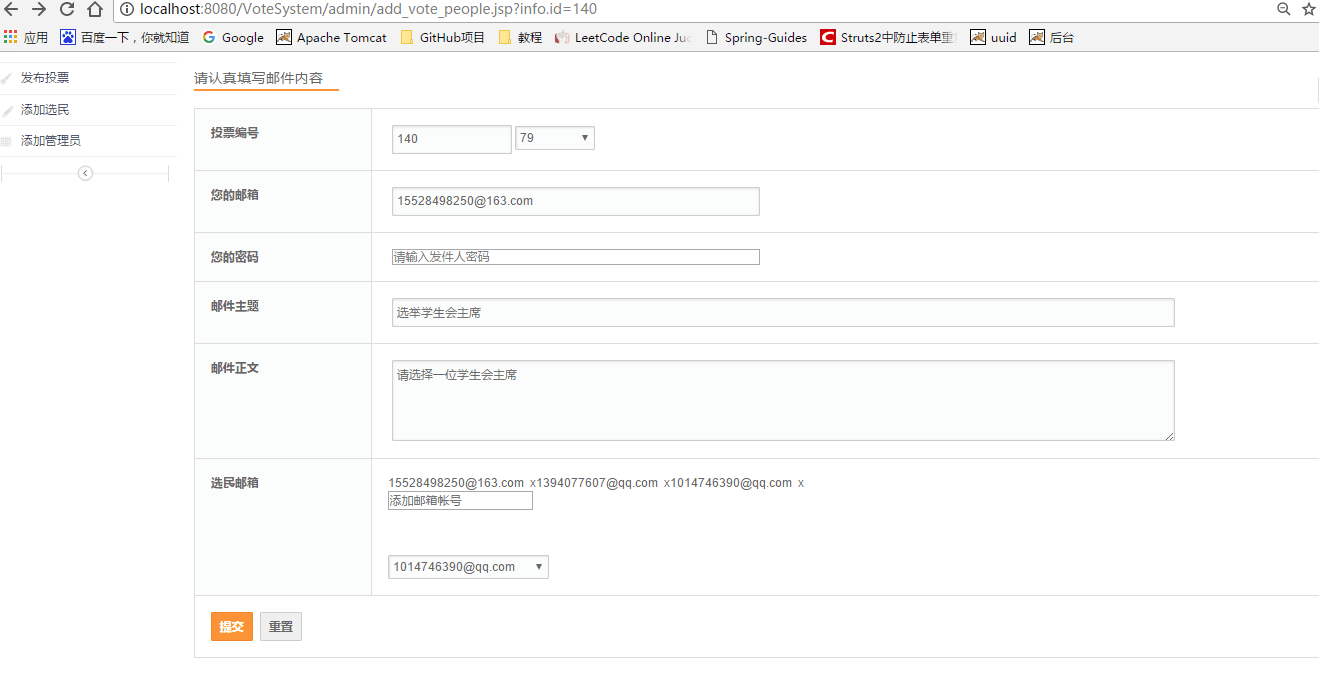


图1-3-2-1-2 编辑邮件

**1.3.2.2匿名投票**

每位投票者通过邀请，通过网站生成关联环签名，投票中心可以通过环签名确认投票者的身份合法性，并分配给投票者用于标识唯一身份的序列码，用户完成匿名注册。

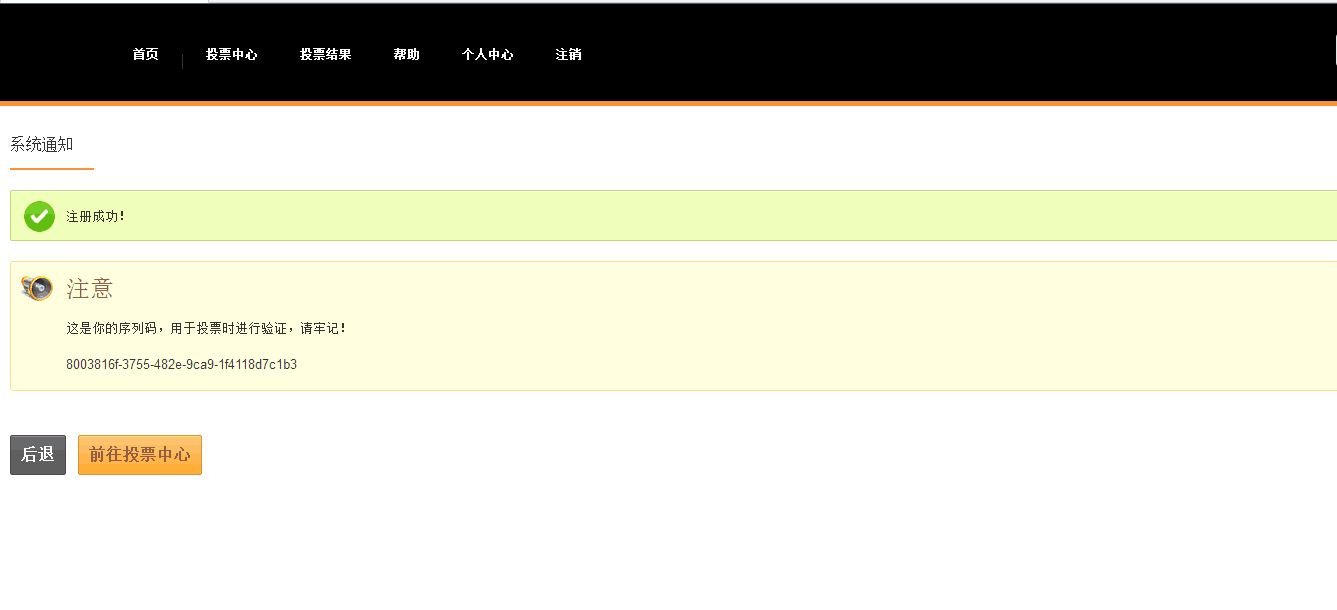


图1-3-2-2-1 匿名注册

投票者凭借身份序列码进行匿名投票。

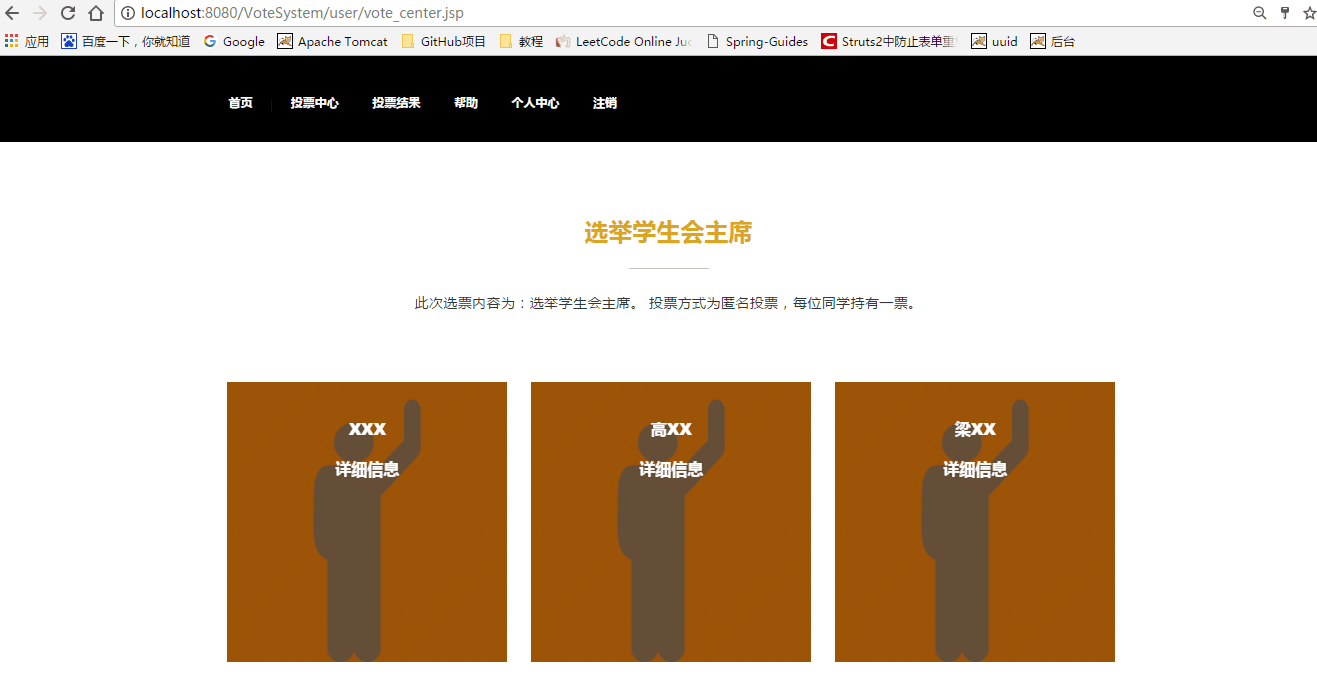
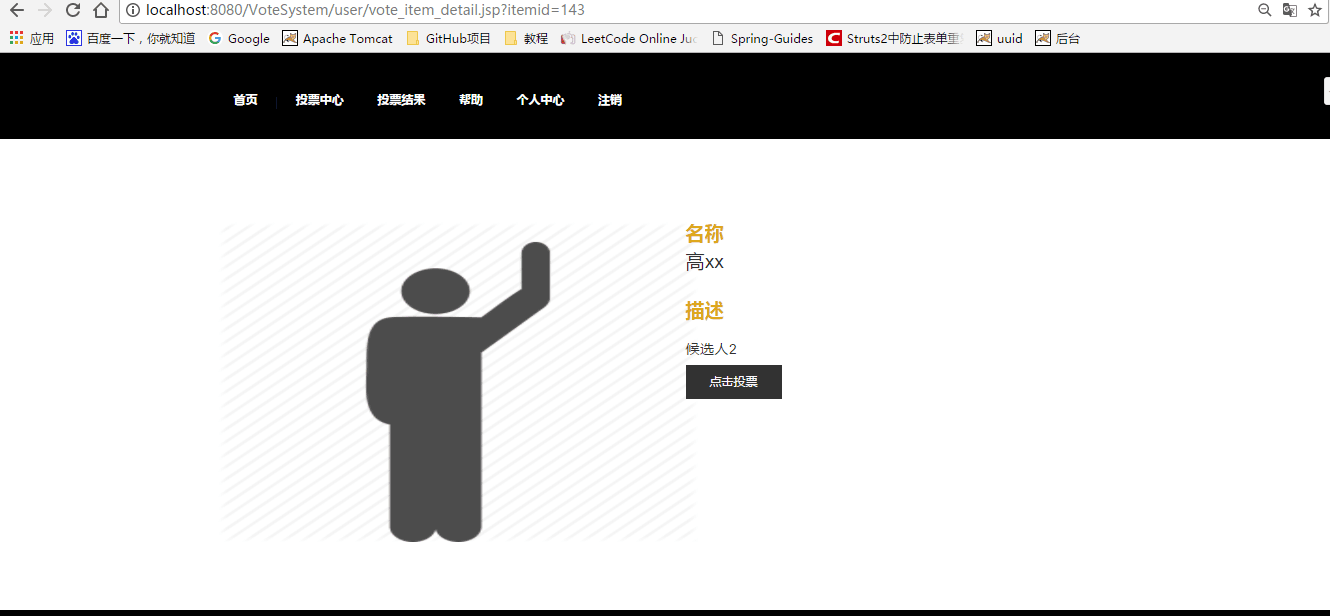


图1-3-2-2-2 选举内容显示

图1-3-2-2-3 匿名投票

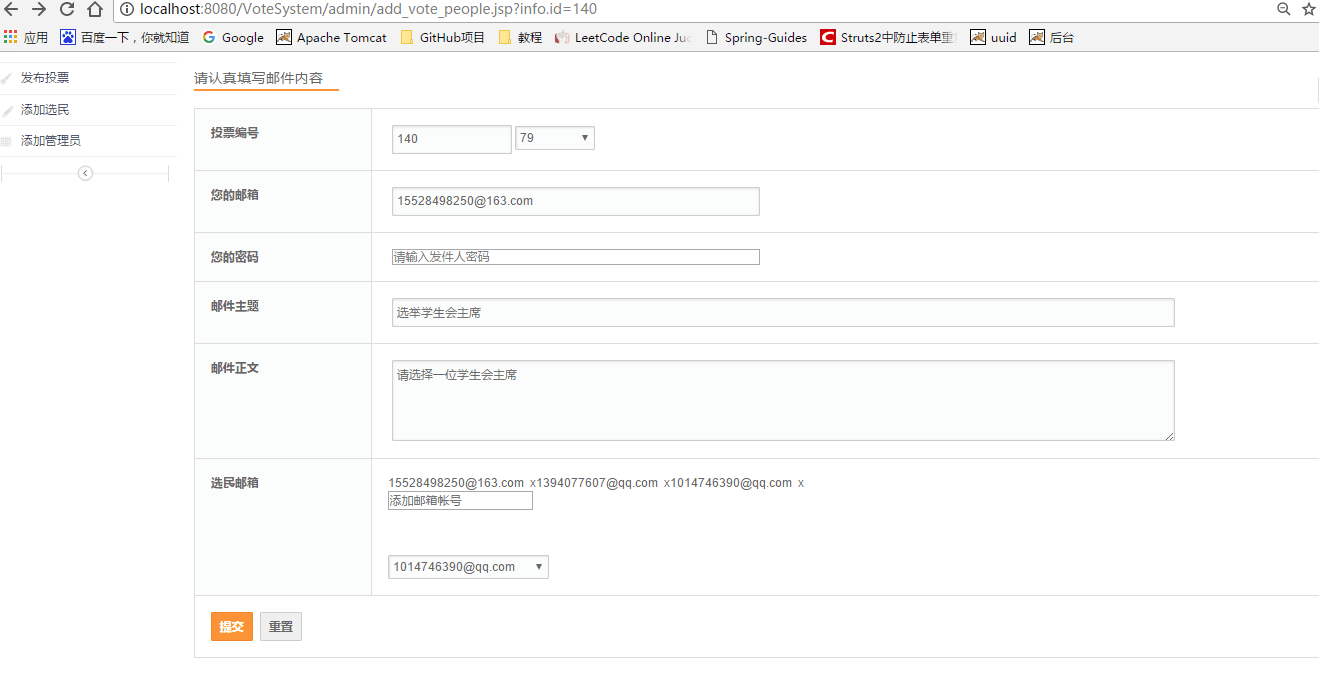


图1-3-2-2-4 编辑邮件

**1.3.2.3查看结果**

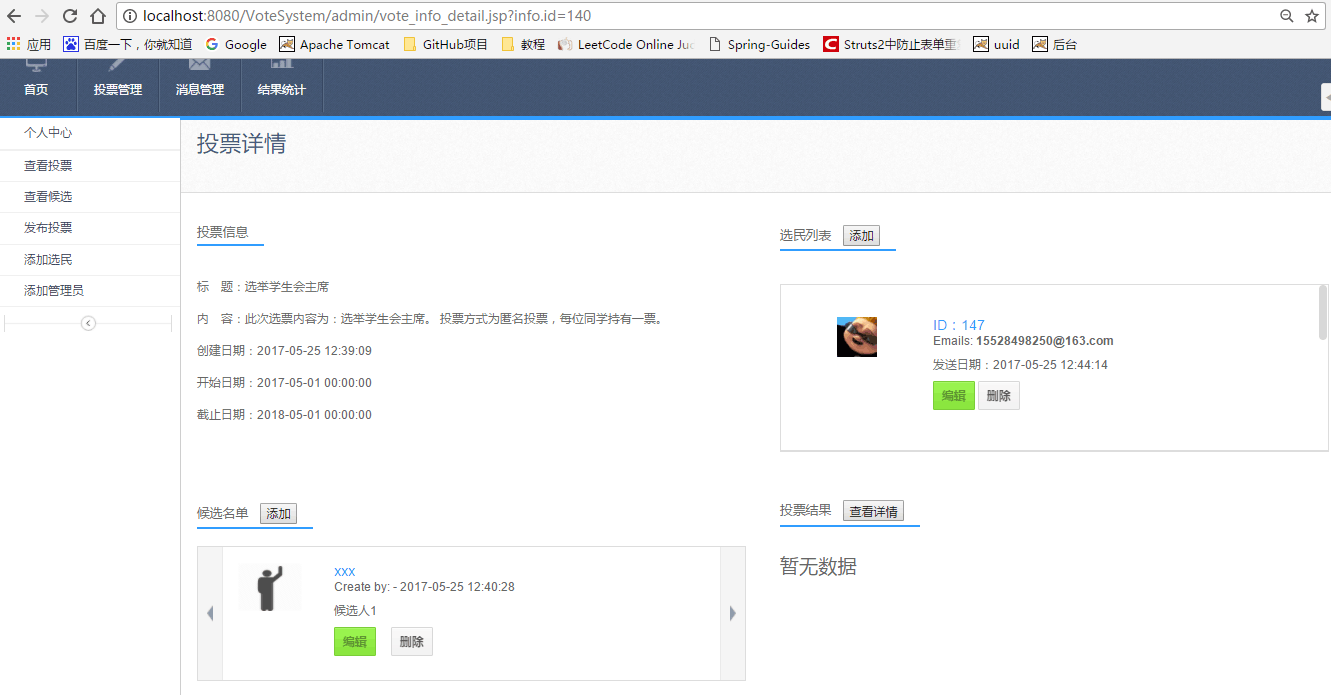
系统包括三个阶段：注册阶段，投票阶段，计票阶段。三个阶段限定时间完成，投票结束前不显示投票结果。

图1-3-2-3-1 投票结束前

投票结束后，投票者和投票中心可以前往网站投票结果页面查看本次投票结果。

**1.3.2.4公开验证**

在选举的中间过程，选举的中间结果不会被泄漏，且投票的状态变化公布在网站的公告信息表中，注册阶段结束公布List1，投票阶段结束公布List2，计票阶段结束公布List3。如下列图示。

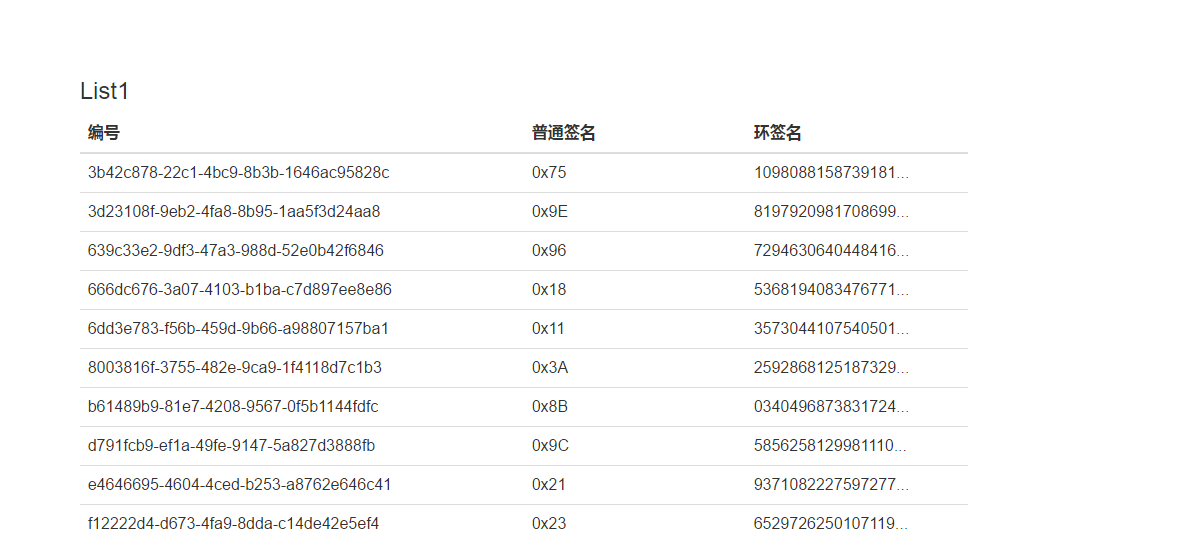
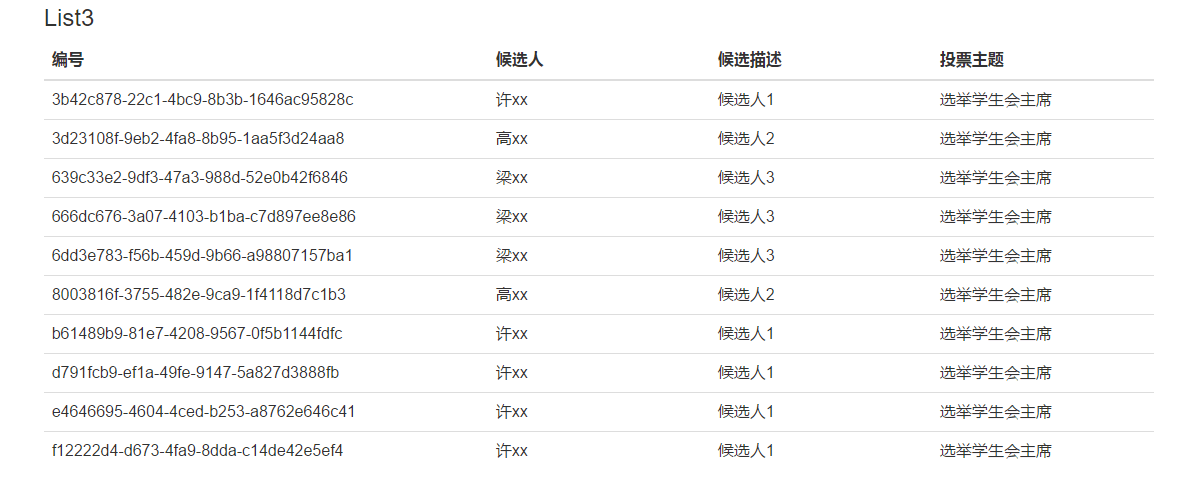
图1-3-2-4-1 List1

图1-3-2-4-2 List2

图1-3-2-4-2 List3

投票人和投票中心可凭借手中持有信息对信息表进行公开验证，对各阶段进行有效监督。

## 1.4 项目特色

**1.4.1项目创新**

对比于其他电子投票系统，本项目的创新点在于：

1. **具有稳固性，可在任意阶段弃票**

投票者在任意阶段弃权，体现在各公告信息表中的投票者数目不相等，并不会影响选举的正常进行。每一个阶段的操作的数据，都基于上一个阶段，因此当投票者没有进行前一个操作，对应的数据不会出现在下一个阶段的信息表中，则该投票者的投票不会被计算，则投票者在任何阶段的弃票都不会影响下一个阶段。

在注册阶段，用户部分信息会公布在于List1上，包含标识用户身份的序列码。只有具有序列码的投票者才能进行下一个阶段--投票阶段，因此在注册阶段弃票不会影响后续投票功能。

同时只有在投票后公布的List2中出现了序列码的投票者的选票才能在下一个阶段被计算，因此在投票阶段弃权不会影响后续计票。

在计票阶段，投票人将消盲后的选票公布于计票信息表List3中，系统根据List3进行计票。超过计票期限，投票者依然未将选票公布于List3中，则被视为中途弃权，不会影响计算选票。

因此，投票者在任何阶段弃权都不会影响整个系统的工作，则本产品具有稳固性。

1. **不依赖任何第三方和匿名通道，实现公平可验证性**

对投票者进行无条件匿名注册，同时为投票者发放具有唯一性的合法临时身份用于后续的投票及计票过程，任何人只能确定投票者的合法性而无法得知其真实身份，因此无需借助匿名通道。

同时，每个过程都接受监督检验，任何不诚实行为也可以被制止。

从整个系统的角度，涉及到投票者信息的投票中心会将信息公布到对应的公告信息表中。

在投票阶段，投票中心将 ｛消息，对消息的关联环签名，对消息的普通个体签名，Ni｝保存到注册信息表List1中，公布List1，接受公开验证；在注册公示时间内，未能在注册期及时注册的投票者可以公开注册，公布自己的签名，要求注册中心为其发放身份序列码和对应签名，并公布其注册信息于List1中；同时已注册者查询List1，如果查询不到自己的注册信息，投票者可以公示自己的签名{消息，对消息的关联环签名，对消息的普通个体签名，Ni}，并公开验证签名。对消息的关联环签名和私钥对消息进行的普通个体签名的正确性，进而要求投票中心在所有人的监督下为其发放身份序列码和对应签名，并公布投票者相应信息于List1上。能够有效防止注册中心的不诚实行为。

在投票阶段，将序列号、选票、选票的签名信息加入List2。投票结束后公布List2，接受公开验证；在投票公示时间内，未能在投票期及时投票的投票者可以公开投票，公布自己的盲化选票和签名，要求投票中心为其发送签名并将其投票结果公布于List2中；同时已投票者查询List2，如果查询不到自己的投票信息，投票者可以公布自己的序列码、盲化选票和签名，在所有人的监督下，要求投票中心为其签名并发送选票、选票的签名，并将其投票结果公布到List2中。能够有效防止数据管理机构的不诚实行为。

在计票阶段，投票人将消盲后的选票公布于计票信息表List3中，能够有效防止投票中心的不诚实行为。

从而，整个系统的各个模块独立、公开、透明，接受众人的监督，具有可验证性。

从投票者角度，任何不诚实的投票行为也会被发现和制止，也具有公平可验证性。

首先，保证了投票人的身份合法具有注册资格，进而获得注册机构为其分配的唯一标识身份的序列码Ｎ。其次，在其后的投票和计票环节中，都使用Ni，且Ni分别公布在List2，List3中，如投票者重复投票，则会在对应表中出现冗余，从而杜绝投票者重复投票行为。

因此，每个过程都将接受大家的监督，即可以防止第三方泄密，同时也具有公平可验证性。

1. **提升了关联环签名验证身份的效率，适用于大规模选举**

环签名的大计算量成为制约其应用于大规模电子投票的主要因素。与现有方案不同的是，本产品将计算量大的关联环签名应用于注册阶段实现合法投票者的匿名注册，并通过为合法投票者发放一个唯一的临时身份标识Ｎ，并结合自己产生的公私钥对，用于后续投票和计票阶段的匿名身份认证。由于本产品在投票阶段仅使用公私钥对实现普通的个体签名及验证，故有效保证了投票阶段的高效性和实用性，适合于大规模选举。

1. **解决了匿名性，安全性，公平性之间的矛盾**

电子投票系统的安全需求存在内在的矛盾性。

矛盾一：一方面，投票系统需要向投票者提供证据表明前者正确地理解了后者的意图；另一方面，投票系统不能提供给投票者任何票据，使投票者能够向第三方证明自己的投票内容。

矛盾二：一方面，投票系统需要向公众提供证据表明所有的选票都被正确的计数；另一方面，这些证据不能损害投票者的匿名性。

针对矛盾一，本产品在用户生成选票、投票前时，会用友好的交互方式，向用户显示用户的投票内容，让用户进行确定。在投票后，不产生任何票据，则用户不能向第三方证明自己的投票内容。

针对矛盾二，在计票阶段，本年产品的计票模块会公布用户选票信息表List3，包含标识唯一用户的序列码Ni，消盲后的选票。投票者可以根据List3查看是否有自己序列码，从而判断自己的选票是否被进行了正确计算。同时，其他人无法根据序列码判断每个序列码对应的具体的用户，因此保证了投票者的匿名性。

本系统提出的电子投票系统的安全需求矛盾的解决方案，并且基于此方案构建出了满足匿名性、安全性、公平性的投票系统。

## 1.5可行性分析

本产品的运行过程中，使用了多项技术，包括数字签名、盲签名、环签名等；

在用户进行注册前，系统需要初始化，基于环签名理论初始用户群；在注册阶段，涉及到投票者对消息进行关联环签名、用私钥对消息进行普通个体签名；投票中心使用签名验证算法验证用户的签名，判断用户是否重复申请。注册中心分配给用户一个用于唯一标识身份的序列码，也要对其进行签名。

在投票阶段，盲化选票还涉及到了盲签名等理论；

在本产品中大量涉及到签名、验证签名、加密等算法。以下是对每项使用到的技术的理论介绍。

1. **数字签名理论**

计算机和网络技术的发展将人类带入信息化社会，随之而来的是备受关注的信息安全问题。现代密码学已成为信息安全技术的核心，而数字签名是现代密码学研究的主要内容之一。手写签名是一种传统的确认方式，如写信、签订协议、支付确认、批复文件等。显然，数字系统的特点决定了不可能再沿用原先的手写签名方法来实现防伪造和抗抵赖。因此，随着数字化时代的到来，作为一种重要且有效的信息安全技术——数字签名将在政治、军事、电子商务和电子政务等领域有着极为广泛的应用。

国际化标准组织对数字签名有如下定义：附加在数据单元上的一些数据，或是对数据单元所做的密码变换，这种数据或变换允许数据单元的接收者用以确认数据单元来源和数据单元的完整性，并保护数据，防止被人（如接收者）伪造。数字签名是对电子形式的消息进行签名的一种方法，一个签名消息能在一个通信网络中传输。数字签名作为实现电子交易安全的核心技术之一，在实现和保证信息传输的完整性、发送者的身份识别和认证、抗抵赖性等方面都有重要应用。数字签名的目的之一，就是在网络环境中代替传统的手工签字与印章，其可抵御的网络攻击主要有：(1)防冒充（伪造）；(2)可鉴别身份；(3)防篡改（防破坏信息的完整性）；(4)防重放；(5)防抵赖；(6)机密性（保密性）。满足上述要求的数字签名能够实现以下功能：(1)接收方能够证实发送方的身份；(2)发送方在事后不能否认签发的报文；(3)接收方或非法者不能伪造、篡改报文。另外，使用手写签名一旦出现争端，第三方不容易仲裁；而数字签名能给仲裁者提供足够的证据来进行裁决，因此其安全性远远高于前者。此外，数字签名依托于计算机网络，其时效性远远大于前者。

数字签名技术以密码理论为基础，其核心是采用加密技术的加、解密算法来实现对报文的数字签名。从目前的密码学体制来分，可分为对称密钥密码体制和非对称密钥密码体制（即公钥密码体制）。前者是指加密和解密都使用相同的密钥；后者是指加密和解密使用不同的密钥，在这种密码体制中，有一个密钥可以公开，称作公开密钥；另一个密钥保密，称作秘密密钥。这种体制中最有代表性的算法为RSA数字签名算法，这也是本系统使用的签名算法，用于在投票人和投票中心信息交互间防止信息篡改。

数字签名是签名者利用自己私钥对消息进行加密生成签名，验证者利用签名者公钥进行解密验证签名的一种密码技术。一个数字签名体制主要由下列部分构成：(1)一个明文消息空间M：所有待签名的消息组成的集合；(2)一个签名空间Q：所有签名形成的集合；(3)一个签名密钥空间K：所有的可能用于生成签名的密钥的集合；(4)一个认证密钥空间K’：所有的可能用于验证签名的密钥的集合；(5)一个有效的密钥生成算法Gen．N->K×K7，其中K和K’为(3)中定义，N为安全参数空间，当输入安全参数空间N内的一个安全参数时，算法Gen在多项式时间内生成一对公私钥对；(6)一个有效的签名算法瓯印：M×K--->Q，对于消息空间M内一个给定的消息，算法Sign利用算法Gen生成的签名密钥，在多项式时间内生成一个有效的签名：(7)一个有效的验证算法，对于给定的消息和算法Sign生成的签名，利用算法Gen生成的验证密钥验证签名的正确性。

数字签名具备以下基本性质：(1)不可伪造性。只有拥有签名密钥的签名人才能生成正确的数字签名，而在未知签名者私钥的前提下，敌手很难伪造一个能够通过验证的合法数字签名。(2)不可篡改性(保证消息的完整性)。通常情况下，签名者不是对原消息而是对消息的哈希函数值进行签名，然后再将原消息和签名作为一个完整的整体捆绑在一起。由于一个安全的哈希函数是具有抗碰撞性的，即找到两个具有相同哈希函数值的不同消息是困难的。所以，只要对消息做出任何篡改便会导致(消息一签名)对无法 通过验证算法的验证。因此，数字签名具有不可篡改性，即能保证消息的完整性。(3)不可否认性。对于一般的数字签名，任何人均能利用公开的算法验证签名的正确性，并且只有真实签名者的公钥才能满足验证等式。而每个签名者都有一个与其身份对应的公钥证书，所以一旦签名生成，签名者便不能否认自己对消息的签名。因此，数字签名具有不可否认性。数字签名的不可否认性可用来进行身份认证。

1. **盲签名理论**

盲签名(Blind Signature)概念自从Chaum于1982年提出以来，由于其对信息的保密性而得到了广泛的应用。特别是在匿名电子投票和匿名电子支票等电子商务中，盲签名可以被用来保护用户的各种隐私。简单来说，盲签名允许用户得到关于一份信息的签名的同时不向签名方泄漏该信息的内容。1996年，Abe和Fujsaki首次提出了部分盲签名(Partially Blind Signature)的概念，它与盲签名的区别在于，用户除了自己的秘密信息以外，还与签名方享有一部分公有的信息。

盲签名是一种特殊的数字签名，由Chaum于1982年首次提出，并于1983年提出了第一个基于RSA公钥密码的盲签名方案。随后人们又基于离散对数问题和二次剩余提出了各种盲签名方案，目前已有多种成熟的盲签名方案应用在同时需要匿名性和认证性的应用场合中，如在电子现金中实现不可跟踪的电子现金，在电子选举中实现无记名选举等。虽然目前已有许多盲签名模型，并且得到了广泛的应用，但是现有的盲签名方案的安全性大多基于数学困难问题，量子算法与量子计算机的发展对它们构成了极大的威胁。

盲签名是一种特殊的数字签名，可以看作消息拥有者和签名人之间的一种交互协议。如果协议正确执行，消息的拥有者最终将获得签名人对消息的数字签名。而签名人却不知道消息的内容，即便以后将消息和签名公开，也无法追踪消息与自己执行签名过程之间的相互关系。

盲签名不仅具有一般数字签名的所有性质，还有一些特有的性质，一个安全的盲签名体制至少需要满足下面五个性质：

1. 正确性。对于盲签名算法正确执行后输出的对消息的签名，则使用验证函数总可以得到正确的结果。
2. 盲性。也称为匿名性，是盲签名最主要的特性。除请求签名的消息拥有者外，消息内容对任何人(包括签名人)均不可见。即使签名人对消息签名，仍然得不到消息的具体内容。
3. 不可伪造性。任何不知道签名人私钥的人都无法有效地计算出一个能够通过签名验证方程的消息—签名对。
4. 不可抵赖性。只要证明消息的签名是合法的，那么签名人不论如何都无法否认他签过这个消息。
5. 不可跟踪性。一旦签名消息公开，签名人不知道何时签的，即使留有当时的签名信息，也无法追踪到消息内容。

盲签名的这些良好特性应用在本产品中盲化选票步骤中，保证了选票的匿名性。投票中心接收及公布的选票均是盲化后的选票，因此在计票之前，除投票者自身外任何人都无法获知选票内容有效防止选票内容向第三方出示的行为。

1. **环签名理论**

Rivest和Shamir等人于2001年首次提出了环签名的概念，给出了分别基于RSA和Rabin的两个环签名方案，并且证明了RSA版本方案在Random Oracle模型下的安全性。环签名其实就是简化了的群签名，它要求只有该组内的成员可以完成签名，同时不泄漏签名者的确切身份。然而，与群签名不同的是，环签名不存在组织管理者，不存在回收过程，不需要组内成员之间的任何合作。组内的任何成员在已有自己的私钥和其他所有成员公钥的情况下，不需要其他成员的帮助即可签署消息，且不会受到其他成员的干预。当验证者得到这样的一份签名后，除了能够验证出该签名是否出自该组，是否合法的签名以外，无法得到有关该成员身份的确切信息。

环签名解决了群签名中群管理员权限过大导致群签名具有可追踪性的缺点，它具有对签名人无条件匿名的特点，验证者能够确定的是签名者为环中某一成员，但不能确定真正签名者的身份。因此非常适合于需要对信息进行长期保护的特殊环境中。目前，环签名在机密信息的泄露，匿名电子投票，电子现金系统，电子政务以及无线传感网络等方面都有着广泛的应用，因而成为当前密码学研究的一个热点。然而，在国内对环签名的研究才处于初期阶段，并且在不同环签名方案被提出的同时，也发现环签名依然存在许多问题亟需解决，如签名长度正比于环成员总数，方案的可证明安 全性仍有待提高，怎样设计出新的基于不同密码体制的特殊环签名以及利用不同的环签名的解决现实需求等等。

关联环签名在环签名的基础上，通过增添关联标签，使得验证者能够确定不同环签名是否由同一签名者所签。然而，现有的关联环签名方案均是建立在离散对数公钥密码体制上，且绝大多数方案仅具备弱匿名性的特点。因此，如何设计出基于其他公钥密码体制、具有强匿名性的关联环签名具有重要的理论价值和研究意义。

选举作为普通公民依法享有的一项基本权利，在与日俱进的社会环境中，它的方式不可能一成不变。传统的选举方式浪费大量的人力资源，而且效率很低，还会由于人为的因素引起许多失误和违规行为。虽然电子投票已经在实践中有了一定的应用，但其应用范围还远远不够，主要由于现有电子投票方案依然存在安全性不高和效率低下等问题，使得电子投票在某些安全性级别要求更高或者需要适用于大群体的应用场合存在一定的局限性。因此，如何利用环签名等新型密码学技术，解决电子投票系统中存在可信中心、选票碰撞、投票者中途弃权、效率低下、不适合于大群体选举等问题，设计出更公平、高效、安全的电子投票方案，具有重要的现实意义。

环签名一般主要由签名算法和签名验证算法两部分组成，有一些环签名算法还有系统初始化过程，主要是为了对签名时所需的某些公开参数进行设置。这里仅仅对环签名中的签名算法和验证算法进行描述。

环签名具有一条和群签名相同的性质是签名者模糊性，然而不同点是环签名可以实现真实签名者的无条件匿名性，即任何人任何时候都无法判断真实签名者的身份信息，而群签名中群管理员可以随时揭露出具体签名者，只需要利用陷门信息就可以做到。

环签名的生成过程是：真实的签名者和任意选取的一组成员形成一个集合，用自己的私钥和其它成员的公钥对消息签署信息。签名者组建的这个集合称作环（Ring），任意一个环成员代表整个环所产生的签名信息我们称作环签名（Ring Signature）。能够被签名接收者和签名验证者确认签名信息是来自某一个环，但无法知道真正的签名者身份。环签名具有以下特点：

1. 不需要预先建立集合，使成员提前加入和撤出，集合的形成只是根据需要由签名人自己在签名前指定，随时组建这个集合。
2. 无特殊的管理者。
3. 不可能揭露真实签名人的身份信息，验证者能通过验证确定签名信息的合法性以及签名者是环中成员，但没有人能够准确判断具体是哪一位成员作出了签名信息。无条件匿名性和不可伪造性是一个环签名方案必须满足的安全性能。无条件匿名性：攻击者即使可以非法获取所有可能签名者的私钥，也不能由此推断出真正签名者的身份信息。不可伪造性：外部攻击者不知道真实签名者私钥的条件下，截获多条签名后的消息，他也不可能分析出签名者的私钥进而以不可忽略的概率成功伪造一个新消息的合法环签名。

环签名应用在本产品的系统初始化，对投票者进行无条件匿名注册。投票者利用关联环签名进行匿名注册，投票中心只能确认投票者为n个合法投票者之一，但无法确定投票者真实身份．同时投票者为自己生成一对公私钥对，与注册中心为其发放的具有唯一身份标志的序列码Ni相对应，并用于后续投票和计票过程的身份验证。任何人只能确定以对应某一合法投票者，却无法得知其真实身份。很好实现了投票者的匿名性。

**第二章 实现方案**

**2.1系统架构设计**

**2.1.1系统架构总体设计**

本系统旨在开发一个基于关联环签名的安全电子投票平台。解决电子投票中存在的选票碰撞、第三方机构（投票中心）欺诈选民、需借助匿名通信信道发送选票和投票效率低等问题，接合电子投票的基本原理，采用关联环签名技术实现匿名注册，利用盲签名盲化选票和引入身份序列码保证选票唯一性的方法，设计了一个抗欺诈的安全电子投票平台。

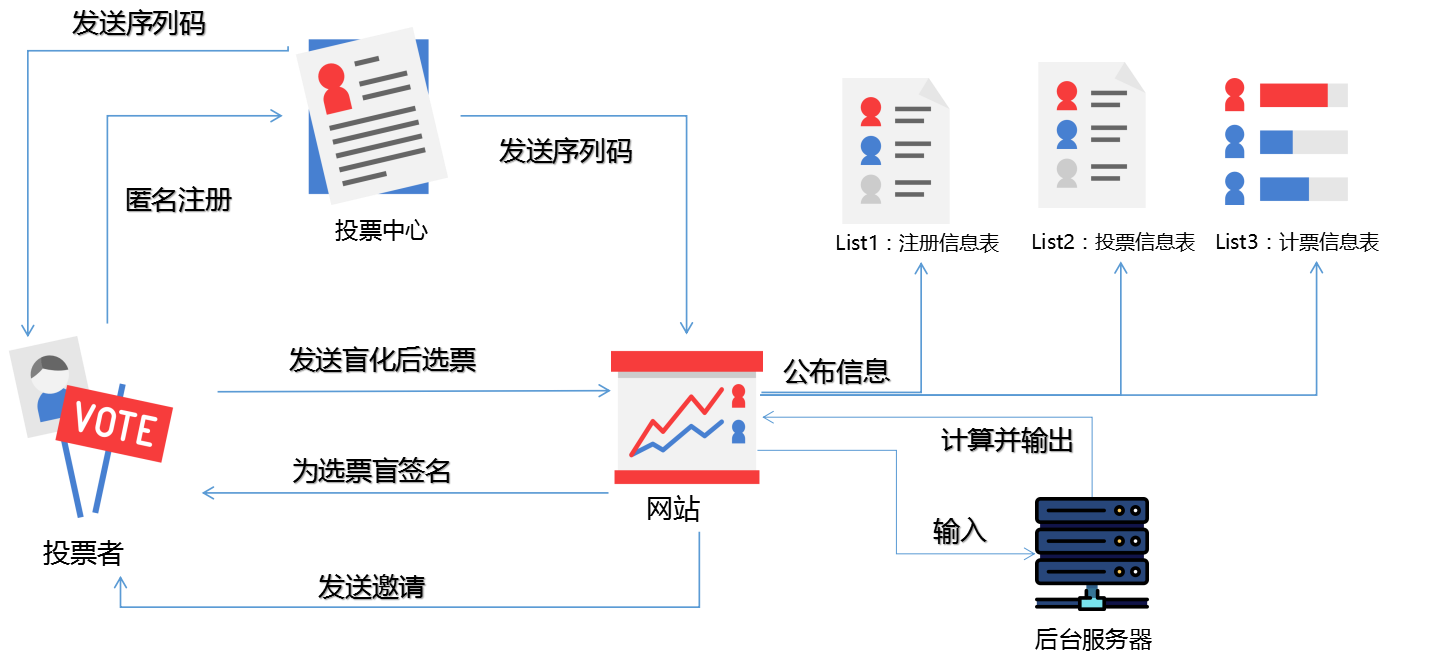
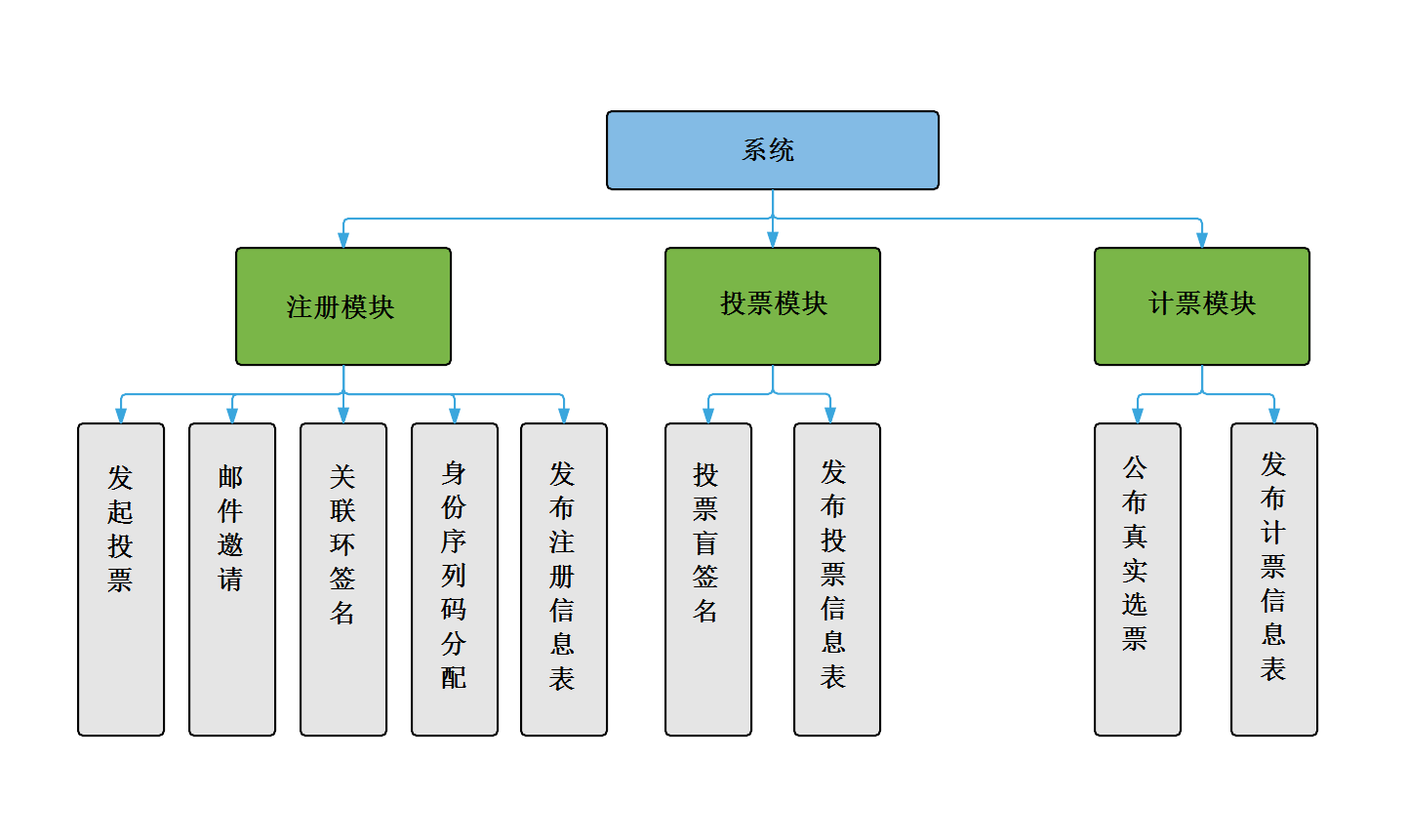
整个系统可分为三个部分，投票用户端、网站端、投票中心端。投票用户作为关联环体中的某一成员进行匿名投票表决。网站端作为投票用户和投票中心之间的中枢架起相互信任的桥梁，对整个投票过程中数据的真实有效负责，并在公告牌上明示注册信息表List1，投票信息表List2，计票信息表List3以供相互查对。投票中心端验证每一个投票用户的身份合法性，为合法投票者发放身份序列码Ni，验证盲化选票的合法性，并为盲化选票签名。

图2-1-1-1 系统架构示意图

1. 投票者：作为环体中的某一成员对候选人进行投票。任何人只能确定投票者的合法性而无法得知其真实身份
2. 投票中心：在网站上创建投票，设置投票人范围。验证每一个投票者身份的合法性，为合法投票者发放身份序列码Ni，并为Ni签名。
3. 网站：在投票者、投票中心与后台服务器建立信息交互桥梁，显示信息。
4. 后台服务器：初始化系统；邀请合法投票人进行注册；验证盲化选票的合法性，并为合法选票进行盲签名。对投票者去盲后的选票进行验证并签名，然后统计并公布最终结果。同时公布注册、投票、计票阶段对应的信息表。
5. 信息表：List1注册信息表，List2投票信息表，List3计票信息表，接受各方的监督与验证。

**2.1.2系统模块划分**

图2-1-2-1 系统功能模块图

1. 注册模块：注册模块主要为用户实现匿名注册。首先由注册中心创建投票主题，设置投票者范围。注册模块根据范围发送邀请。投票者接受邀请后，进行匿名注册，其中使用关联环签名实现。同时注册模块为每一位投票者分配标识唯一身份的序列码，实现抗选票碰撞。结合前述二者，保证了投票者的秘密性。其他人只能知道某序列码是合法投票者中的一员，而无法知道其具体身份。同时，注册模块公布注册信息表，保证未及时注册的投票者及时注册。
2. 投票模块：投票模块主要实现匿名投票。投票者生成选票后，使用盲签名盲化选票，发送给对应功能模块。后者再为该选票进行签名，保证选票的正确性。同时投票模块发布投票信息表，可防止投票者未及时投票，以及防止选票泄露，并接受公众监督。
3. 计票模块：计票模块根据投票模块的数据为选票进行消盲，并进行计票，且将选票信息公布，接受公众监督。

**2.2系统概要设计**

**2.2.1系统用例图设计**

整个系统完成的功能详见如下用例图：

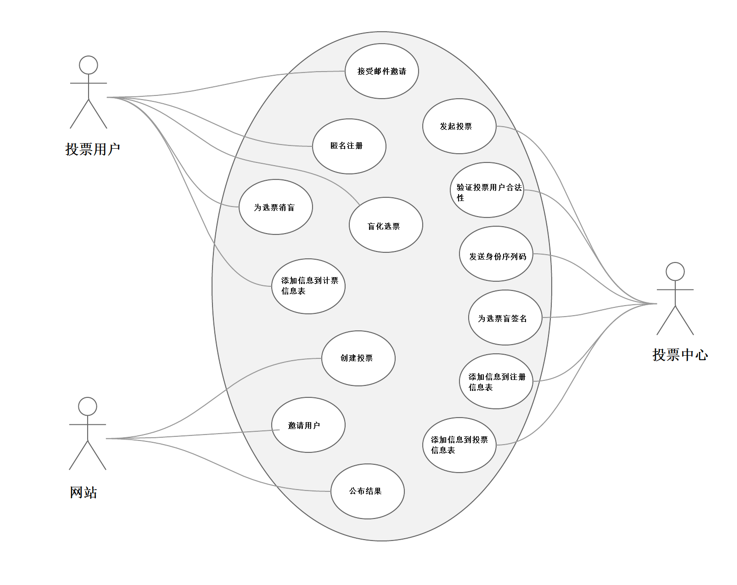


图2-2-1-1 系统用例图

**2.2.2系统流程设计**

**2.2.2.1系统概要流程设计**

1. 投票中心在网站上创建投票，设置投票人范围；
2. 网站通过注册中心的设置，向投票人发起邀请；
3. 投票者通过邀请在投票中心进行匿名注册，投票中心分配给投票者用于标识唯一身份的序列码；
4. 投票者发送盲化选票，投票中心为其盲签名；
5. 网站进行计票，公布信息表List1（包含用户序列码、及其签名、对消息的关联环签名、私钥对消息进行的普通个体签名、公钥），List2（身份序列码、盲化后的选票、选票的盲签名），List3（身份序列码、消盲后的选票），及选票的最终结果，接受验证。

**2.2.2.2系统具体流程设计**

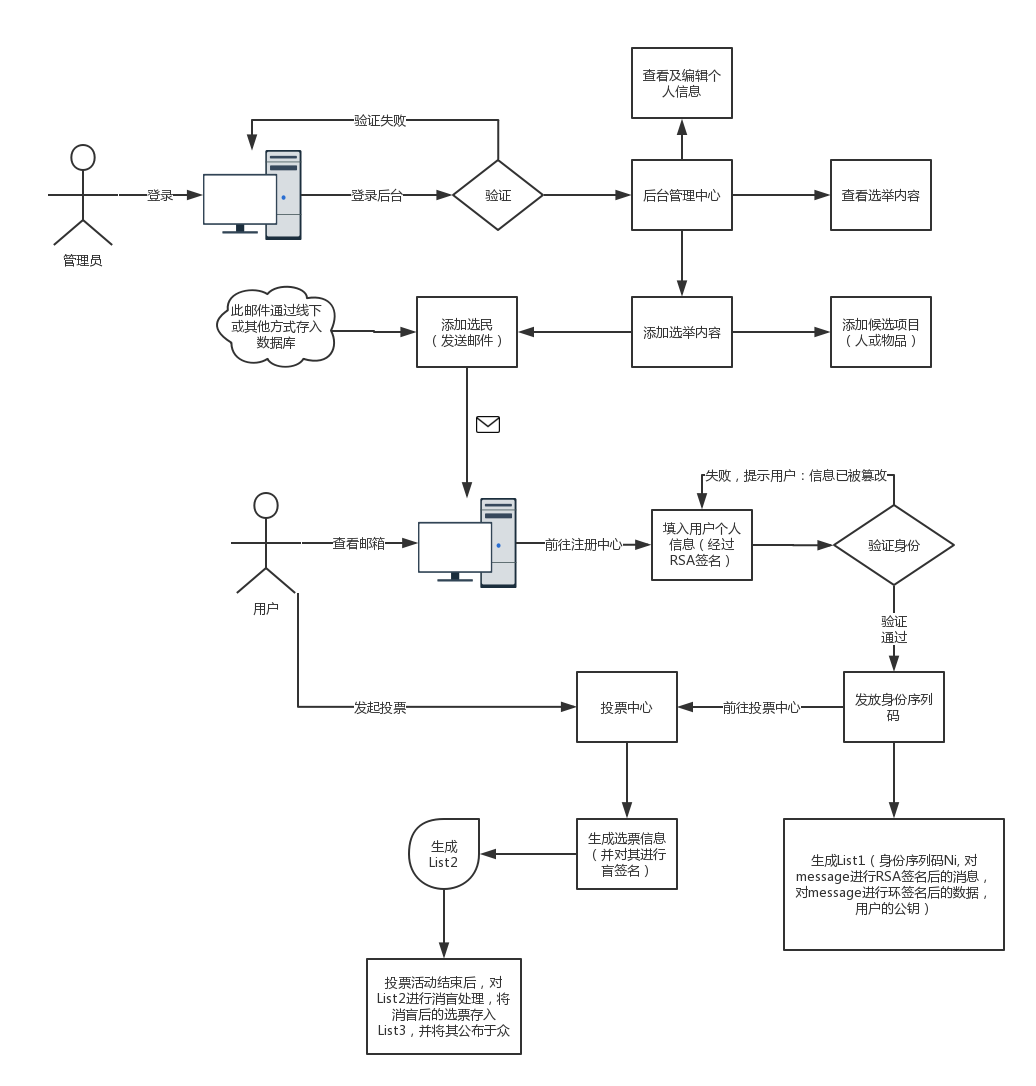


图2-2-2-2-1 系统具体流程图

1. **注册阶段**

通过验证的合法投票者在网站进行匿名注册，具体步骤如下：

1. 按照提前生成好的公私钥对系统进行初始化，其中公钥集合L={y1,y2,……,yn}由所有具备选举几个者公钥组成。
2. 投票者Vi(1≤i≤n)利用关联环签名生成算法对消息m进行关联环签名，生成签名(C,S1,S2,……,Sn,Y)，同时为自己生成另外一对基于RSA签名体制的公私钥对，并利用其中的私钥对消息m进行普通的个体签名，最终将消息，两个签名以及自己生成的公钥一起发送给投票中心。
3. 投票中心按照关联环签名验证算法，首先验证签名(C,S1,S2,……,Sn,Y)的正确性。若正确，则检查该签名中的Y值是否已存在于注册信息表List1中，如果存在，则说明Vi重复注册，拒绝此次请求；如果不存在，说明Vi为新的合法投票者。
4. 如果Vi是新的合法投票者，投票中心利用收到的公钥验证普通签名是否正确。若正确，投票中心为Vi随机选取一个具有唯一标识的身份序列码Ni，并对Ni签名；若不正确，投票中心要求投票者重新发送消息。最后，投票中心将身份序列码Ni和签名结果发送给投票人Vi，同时将相关信息保存到注册信息表List1中。
5. 注册结束后网站公布List1,接受公开验证。在注册公示时间内，未能在注册期及时注册的投票者可以公开注册，公布自己的签名，要求投票中心为其发放身份序列码和对应签名，并将其信息添加到List1中，同时已注册者查询List1，如果查询不到自己的注册信息，投票者可以公示自己的签名，并公开验证消息m的关联环签名和普通签名的正确性，进而要求投票中心在所有人的监督下为其发放身份序列码和对应签名，并添加投票者的相应信息于List1上。上述方法能够有效防止注册中心的不诚实行为。此外，若公示时间结束后，投票者依然没有注册，则被视为自动弃权。
6. **投票阶段**

在规定的投票期限内，投票者按照如下协议进行投票（只有Ni存在于List1的投票者才有权参与如下投票过程）：

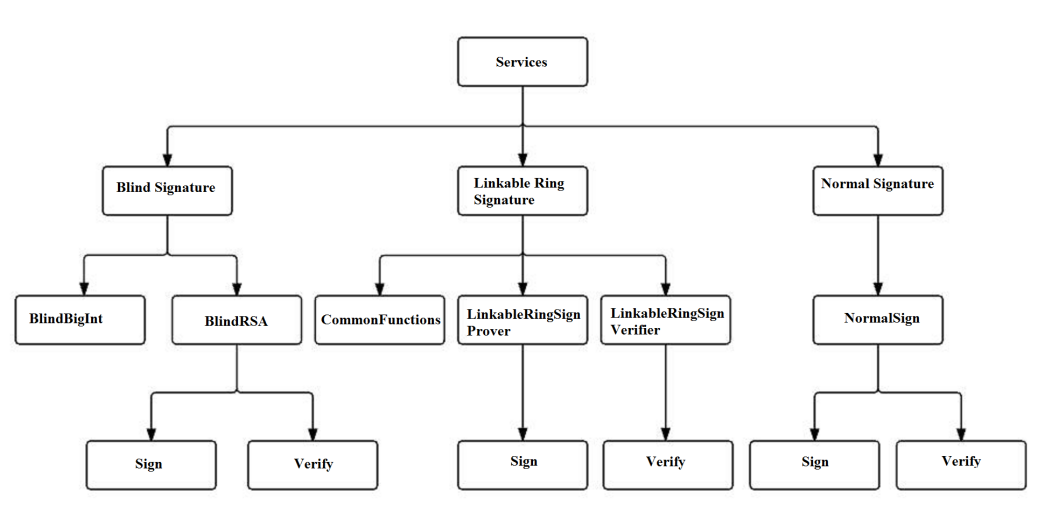
1. 投票者Vi首先生成电子选票Ei，然后随机选择一盲化因子对选票进行盲化。
2. Vi对选票进行签名，然后将盲化选票、Ni、选票签名发给投票中心。
3. 投票中心首先检查Ni是否已经存在于投票信息表List2中。如果已经存在，说明投票者Vi重复投票，拒绝此次请求；如果不存在，验证签名的正确性。
4. 若签名正确，投票中心为Vi的盲化选票进行签名，并将盲化选票和签名发送给Vi，同时将相关信息保存到投票信息表List2中；若不正确，则要求投票者重新发送。
5. 投票结束后公布List2，接受公开验证。在投票公示时间内，未能在投票期间及时投票的投票者可以公开投票，公布自己的盲化选票和签名，要求投票中心为其发送签名并将其投票结果公布于List2中；同时已投票者查询List2，如果查询不到自己的投票信息，投票者可以公布自己的盲化选票和签名，在所有人的监督下，要求投票中心为其签名并发送盲签名等信息给投票者，并将其结果公布到List2中。如果公示期结束，投票者依旧没有投票，则被视为中途放弃。
6. **计票阶段**

投票人Vi将消盲后的选票公布于计票信息表List3中，网站根据List3统计并公布选举的最终结果（只有Ni存在于List2中的投票者才有权参与如下计票过程）：

1. Vi对投票中心的盲签名做消盲，进而得到投票中心对选票的签名。
2. Vi对选票Ei签名，然后将相关信息公布于List3中。任何人均可验证最终计票信息的正确性。若投票中心和投票人对投票的签名都正确，说明选票为Vi的合法投票，并且选票没有被篡改。如果Ni在List3中重复出现，说明投票者Vi重复投票，仅对其选票统计一次。
3. 超过机票期限，投票者依然未将选票公布于List3中，则被视为中途弃权。
4. 网站根据List3统计并公布投票的最终结果。

**2.2.3网站数据处理模块设计**

服务器数据处理模块使用C++开发，用于处理投票用户和投票中心传来的数据。结构图如下：

图2-2-3-1 数据处理模块结构图

数据处理模块主要完成分配公私钥对、关联环签名的签名与验证、盲签名的签名与验证、普通RSA签名的签名与验证等功能。在整个系统中发挥了至关重要的作用。

**2.2.4网站设计**

Web后端使用SSH框架开发，并采用Tomcat作为服务器；主要分为投票管理中心和用户投票中心两部分。系统采用MVC架构，使用Strus2作为控制层，负责数据的处理与转发；Hibernate框架负责对象的持久化工作；Spring负责事务的管理以及对象的创建工作。Web前端使用jsp实现，并使用jQuery、BootStrap作为UI支持，使用Ajax技术提高了用户交互。

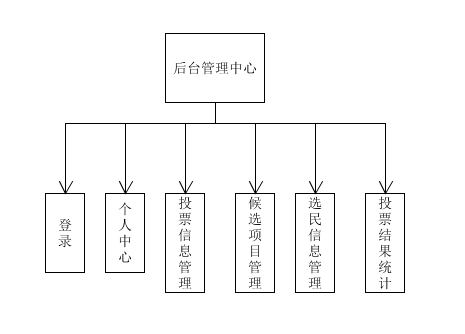


图2-2-4-1后台管理中心架构图

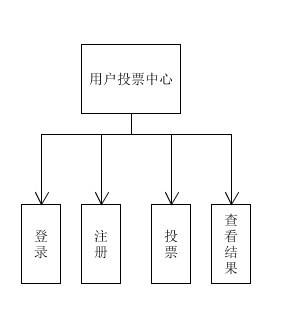


图2-2-4-2 用户投票中心架构图

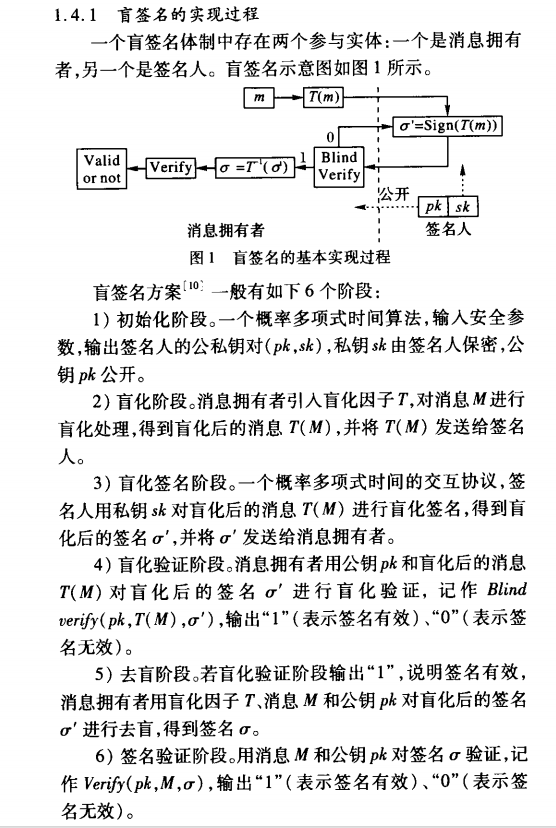
**2.3系统详细实现**

**2.3.1网站数据处理模块实现**

**2.3.1.1 BlindSignature（盲签名算法部分）**

1. **盲签名算法实现**

一个盲签名体制中存在两个参与实体：一个是消息拥有者，另一个是签名人。

图2-3-1-1-1 盲签名算法示意图

盲签名方案一般有如下六个阶段：

1. 初始化阶段。一个概率多项式时间算法，输入安全参数，输出签名人的公私钥对（pk,sk）,私钥sk由签名人保密，公钥对公开。
2. 盲化阶段。消息拥有者引入盲化因子T，对消息M进行盲化处理，得到盲化后的消息，并将发送给签名人。
3. 盲化签名阶段。一个概率多项式时间的交互协议，签名人用私钥sk对盲化后的消息进行盲化签名，得到盲化后的签名，并将发送给消息拥有者。
4. 盲化验证阶段。消息拥有者用公钥pk和盲化后的消息对盲化后的签名进行盲化验证，记作，输出“1”（表示签名有效），“0”（表示签名无效）。
5. 去盲阶段。若盲化验证阶段输出“1”，说明签名有效，消息拥有者用盲化因子T，消息M和公钥pk对盲化后的签名进行去盲，得到签名**σ**。
6. 签名验证阶段。用消息M和公钥pk对签名σ验证，记作，输出“1”（表示签名有效），“0”（表示签名无效）。
7. **盲签名代码实现**
8. **BlindBigInt类**

| **编号** | **函数名** | **说明** |
| --- | --- | --- |
| 1 | BlindBigInt::BlindBigInt() | 构造函数1 |
| 2 | BlindBigInt::BlindBigInt(int) | 构造函数2 |
| 3 | BlindBigInt::BlindBigInt(int,int) | 构造函数3 |
| 4 | BlindBigInt::~BlindBigInt() | 析构函数 |
| 5 | BlindBigInt::isZero() | 判断是否为0 |
| 6 | BlindBigInt::toULong() | 将数值转换成是十进制的长整型数 |
| 7 | BlindBigInt::toString() | 将数值转换成字符串 |
| 8 | BlindBigInt::toHexString() | 将数值转换成十六进制格式表示的字符串 |
| 9 | BlindBigInt::operator[]() | 获得数值特殊一位上的值 |
| 10 | BlindBigInt::operator\*() | 将两个BlindBigInt数值相乘 |
| 11 | BlindBigInt::operator\*=() | 将两个BlindBigInt数值相乘并赋值 |
| 12 | BlindBigInt::operator+() | 将两个BlindBigInt数值相加 |
| 13 | BlindBigInt::operator>=() | 比较当前BlindBigInt数值是否大于等于参数BlindBigInt数值 |
| 14 | BlindBigInt::operator>() | 比较当前BlindBigInt数值是否大于参数BlindBigInt数值 |
| 15 | BlindBigInt::operator<=() | 比较当前BlindBigInt数值是否小于等于参数BlindBigInt数值 |
| 16 | BlindBigInt::operator<() | 比较当前BlindBigInt数值是否小于参数BlindBigInt数值 |
| 17 | BlindBigInt::operator/() | 将两个BlindBigInt数值相除 |
| 18 | BlindBigInt::operator==() | 判断两个BlindBigInt数值是否相等 |
| 19 | BlindBigInt::operator%() | 将两个BlindBigInt数值取余 |
| 20 | BlindBigInt::operator<<=() | 对BlindBigInt数值做左移位操作 |
| 21 | BlindBigInt::operator|=() | 将两个BlindBigInt数值做或位操作 |
| 22 | BlindBigInt::operator&=() | 将两个BlindBigInt数值做和位操作 |
| 23 | BlindBigInt::operator-() | 将两个BlindBigInt数值相减 |
| 24 | modPow() | x^ymodm |
| 25 | binToHex() | 将十进制字符串转换为十六进制字符串 |
| 26 | bitset::add() | 做位相加操作 |
| 27 | bitset::subtract() | 做位相减操作 |

表2-3-1-1-1 类BlindBigInt的实现

1. **BlindRSA类**
2. **实现欧拉函数的变量**

| **编号** | **变量** | **说明** |
| --- | --- | --- |
| 1 | Int p,q | 十七位的随机数,p!=q |
| 2 | BlindBigIntn | n=p\*q |
| 3 | BlindBigIntphi | Phi=(p-1)(q-1) |
| 4 | BlindBigInte | gcd(e,phi)==1 |
| 5 | BlindBigIntd | [e\*d==1]mod n |

表2-3-1-1-2 实现欧拉函数的变量

1. **实现欧拉函数的函数**

| **编号** | **函数名** | **说明** |
| --- | --- | --- |
| 1 | BlindRSA::BlindRSA() | 构造函数1 |
| 2 | BlindRSA::BlindRSA(int) | 构造函数2 |
| 3 | BlindRSA::BlindRSA(int,int) | 构造函数3 |
| 4 | BlindRSA::~BlindRSA() | 析构函数 |
| 5 | BlindRSA::GetPHI() | 返回变量phi |
| 6 | BlindRSA::getP() | 返回变量p |
| 7 | BlindRSA::getQ() | 返回变量q |
| 8 | BlindRSA::getPublicKey() | 返回公钥 |
| 9 | BlindRSA::getPrivateKey() | 返回私钥 |
| 10 | BlindRSA::getModulus() | 返回modulus=p\*q |
| 11 | BlindRSA::setPublicKey(unsignedint) | 设置公钥 |
| 12 | BlindRSA::setPublicKey(BlindBigInt) | 设置公钥 |
| 13 | BlindRSA::encrypt() | 加密 |
| 14 | BlindRSA::decrypt() | 解密 |
| 15 | BlindRSA::modInverse() | 反向取余 |
| 16 | BlindRSA::isPrime() | 素数验证 |
| 17 | BlindRSA::isPrimeMR() | 素数验证 |
| 18 | BlindRSA::isPrimeDiv() | 素数验证 |
| 15 | BlindRSA::gcd(int,int) | Euclidean算法 |
| 16 | BlindRSA::gcd(BlindBigInt,BlindBigInt) | Euclidean算法 |

表2-3-1-1-3 实现欧拉函数的函数

**2.3.1.2 LinkableRingSignature（关联环签名部分）**

1. **关联环签名算法实现**
2. 结合函数（combiningfunction）

结合函数环签名方案中的一个重要组成部分。

定义为结合函数，其中，k是输入，v是随机选取的初始值，是上的任意值，输出z是上的值。

选定k和v的值后，结合函数有以下的特点：

1. 对于每个s,，以及其他给定输入，方程，是从到输出z的一一映射。
2. 对于每个s,，给定一个b比特长读的值z，和除以外的所有的值，可以求解满足的b比特长度值。
3. 如果一个攻击者不能你想求解陷门方程中的某个方程，那么给定k , v和z,它就很难根据求解中的任何一个值。
4. 结合算数为以下形式：

其中定义为。

1. 只要输出z与v相等，该方程的计算过程就会连成一个环形，这也是环签名名称的由来。

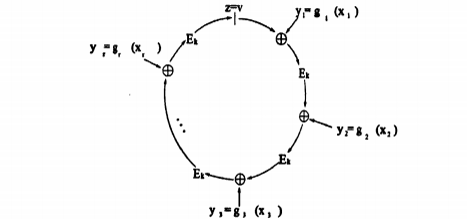


图2-3-1-2-1 环签名示意图

1. 关联环签名算法
2. **参数初始化**

设q是一个素数，是q阶子群,在G中求解离散对数问题是困难的。，是的单向散列函数。对i=1,2，...，n，每一用户拥有的私钥对，且。

1. **签名生成**

设消息，公钥集合为，签名者私钥为，对应公钥为，其中，利用如下算法产生关联环签名：

1. 计算和，然后选择，计算。
2. 对，选择，计算。
3. 计算。
4. 最后，输出签名。
5. **签名验证**

验证者受到关于m及L的签名后，按如下步骤验证签名的正确性：

1. 计算，然后对于计算，，，。
2. 检验是否成立，如果成立，则接受签名，否则拒绝此签名。
3. **关联环签名代码实现**
4. **CommonFunction类**

| **编号** | **函数名** | **说明** |
| --- | --- | --- |
| 1 | CommonFunctions::GetGroupParameters() | 获得群参数 |
| 2 | CommonFunctions::IntegerToString() | 将CryptoPP::integer转为字符串 |
| 3 | CommonFunctions::GenerateString() | 将向量中的各值转为字符串 |
| 4 | CommonFunctions::Hash1() | SHA1哈希函数 |
| 5 | CommonFunctions::Hash2() | SHA2哈希函数 |

表2-3-1-2-1 类CommonFunction实现

1. **LinkableRingSignProver类**
2. **实现的变量**

| **编号** | **变量** | **说明** |
| --- | --- | --- |
| 1 | Integer\_private\_key | 私钥 |
| 2 | unsignedint\_self\_identity | 当前环成员 |
| 3 | unsignedintnum\_members | 环成员人数 |
| 4 | Integerg | 群 |
| 5 | Integerq | 群序 |
| 6 | stringm | 签名消息 |
| 7 | vector<Integer>public\_keys | 公钥向量 |
| 8 | IntegerC | 关联环参数 |
| 9 | IntegerY | 关联环参数 |
| 10 | vector<Integer>S | 关联环参数 |

表2-3-1-2-2 LinkableRingSignProver变量

1. **实现的函数**

| **编号** | **函数名** | **说明** |
| --- | --- | --- |
| 1 | LinkableRingSignProver::LinkableRingSignProver() | 构造函数 |
| 2 | LinkableRingSignProver::GenerateSignature() | 产生签名 |
| 3 | GeneratePublicPrivateKeys() | 产生公私钥对 |

表2-3-1-2-3 LinkableRingSignProver函数

1. **LinkableRingSignVerifier类**
2. **实现的变量**

| **编号** | **变量** | **说明** |
| --- | --- | --- |
| 1 | IntegerC | 关联环参数 |
| 2 | IntegerY | 关联环参数 |
| 3 | vector<Integer>S | 关联环参数 |
| 4 | unsignedintnum\_members | 环成员人数 |
| 5 | Integerg | 群 |
| 6 | Integerp | 素数 |
| 7 | Integerq | 群序 |
| 8 | stringm | 签名消息 |
| 9 | vector<Integer>public\_keys | 公钥向量 |

表2-3-1-2-4 LinkableRingSignVerifier变量

1. **实现的函数**

| **编号** | **函数名** | **说明** |
| --- | --- | --- |
| 1 | LinkableRingSignVerifier::LinkableRingSignVerifier() | 构造函数 |
| 2 | LinkableRingSignVerifier::VerifySignature() | 验证函数 |

表2-3-1-2-5 LinkableRingSignVerifier函数

**2.3.1.3 NormalRSASignature（普通RSA数字签名部分）**

1. **普通RSA数字签名算法实现**

数字签名是签名者的利用自己私钥对消息进行加密生成的签名，验证者利用签名者公钥进行解密验证签名的一种密码技术。

数字签名体制一个数字签名体质主要由下列部分构成：

1. 一个明文消息空间M：所有待签名的消息组成的集合；
2. 一个签名空间Ω：所有签名形成的集合；
3. 一个签名密钥空间K:所有的可能用于生成签名的密钥的集合；
4. 一个认证密钥空间：所有的可能由于验证签名的密钥结合；
5. 一个有效的密钥生成算法，其中K和为（3）中定义，N为安全参数空间，当输入安全参数空间N内的一个安全参数时，算法Gen在多项式时间内生成一对公私钥对；
6. 一个有效的签名算法，对于消息空间M内的一个给定的消息，算法Sign利用算法Gen生成的签名密钥，在多项式时间内生成一个有效签名；
7. 一个有效的验证算法，对于给定的消息和算法Sign生成的签名，利用算法Gen生成的验证密钥验证签名的正确性。
8. 对任意的和任意的表示σ是用密钥sk生成的m的签名。
9. 对于任意的私钥，用pk表示与sk相对应的公钥，则对应于和必有
10. **普通RSA数字签名代码实现**
11. **normalSign类**
12. **实现的变量**

| **编号** | **变量** | **说明** |
| --- | --- | --- |
| 1 | stringm | 消息 |
| 2 | AutoSeededRandomPool | 随机种子参数 |
| 3 | IntegerN,P,Q,D,E | RSA签名参数 |
| 4 | stringencoded | 解密字符串 |
| 5 | boolresult | 验证签名结果 |
| 6 | InvertibleRSAFunction | RSA签名钥对 |

表2-3-1-3-1 类normalSign的变量

1. **实现的函数**

| **编号** | **函数名** | **说明** |
| --- | --- | --- |
| 1 | normalSign::normalSign() | 构造函数 |
| 2 | normalSign::getNormalRSASignature() | RSA数字签名 |
| 3 | normalSign::verifyNormalRSASignature() | RSA数字签名验证 |

表2-3-1-3-2 类normalSign的函数

**2.3.2网站实现**

**2.3.2.1 数据库设计**

1. **数据表**
2. List1

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | varchar | 40 | 用户序列码 | 必要 |
| signature | varchar | 200 | 序列码签名 | 必要 |
| ring\_signature | varchar | 200 | 关联环签名 | 必要 |
| personal\_signature | varchar | 200 | 普通个体签名 | 必要 |
| public\_key | varchar | 200 | 公钥 | 必要 |

表2-3-2-1-1 数据表list1

1. List2

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | varchar | 40 | 主键 | 必要 |
| blind\_vote | varchar | 200 | 序列码签名 | 必要 |
| signed\_vote | varchar | 200 | 关联环签名 | 必要 |

表2-3-2-1-2 数据表list2

1. List3

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | varchar | 40 | 主键 | 必要 |
| signature | varchar | 200 | 序列码签名 | 必要 |
| ring\_signature | varchar | 200 | 关联环签名 | 必要 |
| personal\_signature | varchar | 200 | 普通个体签名 | 必要 |
| public\_key | varchar | 200 | 公钥 | 必要 |

表2-3-2-1-3 数据表list3

1. 管理员信息表（admin）

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | varchar | 40 | 主键 | 必要 |
| name | varchar | 20 | 帐号 | 必要 |
| password | varchar | 50 | 密码 | 必要 |
| authority | int | 4 | 常住地址 | 必要 |
| email | varchar | 50 | 联系邮箱 | 必要 |

表2-3-2-1-4 管理员信息表

1. 投票信息表（voteinfo，用于保存一次投票事件的详细信息）

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | int | 10 | 主键 | 必要 |
| title | varchar | 200 | 标题 | 必要 |
| content | varchar | 500 | 详情 | 必要 |
| create\_time | datetime | (标准日期格式) | 创建时间 | 必要 |

表2-3-2-1-5 投票信息表

1. 候选名单表（voteitem）

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | int | 10 | 主键 | 必要 |
| name | varchar | 50 | 名称 | 必要 |
| description | varchar | 500 | 简介 | 必要 |
| create\_time | datetime | (标准日期格式) | 创建时间 | 必要 |
| infoid | int | 10 | 外键（参考voteinfo.id） | 必要 |

表2-3-2-1-6 候选名单表

1. 图片信息表（image，保存候选名单的图片路径）

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | int | 10 | 主键 | 必要 |
| path | varchar | 100 | 图片路径 | 必要 |
| itemid | int | 10 | 外键（参考voteitem.id） | 必要 |

表2-3-2-1-7 图片信息表

1. 投票结果表（ballot，保存投票结果）

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | int | 10 | 主键 | 必要 |
| infoid | int | 10 | 外键（参考voteinfo.id） | 必要 |
| itemid | int | 10 | 外键（参考voteitem.id） | 必要 |
| userid | varchar | 40 | 外键（参考user.id） | 必要 |

表2-3-2-1-8 数据表投票结果表

1. 选民邮箱（email，向指定选民的邮箱发起投票请求）

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | int | 10 | 主键 | 必要 |
| account | varchar | 50 | 邮箱帐号 | 必要 |

表2-3-2-1-9 选民邮箱

1. 邮件信息表（receiver，保存已发送邮件的信息）

| **字段名** | **数据类型** | **长度** | **描述** | **必要性** |
| --- | --- | --- | --- | --- |
| id | int | 10 | 主键 | 必要 |
| email | varchar | 100 | 邮箱帐号 | 必要 |
| send\_date | datetime | 标准日期格式 | 发送日期 | 必要 |
| infoid | int | 10 | 外键（参考voteinfo.id） | 必要 |

表2-3-2-1-10 邮件信息表

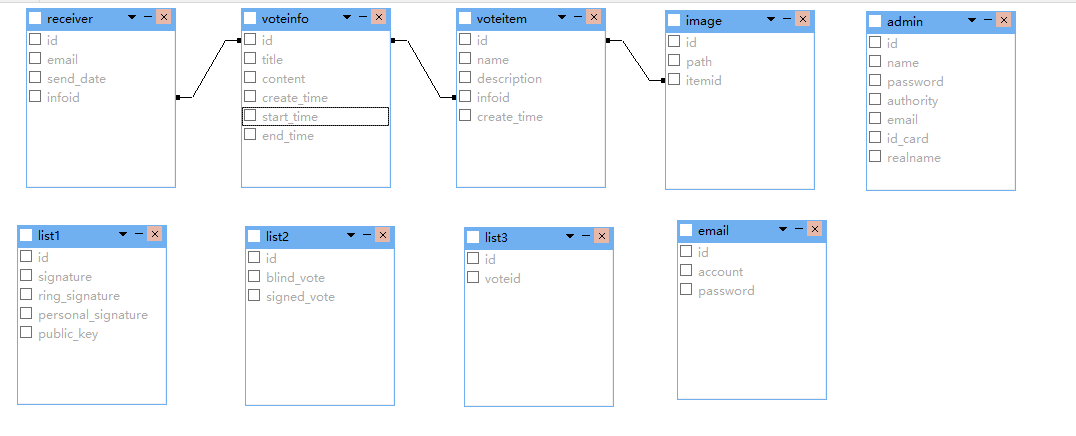
1. **数据库表映射关系图**

图2-3-2-1-1 数据库表映射图

**2.3.2.2 实体类**

1. Admin

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | String | private |
| name | String | private |
| password | String | private |
| realname | String | private |
| id\_card | String | private |
| authority | int | private |
| email | String | private |

表2-3-2-2-1 实体类Admin

说明：投票中心后台管理员，负责创建投票内容，添加候选名单，添加选民，发起投票。

1. User

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | String | private |
| name | String | private |
| email | String | private |
| phone | String | private |
| address | String | private |
| others | String | private |

表2-3-2-2-2 实体类User

说明：用户信息实体类，用于封装用户数据；当用户接收到管理中心发出的投票注册邮件，进行注册后，将用户填写的信息封装到User类中，并将该实体信息传入一个字符串中，然后对该数据进行验证，判断是否在传输的过程中数据遭到篡改。

1. Vote

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | int | private |
| infoid | int | private |
| itemid | int | private |
| userid | String | private |
| signature | String | private |

表2-3-2-2-3 实体类Vote

说明：投票结果实体类。

1. VoteInfo

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | int | private |
| title | String | private |
| content | String | private |
| createTime | String | private |
| startTime | String | private |
| endTime | String | private |
| items | List<VoteItem> | private |

表2-3-2-2-4 实体类VoteInfo

投票内容实体类：由管理员创建一个投票信息，并将该投票信息保存到数据库中。

1. VoteItem(候选名单实体类)

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | int | private |
| name | String | private |
| description | String | private |
| voteInfo | VoteInfo | private |
| createTime | String | private |
| image | Image | private |

表2-3-2-2-5 实体类VoteItem

针对每一次投票活动，添加响应的候选名单（人或物），该类封装了候选名单的信息。

1. Image

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | int | private |
| path | String | private |
| item | VoteItem | private |

表2-3-2-2-6 实体类Image

对于每一个候选名单，可以上传一张图片，该实体用于保存对应候选项目的图片名。

1. Email

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | String | private |
| account | String | private |
| password | String | private |

表2-3-2-2-7 实体类Email

1. List1

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | String | private |
| signature | String | private |
| ring\_signature | String | private |
| personal\_signature | String | private |
| publickey | String | private |

表2-3-2-2-8 实体类List1

* Id：字段对应选民的身份序列码
* signature：通过用户的私钥对id进行RSA签名后得到的数据
* ring\_signature:对用户注册信息进行关联环签名后得到的数据
* personal\_signature:通过用户的私钥对消息进行RSA签名后得到的数据
* publickey：用户的公钥

1. List2

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | String | private |
| blind\_vote | String | private |
| Signed\_vote | String | private |

表2-3-2-2-9 实体类List2

* id：用户的身份序列码
* blind\_vote：盲化后的选票数据
* signed\_vote:对选票进行RSA签名后得到的数据

1. List3

| **变量名** | **变量类型** | **变量属性** |
| --- | --- | --- |
| id | String | private |
| voteid | int | private |

表2-3-2-2-10 实体类List3

* id：用户身份序列码
* voteid：对选票消盲后得到的结果，即选票的实际id

**2.3.2.3 Web前端设计**

前端主要使用jsp书写，附加bootstrap以及jQuery库，用于提高界面美观度以及增强用户交互体验。前端主要分为后台管理以及投票中心两部分。

1. 后台

| **名称** | **功能** |
| --- | --- |
| add\_vote\_item.jsp |  |
| add\_vote\_people.jsp | 添加选民 |
| admin\_login.jsp | 后台登录界面 |
| create\_vote\_error.jsp | 创建投票信息失败提示 |
| create\_vote\_info\_success.jsp | 创建投票信息成功提示 |
| create\_vote\_info.jsp | 创建投票信息 |
| create\_vote.jsp |  |
| display\_send\_email.jsp | 显示已发送邮件列表 |
| display\_vote\_info.jsp | 显示已创建的投票信息 |
| display\_vote\_item.jsp | 显示指定投票信息的候选名单 |
| edit\_vote\_info.jsp | 编辑投票信息 |
| header.jsp | 页面头部 |
| main.jsp | 后台首页 |
| menu.jsp | 菜单栏 |
| send\_email\_error.jsp | 发送邮件失败提示 |
| send\_email\_success.jsp | 发送邮件成功提示 |

表2-3-2-3-1 后台实现主要文件

1. 投票中心

| **名称** | **功能** |
| --- | --- |
| display\_vote\_item.jsp | 显示指定投票信息的候选名单 |
| footer.jsp | 页脚 |
| header.jsp | 页面头部内容 |
| login.jsp | 登录界面 |
| menu.jsp | 菜单栏 |
| user\_register\_success.jsp | 用户注册成功提示 |
| user\_register.jsp | 用户注册界面 |
| vote\_center.jsp | 投票中心 |
| vote\_error.jsp | 投票错误提示 |
| vote\_success.jsp | 投票成功提示 |

表2-3-2-3-2 投票中心实现主要文件

**2.3.2.4 Web后台设计**

后台采用SSH（spring，struts2，Hibernate）框架；spring负责事务管理；struts2为控制层，负责处理前端页面提交的数据，并返回相应的结果；Hibernate负责对象的持久化映射。详细实现如下所示：

1. **Action层**
2. AdminLoginAction
3. 所用变量表

| **名称** | **数据类型** | **属性** |
| --- | --- | --- |
| admin | Admin | private |
| result | String | private |

表2-3-2-4-1 实现AdminLoginAction变量

1. 所用函数表

| **函数名** | **参数** | **返回类型** |
| --- | --- | --- |
| AdminLoginAction | void | void |
| getResult | void | String |
| getAdmin | void | Admin |
| setAdmin | Admin | void |
| execute | void | String |

表2-3-2-4-2 实现AdminLoginAction函数

响应管理员登录事件：当管理员登录后台时，此action查询该管理员帐号密码是否存在于数据库中；如果存在则验证通过，进入后台管理页面，反之向前端页面发出错误提示信息，要求用户重新输入账号密码。

1. RegisterAction
2. 所用变量表

| **名称** | **数据类型** | **属性** |
| --- | --- | --- |
| user | User | private |
| info | VoteInfo | private |

表2-3-2-4-3 实现RegisterAction变量

1. 所用函数表

| **函数名** | **参数** | **返回类型** |
| --- | --- | --- |
| RegisterAction | void | void |
| getUser | void | User |
| setUser | User | void |
| getInfo | void | VoteInfo |
| setInfo | VoteInfo | void |
| register | void | String |

表2-3-2-4-4 实现RegisterAction函数

1. LoginAction
2. 所用变量表

| **名称** | **数据类型** | **属性** |
| --- | --- | --- |
| user | User | private |
| uuid | String | private |

表2-3-2-4-5 实现LoginAction变量

1. 所用函数表

| **函数名** | **参数** | **返回类型** |
| --- | --- | --- |
| getUuid | void | String |
| setUuid | String |  |
| getUser | void | User |
| setUser | User | void |
| login | void | String |
| checkUuid | void | boolean |

表2-3-2-4-6 实现LoginAction函数

响应用户登录事件：用户持有唯一的身份序列码（uuid），用户通过此uuid进行登录，此action查询数据表List1中是否存在该uuid，如果存在则登录成功。

1. AddVoteitemAction
2. 所用变量表

| **名称** | **数据类型** | **属性** |
| --- | --- | --- |
| info | VoteInfo | private |
| item | VoteItem | private |
| upload | File | private |
| uploadFileName | String | private |
| uploadContentType | String | private |
| savePath | String | private |

表2-3-2-4-7 实现AddVoteitemAction变量

1. 所用函数表

| **函数名** | **参数** | **返回类型** |
| --- | --- | --- |
| getInfo | void | VoteInfo |
| setInfo | VoteInfo | void |
| getItem | void | Item |
| setItem | Item | void |
| getUpload | void | File |
| setUpload | File | boolean |
| getUPloadFileName | void | String |
| setUPloadFileName | String | void |
| getUploadContentType | void | String |
| setUploadContentType | String | void |
| getSavePath | void | String |
| setSavePath | String | void |
| addVoteItem | void | String |

表2-3-2-4-8 实现AddVoteitemAction函数

添加候选名单：主要负责文件上传以及保存管理员添加的候选名单。通过struts2的标签<s:file/>，将文件保存在属性upload中，uploadFileName保存图片名称，uploadContentType保存文件类型，savePath保存文件存放的路径。

方法addVoteItem()负责文件的保存与数据的持久化。

1. ServiceAction
2. 所用变量表

| **名称** | **数据类型** | **属性** |
| --- | --- | --- |
| info | VoteInfo | private |
| item | VoteItem | private |
| sender | Email | private |
| text | Text | private |

表2-3-2-4-9 实现ServiceAction函数

1. 所用函数表

| **函数名** | **参数** | **返回类型** |
| --- | --- | --- |
| getText | void | Text |
| setText | Text | void |
| getSender | void | Email |
| setSender | Email | void |
| getItem | void | VoteItem |
| setItem | VoteItem | void |
| getInfo | void | VoteInfo |
| setInfo | VoteInfo | void |
| getVoteItem | void | String |
| getEmails | void | String |
| setEmails | String | void |
| createVoteInfo | void | String |
| getVoteInfo | void | String |
| deleteVoteInfo | void | String |
| editVoteInfo | void | String |
| getSendEmail | void | String |

表2-3-2-4-10 实现ServiceAction函数

1. 主要负责的后台业务

| **属性/方法** | **作用/工作** |
| --- | --- |
| info:VoteInfo | 存储投票信息 |
| item:VoteItem | 存储候选项目 |
| sender:Email | 存储发件人邮箱 |
| text:Text | 存储邮件内容 |
| createVoteInfo() | 创建投票 |
| getVoteInfo() | 获取投票信息列表 |
| deleteVoteInfo() | 删除投票信息 |
| editVoteInfo() | 编辑投票信息 |
| getEmail() | 获取数据库存储的邮箱列表 |
| getSendEmail() | 获取一发送的邮件列表 |
| sendEmails() | 发送邮箱 |

表2-3-2-4-11 ServiceAction主要负责的后台业务

1. VoteAction

|  |  |
| --- | --- |
| **属性/方法** | **作用/功能** |
| itemid:int | 候选名单 |
| vote() | 获取用户投票信息，并将处理后的数据存入数据库 |

表2-3-2-4-12 实现VoteAction

1. **DAO（数据访问对象）层**

| **类名** | **功能** |
| --- | --- |
| AdminDao.java | 注册功能操作 |
| Email.java | 邮件功能操作 |
| ListDao.java | 列表功能操作 |
| ReceiverDao.java | 接受功能操作 |
| VoteDao.java | 选举功能操作 |

表2-3-2-4-13 数据访问对象层的实现

该层主要负责数据的CRUD操作，通过Hibernate的SessionFactory生成一个与数据库的session会话，由该session开启一个事务，在该事务内实现相应的增删改查操作。由于各个类的功能大同小异，这里不再多做赘述。

1. **util层（工具类）**

| **类名** | **功能** |
| --- | --- |
| Base64Util.java | 将数据进行Base64编码以及解码工作，为邮件发送提供数据处理服务 |
| DateUtil.java | 获取系统当前时间并进行格式化 |
| EmailUtil.java | 邮件发送工具类，通过jmail提供的smtp服务发送邮件 |
| FileUpload.java | 文件上传工具类，将上传的文件保存到本地。 |
| GenerateUUID.java | 为新注册的用户生成一个uuid |
| HibernateUtil.java | SessionFactory以及Session的创建工具类 |
| TCPUtil.java | TCP客户端工具类，对每一次与C++代码的交互建立一次TCP连接，并发送和接收相应的数据。 |

表2-3-2-4-14 相关工具的实现

**第三章 系统测试**

**3.1 测试目的**

本系统设计并实现了一个基于关联环签名的安全电子投票系统，提供了一套完整的安全投票方案。为保证系统能稳定高效得运行，并且功能实现无误，对系统进行测试。

**3.2 测试环境**

**3.2.1 硬件环境**

服务器的测试环境为台式电脑，具体硬件配置如下：

|  |  |
| --- | --- |
| CPU | 酷睿i5 700 |
| 显卡 | 索泰GTX460 1G首发版 |
| 主板 | 华硕P7P55 LX |
| 内存 | 金士顿12G 内存 |
| 硬盘 | 西部数据1TB 7200转 |
| 网卡 | 主板集成网卡 |

表3-2-1-1 测试PC配置表

**3.2.2 软件环境**

测试的所用工具为Visual Studio2013与MyEclipse-2016-CI-7。

**3.2.3 VS应用开发环境配置**

Crypto++是一个开源的C++加密算法库，它包括密码、消息认证码,单向散列函数,公钥密码机制,关键协议方案,缩小压缩等加密算法。接下来将介绍Crypto++库从下载到项目应用的基本步骤。

**3.2.3.1下载**

1. 登陆官网：https://www.cryptopp.com/
2. 点击“Download”，将显示如下的下载列表，选择“5.6.5 Release”。

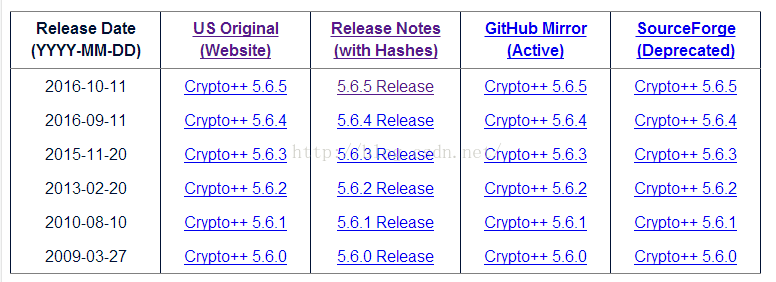


图3-2-3-1-1 Crypto++版本

**3.2.3.2编译**

1. 将下载的压缩文件解压后，找到“\*.sln”文件。 Crypto++是一个跨平台的库，支持“MSVC6.0 - VS2015”、“GCC3.2 - 7”等C++编译器。选择用VS2013进行编译。
2. 直接用VS2013打开“\*.sln”文件，它会提示“升级VC++编译器和库”，确认即可。升级完毕，在VS的解决方案资源管理器中，可以看到该solution下有四个工程，分别是：cryptdll、crypttest、cryptlib和dlltest。其中，crypttest需要依赖cryptlib，而dlltest则是对cryptdll的测试工程。
3. 修改MSVC Runtime Library，Crypto++库支持多线程，上述四个工程，默认的多线程运行时库都是静态版本的，即“MT”或“MTd”，我们需要将它们修改为“MD”或“MDd”。
4. 编译，可以对每个工程进行单独编译，也可以对整个solution进行编译（选择菜单栏上的“build”）。我选择的是编译整个solution，因为单工程编译时，编译cryptdll是会报错，暂时没去查它是为什么。
5. 编译后，会生成一系列文件夹和文件，“Win32->Output->cryptlib.lib”，这个文件就是生成的库文件。事实上，这种方式仅仅只编译了cryptlib和crypttest。

**3.2.3.3部署**

从上面的编译可知，Crypto++库有两种发布方式：一种是DLL（动态链接库），一种是Lib（静态链接库）。我在此先尝试Lib方式。将以一个win32Console工程为例，讲解如何部署Crypto++库。

1. 新建Win32 Console工程，直接用VS2015新建一个名为“AES”的Win32 Console工程，默认即可。
2. 拷贝头文件，Lib方式发布Crypto++库需要在目标工程下包含Crypto++源码文件夹下所有的头文件。事实上，这些头文件还是挺多的，有145个。为了方便管理，我新建一个“Include”文件夹，并编写一个bat脚本来将这些头文件拷贝到“Include”文件夹内。将这个Include文件夹放到目标工程源码目录下，并设置它为包含目录。
3. 拷贝lib文件，在目标工程源码目录下，新建一个“Library”文件夹，将“cryptlib.lib”拷贝进去，并将该文件夹设为“链接器->附加依赖库目录”。
4. 链接库文件，在此，我选择用“#pragma comment(lib,"cryptlib.lib")”预编译指令将它链接进去。也可以在工程文件中显示设置链接库。
5. 包含对应的头文件和应用命令空间。

**3.2.4 Web开发环境搭建**

**3.2.4.1 数据库环境搭建**

本系统采用的是MySQL数据，版本号为5.5

**3.2.4.2 Web前端环境搭建**

Web前端采用jsp(Java Server Page)编写，另外采用jQuery，Bootstrap等框架提高用户交互质量。

**3.2.4.3 Web后台环境搭建**

1. Spring框架环境搭建：本系统使用的是Spring-fremework-3.3.8，引用的jar文件如下

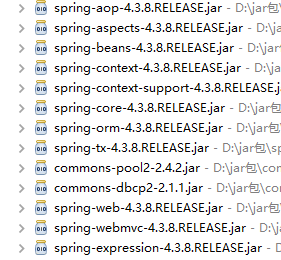


图3-2-4-3-1 引用Spring的jar文件

1. Struts2框架环境搭建：本系统采用的是Struts-2.5.10，所引用的jar文件如下

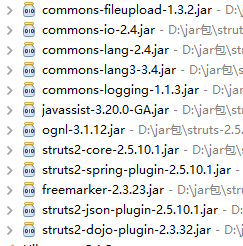


图3-2-4-3-2 引用Struts的jar文件

1. Hibernate框架环境搭建：本系统采用的是Hibernate-5.15，引用的jar文件如下：

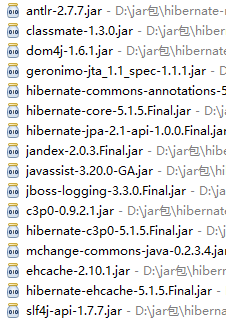


图3-2-4-3-3 引用Heibernate的jar文件

1. JDBC Driver：本系统采用的是数据库驱动为mysql-connector-java-6.0.5

**3.2.4.4 服务器环境搭建**

本系统采用的服务器为：Tomcat-9.0

**3.3 功能测试**

功能测试主要是为了保证系统能正常运行，各个功能能够提供正确的服务。

**3.3.1 公私钥对产生测试**

1. 测试目的

保证产生的公私钥对符合使用的条件并且不重复。

1. 测试用例及结果

|  |  |
| --- | --- |
| 测试用例ID | LR-01-01 |
| 测试用例名称 | 公私钥对正常产生 |
| 测试用例说明 | 根据输入参与投票的人数产生相对的公私钥对 |
| 预置条件 | 通过Socket获得参与投票的人数 |
| 输入 | 参与投票的人数 |
| 预期结果 | 生成对应人数的公私钥对，并打印其中一对 |
| 实际结果 | 如图所示，正常生成五十对公私钥，并打印出其中第三十二对。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-1-1 LR-01-01

|  |  |
| --- | --- |
| 测试用例ID | LR-01-02 |
| 测试用例名称 | 公私钥对无重复情况 |
| 测试用例说明 | 根据输入参与投票的人数产生相对的公私钥对，验证公私钥对中是否存在重复数据。 |
| 预置条件 | 通过Socket获得参与投票的人数 |
| 输入 | 参与投票的人数 |
| 预期结果 | 生成对应人数的公私钥对，并验证是否有数据重复的情况。 |
| 实际结果 | 如图所示，将生成的存放公私钥对的向量先进行去重，然后查看向量的大小，如果向量大小不变，则证明不存在数据重复的情况。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-1-2 LR-01-02

1. **测试描述**

产生的公私钥对满足公私钥对性质，可正常用于之后的关联环签名。

**3.3.2 关联环签名正确签名验证测试**

1. 测试目的

保证关联环签名符合电子投票系统的安全需求。

1. 测试用例及结果

|  |  |
| --- | --- |
| 测试用例ID | LR-02-01 |
| 测试用例名称 | 关联环签名正常运行 |
| 测试用例说明 | 关联环签名在投票人签名后经过投票中心验证为合法投票人 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-1 LR-02-01

|  |  |
| --- | --- |
| 测试用例ID | LR-02-02 |
| 测试用例名称 | 关联环签名签名私钥异常 |
| 测试用例说明 | 关联环签名在投票人签名阶段使用了他人私钥 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为非合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为非合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-2 LR-02-02

|  |  |
| --- | --- |
| 测试用例ID | LR-02-03 |
| 测试用例名称 | 关联环签名验证时公钥异常 |
| 测试用例说明 | 关联环签名在验证阶段疏漏了部分签名人 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为非合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为非合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-3 LR-02-03

|  |  |
| --- | --- |
| 测试用例ID | LR-02-04 |
| 测试用例名称 | 关联环签名验证时公钥异常 |
| 测试用例说明 | 关联环签名在验证阶段公钥环中有数据遭到修改 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为非合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为非合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-4 LR-02-04

|  |  |
| --- | --- |
| 测试用例ID | LR-02-05 |
| 测试用例名称 | 关联环签名验证时参数错误 |
| 测试用例说明 | 关联环签名在验证阶段参数C遭到修改 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为非合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为非合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-5 LR-02-05

|  |  |
| --- | --- |
| 测试用例ID | LR-02-06 |
| 测试用例名称 | 关联环签名验证时参数错误 |
| 测试用例说明 | 关联环签名在验证阶段参数Y遭到修改 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为非合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为非合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-6 LR-02-06

|  |  |
| --- | --- |
| 测试用例ID | LR-02-07 |
| 测试用例名称 | 关联环签名验证时参数错误 |
| 测试用例说明 | 关联环签名在验证阶段参数向量S有值遭到修改 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为非合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为非合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-7 LR-02-07

|  |  |
| --- | --- |
| 测试用例ID | LR-02-08 |
| 测试用例名称 | 关联环签名验证时消息异常 |
| 测试用例说明 | 关联环签名在验证阶段传入的消息遭到修改 |
| 预置条件 | 无 |
| 输入 | 投票人需要签名的消息 |
| 预期结果 | 投票中心验证为非合法投票人 |
| 实际结果 | 如图所示，关联环签名通过投票中心的验证为非合法投票人。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-2-8 LR-02-08

1. 测试描述

关联环签名中，投票中心可以在不知道投票人身份的情况下确认他的合法身份，如果签名验证中出现异常和篡改，则确认为非合法投票人。

**3.3.3 普通RSA数字签名正确签名验证测试**

1. 测试目的

保证普通RSA数字签名符合电子投票系统中数据传输的安全需求。

1. 测试用例及结果

|  |  |
| --- | --- |
| 测试用例ID | LR-03-01 |
| 测试用例名称 | 普通RSA数字签名正常运行 |
| 测试用例说明 | 在消息传递的过程中对消息签名，并在接收消息方进行验证，确认没有被篡改。 |
| 预置条件 | 无 |
| 输入 | 投票人需要传递的消息 |
| 预期结果 | 验证成功 |
| 实际结果 | 如图所示，消息确认未被篡改。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-3-1 LR-03-01

|  |  |
| --- | --- |
| 测试用例ID | LR-03-02 |
| 测试用例名称 | 普通RSA数字签名接收方消息被篡改 |
| 测试用例说明 | 普通RSA数字签名验证阶段的消息与签名时不符 |
| 预置条件 | 无 |
| 输入 | 投票人需要传递的消息 |
| 预期结果 | 验证失败 |
| 实际结果 | 如图所示，验证失败，证明数据被篡改。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-3-2 LR-03-02

|  |  |
| --- | --- |
| 测试用例ID | LR-03-03 |
| 测试用例名称 | 普通RSA数字签名接收方签名异常 |
| 测试用例说明 | 普通RSA数字签名验证阶段的签名发生错误 |
| 预置条件 | 无 |
| 输入 | 投票人需要传递的消息 |
| 预期结果 | 验证失败 |
| 实际结果 | 如图所示，验证失败。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-3-3 LR-03-03

1. 测试描述

为保证系统中信息传递的安全，在信息传递过程中均使用普通RSA数字签名，防止信息被篡改。

**3.3.4 盲签名正确签名验证测试**

1. 测试目的

保证选票盲签名符合电子投票系统中的安全需求。

1. 测试用例及结果

|  |  |
| --- | --- |
| 测试用例ID | LR-04-01 |
| 测试用例名称 | 盲签名过程正常运行 |
| 测试用例说明 | 首先投票人对选票盲化，投票中心对盲化选票签名，之后投票人对选票消盲 |
| 预置条件 | 无 |
| 输入 | 选票信息 |
| 预期结果 | 盲签名成功，选票未被篡改 |
| 实际结果 | 如图所示，盲签名成功，选票前后信息一致，确认未被篡改。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-4-1 LR-04-01

|  |  |
| --- | --- |
| 测试用例ID | LR-04-02 |
| 测试用例名称 | 盲签名过程盲化选票被修改 |
| 测试用例说明 | 在数据传输时盲化选票被篡改 |
| 预置条件 | 无 |
| 输入 | 选票信息 |
| 预期结果 | 盲签名失败 |
| 实际结果 | 如图所示，盲签名失败，同时消盲后可见选票信息前后不一致。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-4-2 LR-04-02

|  |  |
| --- | --- |
| 测试用例ID | LR-04-03 |
| 测试用例名称 | 盲签名过程盲化签名被修改 |
| 测试用例说明 | 在数据传输时签名被修改，投票中心无法确认选票的真实性 |
| 预置条件 | 无 |
| 输入 | 选票信息 |
| 预期结果 | 盲签名失败 |
| 实际结果 | 如图所示，盲签名失败，投票中心验证这不是自己签署的选票。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-4-3 LR-04-03

1. 测试描述

为保证投票人和投票中心都能验证选票信息未被篡改的同时，也能对选票信息保密，使用盲签名操作选票，并确认该过程的安全性。

**3.3.5 各阶段Socket数据传输测试**

1. 测试目的

保证数据传输满足电子投票系统中的正常需求。

1. 测试用例及结果

|  |  |
| --- | --- |
| 测试用例ID | LR-05-01 |
| 测试用例名称 | Socket数据传输过程正常运行 |
| 测试用例说明 | 通过打印Socket过程确保客户端和服务端正常传输数据 |
| 预置条件 | 无 |
| 输入 | 无 |
| 预期结果 | Socket成功建立连接，在数据传输完成后断开。 |
| 实际结果 | 如图所示，系统中Socket的数据传输正常。 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-5-1 LR-05-01

1. 测试描述

投票人和投票中心之间基于Socket的数据传递正常。

**3.3.6 Web后台管理中心测试**

1. 测试目的：

保证后台管理中心正常运行

1. 测试用例及结果

|  |  |
| --- | --- |
| 测试用例ID | LR-06-01 |
| 测试用例名称 | 后台登录测试 |
| 测试用例说明 | 管理员登录后台 |
| 预置条件 | 无 |
| 输入 | 用户名及密码 |
| 预期结果 | 用户名和密码匹配则进入后台，否则给出错误提示信息。 |
| 实际结果 | 1.登录失败页面    2.登录成功后跳转页面 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-6-1 LR-06-01

|  |  |
| --- | --- |
| 测试用例ID | LR-06-02 |
| 测试用例名称 | 后台登录测试 |
| 测试用例说明 | 创建投票信息 |
| 预置条件 | 无 |
| 输入 | 填写投票信息 |
| 预期结果 | 在用户填写所有表单并点击提交按钮之后，进入投票管理页面并显示所创建的投票内容 |
| 实际结果 | 1.创建投票信息    2.点击提交后进入投票管理页面 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-6-2 LR-06-02

|  |  |
| --- | --- |
| 测试用例ID | LR-06-03 |
| 测试用例名称 | 编辑投票信息 |
| 测试用例说明 | 修改已有投票的信息 |
| 预置条件 | 在已有的投票信息基础上进行编辑 |
| 输入 | 修改投票信息 |
| 预期结果 | 在用户填写所有表单并点击提交按钮之后，进入投票管理页面并显示修改的投票内容 |
| 实际结果 | 1.修改投票信息    2.点击提交后进入投票管理页面 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-6-3 LR-06-03

|  |  |
| --- | --- |
| 测试用例ID | LR-06-04 |
| 测试用例名称 | 查看投票信息 |
| 测试用例说明 | 查看投票信息的详细内容 |
| 预置条件 | 在已有的投票信息基础上进行查看 |
| 输入 | 点击“详情”按钮 |
| 预期结果 | 显示投票信息详细内容 |
| 实际结果 | 1.若无数据，给出提示信息    2.若有数据，显示内容 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-6-4 LR-06-04

|  |  |
| --- | --- |
| 测试用例ID | LR-06-05 |
| 测试用例名称 | 添加候选项目 |
| 测试用例说明 | 为投票内容添加候选项目 |
| 预置条件 | 在已有的投票信息基础上进行添加 |
| 输入 | 点击“添加数据”按钮 |
| 预期结果 | 将填写的数据写入数据库，如果有图片，将其相对路径填入数据库，并将文件保存在响应的文件夹下 |
| 实际结果 | 1.填写表单    2.提交后进入候选名单页面  表格显示：    图片显示：    3.数据库中插入相应数据 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-6-5 LR-06-05

|  |  |
| --- | --- |
| 测试用例ID | LR-06-06 |
| 测试用例名称 | 添加选民 |
| 测试用例说明 | 为投票内容添加选民 |
| 预置条件 | 在已有的投票信息基础上进行添加 |
| 输入 | 填写表单  C:\Users\admin\Desktop\QQ图片20170525164105.pngQQ图片20170525164105 |
| 预期结果 | 指定选民收到投票邮件 |
| 实际结果 | 1.邮件发送后，给出提示信息    2.发送成功后，进入投票信息详情页，可以看到已经有一条收件人的信息    3.选民查看邮箱 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-6-6 LR-06-06

|  |  |
| --- | --- |
| 测试用例ID | LR-06-07 |
| 测试用例名称 | 选民注册 |
| 测试用例说明 | 选民根据邮箱提供的链接，前往注册中心注册 |
| 预置条件 | 投票管理中心发送了投票邀请邮件 |
| 输入 | 点击链接 |
| 预期结果 | 将填写的数据写入数据库，如果有图片，将其相对路径填入数据库，并将文件保存在响应的文件夹下 |
| 实际结果 | 1.填写表单    2. 注册成功后提示：并给出UUID（稍后更新）    3.List1中插入相应数据 |
| 结果分析 | 测试通过 ； 测试未通过 🞏 |

表3-3-6-7 **LR-06-07**

**第四章 创新性**

本项目的创新点如下：

1. **投票者可在任意阶段弃票**

投票者在任意阶段弃权，体现在各公告信息表中的投票者数目不相等，并不会影响选举的正常进行。在各个阶段，公告信息表有公示时间，如果用户超出公示时间不进行相应的操作，则会被视为弃权。每一个阶段操作的数据，都基于上一个阶段，因此当投票者没有进行前一个操作，对应的数据不会出现在该阶段的信息表中，下一个阶段只计算出现在了上一个阶段信息表中的投票者的信息则，则投票者在任何阶段的弃票都不会影响下一个阶段。

在注册阶段，用户部分信息会公布在于List1上。信息表包含分配给该投票者的用于标识唯一身份的序列码。在公示时间内，用户不进行注册，则信息表不会出现该用户的序列码。若公示时间结束后，投票者依然没有注册，则被视为自动弃权。只有List1中出现的序列码的投票者才能进行下一个阶段进行投票，因此在注册阶段弃票不会影响后续投票功能。

在投票阶段，根据注册投票信息表List2中会公布序列号、选票、选票的签名。投票结束后公布List2，接受公开验证。如果公示期结束，投票者依然没有投票，则List2中不会有该投票者的序列号及其他相关信息，则被视为中途弃权。只有在List2中出现了序列码的投票者的选票才能在下一个阶段被计算，因此在投票阶段弃权不会影响后续计票。

在计票阶段，投票人将消盲后的选票公布于计票信息表List3中，系统根据List3进行计票。超过计票期限，投票者依然未将选票公布于List3中，则被视为中途弃权。

因此，投票者在任何阶段弃权都不会影响整个系统的工作，则本产品具有稳固性。

1. **不依赖任何第三方和匿名通道，实现公平可验证性**

系统可以有效地避免由第三方机构欺诈行为引起的安全隐患。

从整个系统的角度，项目构建方案中存在投票中心第三方机构，但由于在选举的任意阶段都采用独立的信息公告机制，且分别设置了公示时间，供投票者查询自己的信息是否得到正确公布。与此同时，在所有人的监督下能够及时发现并有效制止第三方机构的不诚实行为。因此任何第三方机构单独或者合谋均无法冒充投票者进行投票，也不能篡改或者删除投票者的投票信息，避免了第三方机构的恶意行为或合谋攻击引起的安全隐患，即系统的安全性不依赖于对第三方机构绝对可信的前提假设。

从投票者角度，首先，保证了投票人的身份合法具有注册资格，进而获得注册机构为其分配的唯一标识身份的序列码Ｎ。其次，在其后的投票和计票环节中，都使用N，且Ｎ分别公布在List2，LIst3中，如投票者重复投票，则会再练个表中出现冗余，从而杜绝投票者重复投票行为。因此，投票者的任何不诚实的投票行为会被及时发现和制止。

因此，每个过程都将接受大家的监督，即可以防止第三方泄密，同时也具有公平可验证性。

1. **提升了关联环签名验证身份的效率，适用于大规模选举**

大部分的环签名方案的长度都与环中用户的数目成线性增长关系，当环中人数众多的时，需要很大的存储空间来存储签名，需要大量的计算来计算环签名。其大计算量成为制约其应用于大规模电子投票的主要因素。与现有方案不同的是，本产品将计算量大的关联环签名应用于注册阶段实现合法投票者的匿名注册，并通过为合法投票者发放一个唯一的临时身份标识Ｎ，并结合自己产生的公私钥对，用于后续投票和计票阶段的匿名身份认证。由于本产品在投票阶段仅使用公私钥对实现普通的个体签名及验证，故有效保证了投票阶段的高效性和实用性，适合于大规模选举。

1. **解决了匿名性，安全性，公平性之间的矛盾**

电子投票系统的安全需求存在内在的矛盾性。

矛盾一：一方面，投票系统需要向投票者提供证据表明前者正确地理解了后者的意图；另一方面，投票系统不能提供给投票者任何票据，使投票者能够向第三方证明自己的投票内容。

矛盾二：一方面，投票系统需要向公众提供证据表明所有的选票都被正确的计数；另一方面，这些证据不能损害投票者的匿名性。

针对矛盾一，本产品在用户生成选票、投票前时，会用友好的交互方式，向用户显示用户的投票内容，让用户进行确定。在投票后，不产生任何票据，则用户不能向第三方证明自己的投票内容。

针对矛盾二，在计票阶段，本年产品的计票模块会公布用户选票信息表List3，包含标识唯一用户的序列码N，消盲后的选票。投票者可以根据List3查看是否有自己序列码，从而判断自己的选票是否被进行了正确计算。同时，其他人无法根据序列码判断每个序列码对应的具体的用户，因此保证了投票者的匿名性。

本系统提出的电子投票系统的安全需求矛盾的解决方案，并且基于此方案构建出了满足匿名性，安全性，公平性的投票系统。

**第五章 总结**

“基于关联环签名的安全电子投票系统”是“四只unicoin”团队历时三个月，通过需求分析，概要设计，详细设计，编码实现和测试的完整软件工程开发的基于web的关联环签名安全电子投票系统。系统可以有效地避免由第三方机构欺诈行为引起的安全隐患。解决电子投票中存在选票碰撞、第三方机构欺诈选民、需借助匿名通信信道发送选票和投票效率低等问题。

整个系统可分为三个部分，投票用户端、网站端、投票中心端。投票用户作为关联环体中的某一成员进行匿名投票表决。网站端作为投票用户和投票中心之间的中枢架起相互信任的桥梁，对整个投票过程中数据的真实有效负责，并在公告牌上明示注册信息表List1，投票信息表List2，计票信息表List3以供相互查对。投票中心端验证每一个投票用户的身份合法性，为合法投票者发放身份序列码Ni，验证盲化选票的合法性，并为盲化选票签名。

因为项目采用的技术都是现在研究领域的最新研究成果，包括我们通过一定的实验得到的技术创新，设计方案中的很多步骤在后期实现方案的过程中都遇到了一定的困难。包括其中在初步的注册阶段如何避免用户重复注册，同时避免用户重复投票，进一步后面的如何解决选票碰撞、以及投票过程效率过低等的一系列问题。我们的小组成员在遇到各种问题之后积极讨论，通过大家的头脑风暴，将所有想到的方法都进行尝试，通过进一步实验选出最优的方法。

**参考文献**

[1]Chen Xiaofeng, Wang Yumin. A secure electronic voting scheme based on anonymous communication channel [ J ].Acta Eletronica Sinica, 2003 , 3 (31):390一393.

[2]Joseph K. Liu, Victor K. Wei, and Duncan S. Wong.Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. Information Security and Privacy, 2004

[3]Joseph K. Liu and Duncan S. Wong.Linkable Ring Signatures: Security Models and New Schemes.Computational Science and Its Applications–ICCSA 2005.

[4]Sherman SM Chow, Victor K Wei, Joseph K Liu, Tsz Hon Yuen.Ring signatures without random oracles. 2006

[5] Lai Jin, Fan Yushun. A new secure and practical electron-is voting scheme[J].Computer Science,2003,1(30):142一145.

[6]Chaum D. Elections with unconditionally-sector ballots and disruption equivalent breaking RSA [C]//EUROCYFT’88. LNCS 330, Berlin: Springer-vedag,1988:177一182.

[7] Ohta K. An electrical voting scheme using a single admin-istrator[C1//1988 Spring National Convention Record. Japan. MICE, 1988: A-294.

[8] 基于Mix-Net的电子投票系统的安全需求矛盾及解决办法，田 莹 李 星 段海新，ISSN 1000．．1239／CN 11．．1777／TP

[9] 基于时间释放加密和数字签名的匿名电子投票方案，徐紫枫 曾 康 周福才，东北大学软件学院 辽宁 沈阳 110819

[10] 基于关联环签名的抗第三方欺诈安全电子投票方案，张文芳，熊丹，王小敏，0258-2724(2015)05-0905-08 DOI：10．3969／j．issn．0258-2724．2015．05．021

[11] 一个安全电子投票系统的研究与设计\_陈航，Huazhong University of Science and Technology，Wuhan, Hubei 430074, P R. China，April, 2006

[12] 一种高安全的电子投票系统的设计与实现\_石培荣，电子科技大学

[13] 基于同态实现多候选人的电子+选举方案的研究

[14] 基于关联环签名的安全电子投票系统

[15] 基于HElib的安全电子投票方案\_王永恒，徐 晨\*，陈经纬，吴文渊，中国科学院 重庆绿色智能技术研究院 自动推理与认知重庆市重点实验室，重庆 400714

[16] 电子选举中的安全技术研究，韩玮，030 332 009，上海交通大学电子信息与电气工程学院

[17] 电子选举研究的挑战和进展\_王思佳，韩 玮，陈克非，上海交通大学计算机科学与工程系，上海 200030，文章编号：**1000**—**3428(2006)15**—**0007**—**03**

[18] 电子投票系统的安全问题\_英文，ChengChunhung,ChengWaiman,WongKamfai，文章编号:1000-3428(1999)特刊-0061--04

[19]安全电子投票系统的设计与实现\_鲍骎骎，中南大学

1. [↑](#endnote-ref-0)
2. [↑](#endnote-ref-1)
3. [↑](#endnote-ref-2)
4. [↑](#endnote-ref-3)