**面向智慧医疗的属性隐私保护**

**和数据可信删除研究**

**作者姓名 程昱婷**

**指导教师姓名、职称 杨力 教授**

**申请学位类别 工学硕士**

**面向智慧医疗的属性隐私保护**

**和数据可信删除研究**

**作者姓名：**程昱婷

**一级学科：**网络空间安全

**二级学科（研究方向）：**网络空间安全

**学位类别：**工学硕士

**指导教师姓名、职称：**杨力 教授

**学　　院：**网络与信息安全学院

**提交日期：**2020年6月

**西安电子科技大学**

**硕士学位论文**

**学　号　 17151213127**

**密　级　 公开**

**学校代码 10701**

**分类号 TP309.7**

By

Cheng Yuting

Supervisor: Yang Li Title: Professor

June 2020

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Cyber Security

**Achieving Privacy-Preserving for Sensitive Attributes and Data Assured Deletion in Smart Healthcare**

**西安电子科技大学**

**学位论文独创性（或创新性）声明**

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

学位论文若有不实之处，本人承担一切法律责任。

本人签名： 日 期：

**西安电子科技大学**

**关于论文使用授权的说明**

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权属于西安电子科技大学。学校有权保留送交论文的复印件，允许查阅、借阅论文；学校可以公布论文的全部或部分内容，允许采用影印、缩印或其它复制手段保存论文。同时本人保证，结合学位论文研究成果完成的论文、发明专利等成果，署名单位为西安电子科技大学。

保密的学位论文在 年解密后适用本授权书。

本人签名： 导师签名：

日 期： 日 期：

摘要

5G、物联网、边缘计算等通信计算技术的发展催生出新的医疗模式——智慧医疗。与传统的医疗模式相比，智慧医疗在减轻医务人员负担的同时，能够为患者提供更精准有效地治疗。为了保证患者数据地安全共享，需要将其加密并外包至不可信第三方服务器供多个合法用户访问。因此，如何在实现细粒度访问控制的同时保证患者隐私成为智慧医疗发展中的一个关键问题。

作为一种支持细粒度访问控制的加密方案，CPABE（Ciphertext-policy Attribute-Based Encryption）被广泛研究。许多研究人员提出适用于不同场景的CPABE变体。然而，现有的CPABE方案不能直接应用于智慧医疗场景。这些方案要么要求策略明文传输，要么只支持部分策略隐藏。前者会导致患者敏感属性信息的完全泄露，后者则会引起攻击者对健康数据的特殊关注。此外，这些方案没有考虑服务器端数据的可信删除。当第三方服务器没有诚实地执行用户的删除请求时，数据拥有者认为已经删除的健康数据可能依然在网络上传播。因此，本文研究一种既支持全策略隐藏又能保证数据可信删除的访问控制方案。主要研究内容如下：

（1）首先，本文提出一种基于PSI（Private Set Intersection）技术的全策略隐藏方案。先为每个策略定义最小授权集合以及查找策略所有最小授权集合的递归算法。对于一个策略来说，至少包含一个最小授权集合的属性集合是一个授权集合。在此基础上，基于PSI技术实现全策略隐藏。在加密阶段，基于多项式插值求出策略的每个最小授权集合对应的密文。同时，计算多个辅助标记向量。在解密前，用户根据策略密文和辅助向量计算授权关系以及密钥密文之间的映射，完成解密测试。

（2）之后，本文提出了一种基于用户权限撤销的数据可信删除方案。先定义一个策略图，将文件删除转化为策略撤销。并为每个访问策略定义一个关键属性集合以及最小关键属性集合选择算法。在此基础上，借助密文更新实现策略撤销。为了确保可信删除，数据拥有者利用Merkle树对服务器的文件删除进行验证。当且仅当还原后的最新密文与原始密文生成的Merkle根节点一致时，用户才认为文件删除成功。

（3）为了实现用户端高效，用户将指数操作借助已有的可验证指数外包方案外包给多个服务器。本文给出的安全证明表明本方案在保证数据和敏感属性机密性的同时还实现了数据可信删除。最后，基于JPBC（Java Pairing Based Cryptography）库编码实现本方案及对比方案中的所有算法。性能分析和实验结果表明，本方案在实现全策略隐藏和可信删除的同时还保证了用户端高效。

**关 键 词**：智慧医疗，全策略隐藏，秘密求集合交集，策略撤销，数据可信删除

ABSTRACT

The technological advances, such as 5G, Internet of Thing(IoT), edge computing and so on, gave birth to the new pattern, Smart Healthcare. Compared with traditional healthcare, it provides patients with more accurate and effective treatment while reducing the burden on nurses and doctors. To ensure data shared securly, data needs to be encrypted and outsourced to untrusted third-party servers. Therefore, how to achieve fine-grained access control while protecting patient privacy has become a key issue in smart healthcare.

As a fine-grained encryption technology, CPABE is widely studied. Researchers have proposed many variants for different scenarios. However, existing CPABE solutions cannot be directly applied to smart healthcare. They either transfer policy in plaintext, or only hide attribute names. The former causes complete disclosure of patients' sensitive attributes, while the latter makes attackers to pay special attention to this type of data. In addition, these schemes also ignore assured deletion of server-side data. When a request for deletion was not executed honestly by the server, the health data which has been deleted by the data owner may still be spread on the network. In this paper, we study an access control scheme that supports full policy hiding and assured data deletion. The main contributions are as follows.

(1) This paper studies a full policy hiding scheme based on PSI. First we define a minimum authorization set for each policy and design a recursive algorithm to find all minimum authorization sets. For an access policy, a user attribute set containing at least one minimum authorization set is authorized. In the encryption phase, the cipher of each minimum authorization set is generated by constructing polynomial interpolation. At the same time, multiple auxiliary vectors are produced. Before decryption, the user calculates the authorization relationship and the mapping between the secret key and ciphers.

(2) After that, we propose an assured data deletion scheme by revoking all users’ access right. We define a policy graph to convert file deletion to policy revocation. A key attribute set and a minimum key attribute set selection algorithm are defined for each access policy. On this basis, policy revocation is constructed with the help of cipher update. To achieve assured deletion, data owners use Merkle trees to verify cipher update. If and only if the Merkle root of the latest cipher is consistent with that generated by the original cipher, the user is convinced that the file has been deleted successfully.

(3) In order to achieve user-side efficiency, users extend exponent operations to multiple servers by using existing verifiable outsourcing schemes. Then all security proofs indicate that ours guarantees the confidentiality of data and attributes while achieving assured deletion. Finally, all algorithms of ours and two comparison schemes are implemented based on the JPBC library. Performance analysis and experimental results show that ours has better efficiency at the user end.

**Keywords**: Smart healthcare, Full policy hiding, PSI, Policy revocation, Assured deletion

插图索引

图2.1 CPABE加密原语 11

图3.1 系统模型 20

图4.1 策略明文时授权关系和映射关系判断说明 26

图4.2 策略明文时授权及映射关系判断举例 27

图4.3 策略密文时授权及映射关系判断说明 27

图4.4 举例说明标记矩阵的生成及使用 32

图5.1 策略图 46

图5.2 简单策略的关键属性集合 46

图5.3 属性撤销和策略隐藏示例 47

图5.4 密文更新验证示例 50

图5.5 可信删除的安全证明思路 51

图6.1 加密计算开销 59

图6.2 解密计算开销 59

图6.3 策略隐藏开销测试 60

图6.4 授权用户的解密测试开销 61

图6.5 授权用户解密测试开销对比 62

图6.6 交集非空的非授权用户的解密测试开销对比 62

图6.7 删除阶段总开销对比 63

表格索引

表1.1 策略隐藏方案总体比较 5

表4.1 最小授权集合选择算法 30

表4.2 属性包含关系判断 33

表4.3 授权关系判断 34

表4.4 可并行的指数操作分析 36

表4.5 过程A 36

表4.6 过程B 37

表5.1 最小关键属性集合选择算法 48

表6.1 计算参数描述 53

表6.2 指数外包前后的计算开销对比 55

表6.3 策略隐藏中的计算开销 55

表6.4 数据删除中的计算开销 56

表6.5 指数外包中的通信开销 56

表6.6 全策略隐藏中的通信开销 57

表6.7 数据删除中的通信开销 57

表6.8 存储开销 58

表6.9 实验参数设置 58

符号对照表

符号 符号名称

 公钥参数

 主密钥参数

 用户属性集合

 属性私钥

 访问结构对应的矩阵

 明文

 密文

 解密私钥

 数据密钥密文

 验证密钥

 访问策略密文

 中间密钥

 属性私钥与密文之间的映射

 最小授权集合标记矩阵

 策略属性集合标记向量

缩略语对照表

缩略语 英文全称 中文对照

AA Authorized Authority 授权机构

ABE Attribute-Based Encryption 属性加密

CPABE Ciphertext-policy Attribute-Based Encryption 密文策略的属性加密

CS Cloud Server 云服务器

DO Data Owner 数据拥有者

DU Data User 数据访问者

ES Edge Server 边缘服务器

GC Garbled Circuit 混淆电路

JPBC Java Pairing Based Cryptography 基于双线性对的密码库

KPABE Key-policy Attribute-Based Encryption 密钥策略的属性加密

LSSS Linear Secret Sharing Scheme 线性秘密共享

OT Obvious Transfer 不经意传输

PSI Private Set Intersection 秘密求集合交集

目录

[摘要 I](#_Toc40436454)

[ABSTRACT III](#_Toc40436455)

[插图索引 V](#_Toc40436456)

[表格索引 VII](#_Toc40436457)

[符号对照表 IX](#_Toc40436458)

[缩略语对照表 XI](#_Toc40436459)

[第一章 绪论 1](#_Toc40436460)

[1.1 研究背景 1](#_Toc40436461)

[1.2 研究现状 3](#_Toc40436462)

[1.2.1 支持策略隐藏的CPABE 3](#_Toc40436463)

[1.2.2 数据可信删除 5](#_Toc40436464)

[1.3 研究内容 6](#_Toc40436465)

[1.4 论文结构安排 7](#_Toc40436466)

[第二章 背景知识及相关技术 9](#_Toc40436467)

[2.1 属性加密机制相关概念 9](#_Toc40436468)

[2.1.1 单调访问结构 9](#_Toc40436469)

[2.1.2 线性秘密共享机制 9](#_Toc40436470)

[2.1.3 判定性并行双线性Diffie-Hellman指数假设 9](#_Toc40436471)

[2.1.4 CPABE加密原语 10](#_Toc40436472)

[2.2 秘密求集合交集的相关计算工具 11](#_Toc40436473)

[2.2.1 伪随机函数 11](#_Toc40436474)

[2.2.2 集合转多项式 11](#_Toc40436475)

[2.2.3 同态加密 12](#_Toc40436476)

[2.3 可验证的指数外包算法 12](#_Toc40436477)

[2.3.1 固定底数可变指数形式的外包算法 13](#_Toc40436478)

[2.3.2 可变底数可变指数形式的外包算法 13](#_Toc40436479)

[2.4 Merkle哈希树 15](#_Toc40436480)

[第三章 数据访问控制方案整体架构 17](#_Toc40436481)

[3.1 设计目标 17](#_Toc40436482)

[3.1.1 安全目标 17](#_Toc40436483)

[3.1.2 功能目标 18](#_Toc40436484)

[3.2 系统模型 19](#_Toc40436485)

[3.3 系统框架 20](#_Toc40436486)

[3.3.1 系统建立 21](#_Toc40436487)

[3.3.2 数据加密 21](#_Toc40436488)

[3.3.3 数据解密 21](#_Toc40436489)

[3.3.4 数据删除 22](#_Toc40436490)

[3.4 安全模型 22](#_Toc40436491)

[3.4.1 基础安全模型 22](#_Toc40436492)

[3.4.2 全策略隐藏的安全模型 23](#_Toc40436493)

[3.4.3 数据可信删除的安全模型 24](#_Toc40436494)

[第四章 基于秘密求集合交集的全策略隐藏方案 25](#_Toc40436495)

[4.1 问题定义 25](#_Toc40436496)

[4.1.1 最小授权集合 25](#_Toc40436497)

[4.1.2 全策略隐藏与PSI 25](#_Toc40436498)

[4.1.3 PSI模型选择 28](#_Toc40436499)

[4.2 全策略隐藏方案设计 28](#_Toc40436500)

[4.2.1 系统建立 29](#_Toc40436501)

[4.2.2 数据加密 29](#_Toc40436502)

[4.2.3 数据解密 33](#_Toc40436503)

[4.2.4 指数外包 35](#_Toc40436504)

[4.3 正确性证明 37](#_Toc40436505)

[4.3.1 解密正确性 38](#_Toc40436506)

[4.3.2 全策略隐藏正确性 38](#_Toc40436507)

[4.4 安全证明 39](#_Toc40436508)

[4.4.1 基础方案的安全性证明 39](#_Toc40436509)

[4.4.2 全策略隐藏的安全性证明 42](#_Toc40436510)

[第五章 基于访问权限撤销的用户数据可信删除方案 45](#_Toc40436511)

[5.1 问题定义 45](#_Toc40436512)

[5.1.1 策略图 45](#_Toc40436513)

[5.1.2 关键属性集合 46](#_Toc40436514)

[5.1.3 属性撤销与策略删除 47](#_Toc40436515)

[5.2 数据可信删除方案设计 48](#_Toc40436516)

[5.2.1 删除密钥生成 48](#_Toc40436517)

[5.2.2 数据删除 49](#_Toc40436518)

[5.2.3 删除验证 50](#_Toc40436519)

[5.3 数据可信删除的安全性证明 50](#_Toc40436520)

[第六章 性能对比和实验分析 53](#_Toc40436521)

[6.1 性能对比 53](#_Toc40436522)

[6.1.1 计算开销 54](#_Toc40436523)

[6.1.2 通信开销 56](#_Toc40436524)

[6.1.3 存储开销 57](#_Toc40436525)

[6.2 实验分析 58](#_Toc40436526)

[6.2.1 实验环境 58](#_Toc40436527)

[6.2.2 实验结果及分析 58](#_Toc40436528)

[第七章 总结与展望 65](#_Toc40436529)

[7.1 研究工作总结 65](#_Toc40436530)

[7.2 研究工作展望 65](#_Toc40436531)

[参考文献 67](#_Toc40436532)

[致谢 71](#_Toc40436533)

[作者简介 73](#_Toc40436534)

# 绪论

## 研究背景

互联网的普及引发了各类网络安全问题，其中以个人信息泄露最为严重。中国互联网络信息中心发布的《第47次中国互联网络发展状况统计报告》显示：截止2020年12月，有21.9%的网民遭遇过个人信息泄露问题。随着匿名通信系统的发展，越来越多的人们使用匿名网络来保护个人隐私。

洋葱路由（The Onion Route, Tor）作为低延迟匿名网络的代表，能够保护通信双方身份信息不被泄露并对通信内容进行加密。虽然Tor的预期用途是保护其用户的个人隐私以及避免通信行为被监视，但是Tor却被广泛用于非法活动。由于Tor的通信关系难以确认、网站服务难以定位、用户行为难以监控等特点，利用Tor来隐藏真实身份并从事网络违法犯罪活动并逃避监管的现象层出不穷，使用Tor搭建的暗网上充斥着毒品交易、武器贩卖、敲诈勒索、网络攻击、传播谣言以及发布反社会言论等非法行为。2020年9月，欧洲和美国警方合作共同开展名为“Disrup Tor”的行动，针对两地的暗网黑市交易，共缴获500公斤毒品以及64支枪械，没收逾650万美元的现金和虚拟货币。尽管热门暗网市场AlphaBay、Hansa、Empire等接连被取缔，但是仍然有成千上万的暗网黑市贩卖非法产品和服务。此外，国内在暗网上的违法事件也是频频出现：2018年9月，某集团旗下连锁酒店超过一亿条客户信息在暗网售卖；2019年4月，江苏南京某单位信息中心管理的南京市1400万余条居民社保数据被非法盗取，并在暗网上兜售；2019年10月，江苏南通网安部门发现高达500万条银行开户、手机卡注册等公民个人信息被出售。截止2021年1月，Tor每天活跃用户数量超过了270万，洋葱服务暗网站点将近20万个。传统的IP地址分析、深度包检测等网络审查技术面对Tor网络已然失效，对使用Tor网络的犯罪行为调查取证困难重重。

对Tor的主要研究工作可大概分为两类：一类是针对Tor的攻击与监管，利用流量分析或者协议漏洞分析等提出去匿名化攻击方法，具体包括流量追踪、溯源、用户行为分析、洋葱服务定位、通信关系确认等，而网站指纹攻击作为一种识别用户访问网站的被动流量分析方法被广泛研究。另一类是Tor通信系统优化以及隐私增强技术，具体包括路由选择算法优化、洋葱服务优化、流量混淆、协议伪装等，网站指纹防御可以抵抗指纹分析以增强隐私保护，也是当前Tor研究的热点。

传统的网站指纹攻击技术假设用户使用Tor每次仅访问一个网站，攻击者可以知道网页加载何时开始和结束，并且可以收集到完整网站流量，这样的攻击场景未免过于理想化。针对上述缺陷，我们放宽网站指纹攻击假设条件，使用户可以在同一时间连续访问多个网站，在这样的场景下，传统单标签页网站指纹攻击准确率明显下降。本文将在现有研究基础上，探索多标签页网站指纹攻击方法，提高多标签页网站指纹攻击准确率，以期达到对接收方去匿名化的效果。尽管目前没有具体的实例表明网站指纹攻击技术会破坏Tor网络的匿名性，但是研究者们也已经提出了许多网站指纹防御方法。因此我们也提出一种新的网站指纹防御方法，不仅可以防御传统网站指纹攻击，也可以防御我们提出的多标签页网站指纹攻击方法，并在安全和开销之间达到平衡，保证匿名网络的合理应用。

## 研究现状

网站指纹攻击实质上是一种分类问题，对于不同的网站，其网页内容如HTML文件、CSS样式、JavaScript脚本等差异明显，虽然用户可以使用VPN、Tor等方式对通信内容进行加密，但是浏览器在加载网页时生成的数据包序列却不尽相同，攻击者可以根据元数据提取指纹特征，并训练分类器，之后窃听用户访问网站产生的流量数据，使用分类器对流量分类，识别出用户所访问的网站。

为了抵抗网站指纹攻击以保护用户隐私，我们需要强大且有效的网站指纹防御系统。由于WEB服务器没有义务提供防御，所以防御系统不建议部署在WEB服务器上，为了改变或者混淆流量模式，许多防御措施部署在Tor浏览器的洋葱代理（Onion Proxy, OP）或者Tor中继(Onion Relays, ORs)节点上。一般的防御系统利用攻击者无法区分加密数据包的真假，通过添加虚拟数据包以及延迟真实数据包混淆网站指纹，达到防御目的，但是上述操作引入了额外的延迟和数据开销，势必会影响用户的使用体验，并且防御需要易于部署，不能超过Tor原始网络可承受的负载，所以针对网站指纹攻击的防御受到了许多限制。

我们分别从单标签页网站指纹攻击、多标签页网站指纹攻击、网络层网站指纹防御、应用层网站指纹防御四个方面进行介绍国内外研究现状。

### 单标签页网站指纹攻击

网站指纹攻击技术早在1998年被Cheng等人提出，并被应用到对HTTPS的隐私威胁分析上，之后这种技术被应用到各种加密通信的场景【Statistical Identification of encrypted, Hintz, Inferring, website fingerprinting and identification using ordered feature sequences】，如网络代理、OpenSSH、VPN等。随着Dingledine等人在2004年提出第二代洋葱匿名网络Tor，此后这种低延迟的匿名通信系统经过不断迭代，成为最为流行的匿名通信系统。Tor使用路径选择算法选出三个中继节点构建加密通信电路，通过将消息填充固定为512字节，经过层层加密并转发，实现了高匿名度，对于网站指纹攻击极具挑战性，所以之后针对Tor的网站指纹攻击不断出现。2009年，Herrmann等人将数据包长度分布特征输入到朴素贝叶斯分类器中进行分类，然而他们的方法攻击准确率却较低。直到2011年Panchenko团队使用支持向量机（Support Vector Machine, SVM）算法，并且在Herrmann提取的特征基础上增加了HTML大小、数据包突发、数据包长度等新的特征，达到了将近55%的准确率，才真正证明了网站指纹攻击去除Tor匿名化的可行性。

Cai等人对SVM建立模型的现有内核均不满意，他通过计算最佳字符串编辑距离（Optimal String Alignment Distance, OSAD）来衡量网站指纹的相似度，并将OSAD替换为SVM的新内核来实现网站指纹攻击，在Tor上的攻击效果明显比之前要好，在100个网站集合上达到了85%的准确率。2013年，Wang和Goldberg提出在Tor信元层面取代TCP层面提取网站指纹，并且使用了新的距离度量标准来测量指纹相似度，他们将攻击准确率提高了90%，并且这种方法可以突破之前的网站指纹防御。Wang又在此基础上扩大特征集合，提取了包括包排序、传入和传出信元数量以及突发数量等将近4000个特征，并且通过权重初始化与调整操作，为不同特征分配不同权重，使用K近邻（k-Nearest Neighbors, k-NN）分类器降低计算成本，使得攻击效果得到了进一步提升。

2016年，Panchenko等人收集了有代表性的数据集，用于在互联网规模上评估网站指纹攻击，他们提出了一种新的分类器CUMUL，在识别率和计算复杂度方面均优于之前的方法。Hayes等人提出了一种结合随机森林和k-NN的网站指纹识别技术，称为k-FP识别技术，该技术使用随机决策森林来创建页面指纹，使用k-NN分类，在开放世界场景中有着与CUMUL相当的精度。Nasr等人将压缩感知技术应用到网站指纹识别上，该技术可以在降低计算开销的同时保持较高的识别结果。其他的工作[A systematic,k-fp, Feature selection]则按照不同的方法对特征重要性进行排序,Wang等人将基准利率纳入精确度测量，并提出了三种精度优化器用于提高网站指纹攻击的精确度。

近年来，深度学习在许多领域（如语音识别、视觉对象识别和对象检测）显示出优于传统机器学习方法。与传统分类器不同，使用了深度学习得分分类器不需要人工特征工程，Remmer等人的研究表明面对不断变化的WEB内容，自动提取的特征更加强大，当应用深度学习技术自动对Tor流量去匿名化时，在100个网站的封闭世界场景中，他们基于卷积神经网络（Convolutional Neural Network, CNN）的网站指纹攻击获得了超过96%的准确率，并且在900个网站集合上仍然有94%的准确率。Sirinam提出了一种优雅的深度指纹（Deep Fingerprinting, DF）网站指纹攻击方法，DF在一维CNN的基础上构建了超过40层，具有负载架构设计的深度卷及神经网络模型，在无防御的情况下达到了98%的准确率，而精度和召回率也优于其他的深度学习方法。并且对于轻量级的防御WTF-PAD和Walkie-Talkie也有非常不错的效果。其他的工作【Abe,Var-CNN,p1-Fp，Triplet net】则探讨了DL在网站指纹攻击方面的实用性。

### 多标签页网站指纹攻击

2014年，Juarez等人对传统网站指纹攻击强烈的假设条件提出质疑，并且对之前的假设提出了批判性评估，他们的研究表明，当前的网站指纹攻击模型通常忽略了某些变量，如用户的浏览习惯、浏览器的版本和位置差异等，而这些因素对攻击有效性产生了显著影响。

Wang和Goldberg等人针对Juarez的质疑，分别提出解决办法，针对不同用户的不同浏览习惯，提出了基于时间间隙和分类的多标签页分割算法，可以成功分割正时间间隔和零时间间隔的多标签页网站数据包序列，然而在寻找负时间间隔的多标签页数据包序列分割点上，效果却不尽理想，仅仅获得了32%的分割准确率。

Xu等人在Wang提取的特征基础上，使用BalanceCascade-XGBoost算法通过欠采样方法解决分割点发现的数据不平衡问题，通过在不同的数据子集上训练弱分类器，然后联合弱分类器构建最终版XGBoost分类器解决分割点发现问题，他们在Tor数据集上的将负时间间隔的多标签页分割点发现准确率提高到了69%，并且在分割后的初始部分数据包序列上使用特征选择算法提取有用特征，在分割时间为6秒的情况下达到了81%的TPR。

Cui等人提出了基于隐马尔科夫的连续数据包序列的分割点发现算法，将Wang对应的零时间间隔场景准确率从63%提高到了80%，他们的分段算法在缺失了前5%的不完整数据包序列上达到了62%的准确率。在重叠的数据包序列上实现了将近70%的准确率。

Gu和Zhuo等人也分别在多标签页场景下做了网站指纹攻击的尝试，虽然证明了网站指纹攻击在实际场景中仍然有效，但是他们却只是在SSH数据集上做了验证，并没有证明他们的算法在Tor数据集上的有效性。

事实上，自Juarez对网站指纹攻击的假设提出批判性评估之后，多标签页网站指纹攻击相关的研究工作并不多，负时间间隔的分割点发现工作任重而道远。

### 网络层网站指纹防御

早期的防御手段是在VPN以及SSH上做流量混淆操作，Liberatone和Levine在数据包级别上探索了不同的填充方案，Wright等人使用流量变形操作使得访问目标网站是模拟产生另外一个网页的数据包序列，然而，无论是填充还是流量变形操作都不能抵抗网站指纹攻击工作【Peeka-Boo, I Still See You，k-FP】。Dyer等人提出的缓冲定长混淆（Buffered Fixed-Length Obfuscation， BuFLO）通过在一定时间段内以固定间隔发送固定大小的数据包来对页面传输的流量模糊处理，使得访问不同网站产生相似的数据流量，该方法可以使得Wang的k-NN分类器准确率从86%降低到10%，使得Hayes的k-FP分类器准确率从91%降低到21%，然而与之对应的是这种防御方法产生了高延迟和高带宽开销，并且BuFLO无法应对网络拥塞，甚至可能将总传输大小暴露出来【CS-BUFLO,2012Touhing From】。 Cai等人在BuFLO的防御基础上进行改进，提出了拥塞敏感型缓冲定长混淆(CS-BuFLO)防御，该防御方法可以改变数据包传输速率。Tamaraw防御系统减少了数据包固定长度，并且将传入和传出数据包以不同的方式进行处理，避免产生了不必要的填充流量。虽然CS-BuFLO防御和Tamaraw防御均做了改进，但还是产生了巨大的开销，导致这些防御系统难以在真实环境中应用。

2016年，Juarez等人提出了WTF-PAD防御，这种防御方法使用令牌系统生成虚拟数据包，并将其在真实流量的时间间隙中进行自适应填充，该方法因其低开销以及明显的防御效果应用到了Tor浏览器8.5.0版本上。然而，Sirinam的DF攻击方法在WTF-PAD数据集上取得了90%的攻击准确率，宣告了WTF-PAD防御的失败，之后Li的工作也证实了这种防御方法的确泄露了有用信息。2020年，Gong等人提出了零延迟防御FRONT和GLUE，分别在特征丰富的数据包序列头部和连续网页访问的停顿时间插入虚拟数据包，虽然他们的防御没有了时间开销，但是与其他防御相比，将网站指纹攻击准确率降低的幅度较小。

基于流量变形的思想，Nithyanand等人提出的Glove防御为目标网站创建一个相似网页的集合，将防御网站流量插入少量数据包，使得分类器无法区分集合中的网站。Supersequence防御与Glove类似，这种防御方法将不同网站的数据流量进行聚类，以创建一个匿名网站集合，提取这些网站中的最短公共超序列构建防御。2017年，Wang等人在总结了先前工作的基础上，提出了一种高效的网站指纹防御方法Walkie-Talkie，该方法使用半双工通信方法，将传入或者传出方向的数据包缓冲下来并以突发的形式发送，并且构建以突发形式的超序列。然而上述方法依赖于每个站点的先验知识，对于经常变化的动态网站，依然有局限性。并且即使Walkie-Talkie有着较低的数据开销和延迟开销，但是面对DF攻击无法实现有效的防御。

Henri等人提出了分割流量的思想，基于此，他们将用户和入口OR之间的流量通过宽带或者蜂窝等不同的网络进行传输，避免恶意互联网服务提供商（Internet Service Provider，ISP）进行关联攻击，然而他们的流量最终会到达同一个OR节点，位于OR节点处的攻击者可以观察到用户产生的完整流量并实施网站指纹攻击。与Henri的做法不同，Cadena和Mitseva等人利用流量拆分的概念提出了TrafficSliver网络曾防御，建立具有多个入口节点的Tor路径，将用户数据包通过多条不同的Tor路径发送和接收，在中间节点上做流量的合并和拆分，使得控制了某一个恶意入口OR的敌手无法观察到完整流量，可以将最先进的攻击准确率由98%以上降低到16%以下。然而他们的方法并非没有缺陷，当用户连接Tor时使用了网络代理，位于网络代理之前的网关或者网络代理本身可以观察到完整流量，以进行网站指纹攻击。

### 应用层网站指纹防御

应用层一般防御不依赖于填充或者延迟数据包。Tor project建议对网站内嵌资源对象发送的管道大小(即并行处理的请求数量)和请求顺序进行随机化，以防御网站指纹攻击，不过之后的研究【k-fp】证明这种方法是无效的，甚至能够提高攻击准确率。Panchenko等人提出在访问WEB页面时通过加载另外一个其他随机的页面以扰乱攻击者，然而这种防御方法性能比较差，并且开销较大。HTTP混淆（HTTP Obfuscation，HTTPOS）对特定的HTTP请求和响应做填充，但是也无法应对更先进的攻击。Cherubin等人提出通过填充WEB对象，以及插入虚拟WEB对象请求来混淆网站流量，但是这种防御仅仅适用于洋葱服务。Cadena等人提出的TrafficSliver应用层防御通过建立多条Tor路径，利用HTTP 的range选项，在每条路径上随机请求WEB对象的字节，使得控制了恶意入口节点的攻击者每次都只能观察到网站流量的少部分，将网站指纹分类器的检测率降低了近50%，但是如先前所述，TrafficSliver应用层防御也仅能抵抗恶意入口OR的攻击，并不适用于proxy+Tor这种场景。事实上我们的实验方案是对TrafficSliver应用层防御的改进。

## 研究内容

针对多标签页网站指纹攻击现有方案的不足，本文研究多标签页网站序列负时间间隔的分割点发现方法，提出一种新的基于CNN+LSTM的多标签页网站指纹攻击技术。同时针对TrafficSliver应用层防御的缺陷，提出一种基于随机化HTTP资源的应用层网站指纹防御方法。

在多标签页网站序列分割点发现问题上，本文首先在前人提取特征的基础上，发现

在多标签页网站指纹攻击部分，本文首先对多标签页网站指纹攻击的问题和以往的网站指纹的不足进行描述，并对网站指纹的实质进行分析。然后提出基于CNN与LSTM并行的多标签页网站指纹攻击方法，使用CNN提取数据包序列局部空间特征，引入LSTM弥补CNN难以学习到的数据包序列的前后依赖关系的缺陷，分别在不完整数据包序列和重叠数据包序列数据集上开展实验，验证攻击方法的有效性。我们的攻击方法在苛刻条件下，攻击效果仍然能够远远超过其他网站指纹攻击方法。

在网站指纹防御部分，本文提出一种轻量级应用层防御方案。防御系统首先将防御网站集合的网站资源对象按照规则放置到不同的OR节点上，并定期调度与更新。用户访问网站时从原始网站获取HTML资源，从HTML获取其他内嵌资源对象如CSS文件、JavaScript文件等，查找其对应在OR节点位置并从该节点请求。我们从客户端、电路、服务端分析随机化请求最优策略，并在请求网站初始阶段请求其他网站HTML文件混淆流量，最终，我们以13%的数据开销和21%的延迟开销将最优秀的网站指纹攻击方法准确率从98%降低到了40%。

## 论文结构安排

本文共分为六章。其中第一章分别从网站指纹攻击和防御两个方面讨论现有方案的不足并给出后续研究内容。第二章介绍相关背景知识、将会用到的多电路创建方法，以及相关的机器学习模型。第三章到第五章描述本文提出的方法细节。第三章介绍多标签页网站分割点发现。第四章讨论多标签页网站指纹攻击的具体细节以及相关实验结果。第五章则给出提出的网站指纹防御方法的详细内容和防御效果。第六章则对本文的研究内容进行总结并给出进一步的研究计划。

# 背景知识及相关技术

本章第一小节介绍Tor的组成以及相关匿名原理，第二小节介绍网站指纹攻击的威胁模型以及封闭世界和开放世界场景，第三小节介绍网站指纹防御常用的思想与技术，多电路构建的方法和评估防御的开销计算，最后我们介绍本文所使用的机器学习相关算法。

## Tor匿名原理

### Tor的组成

令表示一个参与者集合。访问结构是一个包含的所有非空子集的单调集合，即。集合是一个单调集合，对于任意、属于，当时，可得。假设是的一个非空子集，若则称为的可满足集，否则是的不可满足集。

### 线性秘密共享机制

我们认为一个属性集合上的秘密共享方案是线性的，当且仅当以下两个条件同时成立。

（1）每个属性对应的分量组成群上的向量。

（2）上的分量生成矩阵共有行和列。对于所有，存在一个映射函数将第行映射到一个属性。从群上随机选择整数，，，，，并设定向量。根据方案，是秘密的一个共享形成的向量。

假设是用于访问结构的秘密共享方案。令是任意授权集合，让被定义为。然后，存在一系列常数。如果是秘密中的有效部分，那么有。而对于未授权的属性集合，秘密的分量理论上对其是隐藏的。

### 判定性并行双线性Diffie-Hellman指数假设

判定性q-parallel双线性Diffie-Hellman（q-parallel Bilinear Diffie-Hellman Exponent，q-parallel BDHE）指数问题的定义如下。令群、为阶为素数的循环群。令表示双线性映射。随机选择，是群的生成元。之后，根据式计算参数并将其发送给敌手。



判定性q-parallel BDHE假设是指不存在多项式时间的算法以不可忽略的概率区分和。其中，是群中随机选择的元素。本文中使用的安全假设类似于判定性q-parallel BDHE假设。除了的计算过程不一样之外，其余参数选择与判定性q-parallel BDHE假设一致。其中，的具体计算过程如式所示。



### CPABE加密原语

基础的CPABE方案由四个算法构成，如图2.1所示。其中，包括3个随机化算法，1个确定性算法。前者指当输入相同时，输出不一定相同，后者则是输入相同时输出一定相同。在随机化算法中，通常包含一些随机选择操作保证输出的随机性。在CPABE的基础方案中，只有解密算法是确定性算法。

为初始化算法。在系统建立时执行，且在CPABE方案运行期间只执行一次。为系统中的后续操作生成必要的参数。其输入是系统的安全参数，该值越高，系统的安全性越高，但相应的，系统的效率就会降低。是密钥生成算法。作为一个可信权威第三方，授权机构根据初始化算法中生成的参数以及用户提交的属性集合为用户生成解密密钥。用户属性集合被嵌入到解密密钥中。是加密算法。当数据拥有者想要上传数据时为了保证数据隐私需要使用该算法加密数据。在CPABE中，根据数据拥有者定义的策略生成访问结构，并使用其加密数据。在该过程中，数据拥有者不需要预先指定访问者集合，只需根据语义设定策略即可。是解密算法。数据使用者从云端获取密文和策略。首先判断自己是否被授权。如果被授权，数据使用者解密密文。否则，该用户不执行解密算法。即使非授权用户依然想要执行解密，他也无法得到消息的正确明文。



CPABE加密原语

## 秘密求集合交集的相关计算工具

2004年，Freedman等人[[35](#_ENREF_35" \o "Freedman, 2004 #55)]提出了一种秘密求集合交集的技术，即PSI。在该方案中，基于同态和平衡哈希求每个集合的密文并对其进行比较。根据基础方法的不同，PSI技术主要分为三类：基于不经意传输（Obvious Transfer，OT）的PSI技术、基于混淆电路（Garbled Circuit，GC）的PSI技术、基于公钥加密机制的PSI技术。现有的基于公钥加密机制的PSI技术包括：（1）利用单向伪随机函数将集合中的每个元素映射到某个值，并对哈希值进行加密，之后在密文上进行判断；（2）将集合转化为多项式并求其系数，之后通过具有同态性质的加密机制将其加密传输，之后在密文上进行判断。本节主要对基于公钥加密机制的PSI技术中使用到的基本工具如集合转多项式、同态加密、伪随机函数进行描述。

### 伪随机函数

令是有效的、长度保留的、带密钥的函数。如果对所有多项式时间区分器，存在一个可忽略函数，满足式：



则称是一个伪随机函数。其中是均匀随机选择的，并且也是从将比特字符串映射到比特字符串的函数集合中均匀随机选择出来的。其中，区分器是和预言机自由交互的。因此它能够能适应性地问询，并根据前一个接收到的输出来选择下一个输入。注意，只能执行多项式时间数量的问询且对密钥一无所知。

### 集合转多项式

令表示一个大素数有限域，表示系数均取自的的多项式。在PSI系统中，每个参与者拥有一个集合。集合转多项式的前提是参与者原始集合中的元素均可以在有限域中有一个对应的值。参与者拥有一个包含个数的原始集合，其转化后的集合为。对应的多项式表示为式的形式：



其中，为多项式系数。

任何获取到多项式这个系数的参与者均可以求出自己的集合与集合的交集。例如，另一个参与者的转化后的集合包含元素。当已知多项式的所有系数时，可以进行如式所示判断：



0在这里表示有限域中的零元。如果等式左右两边相等，则表示对应的原始数据是与集合交集中元素。否则，该元素不在交集内。依次遍历集合中的元素可以得到最终的交集。

### 同态加密

1978年，Rivest等人[[36](#_ENREF_36" \o "RIVEST, 1978 #64)]首次提出基于同态加密技术保护数据私密性同时保证数据可用性的方法。同态加密的基本思想在于通过加密函数的同态性质，实现在明文和密文上进行相同运算的效果。因此，同态加密方案主要依赖于加密函数的设计。

具有同态性质的加密函数是指对于任意两个明文，满足式运算：



其中，是加密运算，是解密运算，代表密文域上的运算，表示明文域上的运算。当是加法时，则称该同态算法是加同态；当是乘法时，则称该同态算法是乘同态。

## 可验证的指数外包算法

通常，在指数外包中包含三种形式的参数：底数、指数和最终的计算结果。根据最终计算结果是否保密，指数外包可分为两种形式。在本文中，使用的是最终计算结果可公开的形式。而根据指数计算形式的不同分为固定底数可变指数、可变底数可变指数以及可变底数固定指数。第一种形式可表示为，其中，每一个外包指数运算的底数是一个定值，而指数则由客户端自己指定。第二种形式可表示为，其中，每一个外包指数运算的底数和指数都是由客户端自己指定。第三种形式则是，其中，指数固定，底数由客户端指定。在进行指数外包时，根据具体方案中包含的指数形式选择相应的指数外包方案。本文采用的方案中涉及到前两种形式的指数运算。

### 固定底数可变指数形式的外包算法

本文中固定底数可变指数对应的指数运算形式为。Di Crescenzo等人[[37](#_ENREF_37" \o "Di Crescenzo, 2017 #57)]提出了一种可验证的单次指数外包运算。其关键点在于借助离线阶段计算的指数对将单次指数运算转化为指数长度更小的指数运算，从而提高指数计算的效率。该算法的参与者包括客户端和服务器。客户端指想要外包指数运算的设备，通常是可信的。服务器是执行指数外包计算的辅助设备，通常是不可信的。为了确保外包结果的正确性，客户端将在接收到结果后对其进行验证。执行过程共包括两个阶段：离线阶段和计算阶段。其中，表示乘法循环群，和表示安全参数。

假设客户端需要计算的指数对为。在离线阶段，客户端随机选择，并计算两个参数，。在线阶段的计算流程如下。

（1）客户端随机选择整数，并计算，，并将和发送给服务器。

（2）服务器计算，，并将和发送给客户端。

（3）当指数为0时，客户端直接将计算结果设置为。在这里，0和1均指各自群上的零元和幺元。否则，客户端计算。之后，客户端通过判断与是否相等来验证服务器的计算结果。如果相等，表示服务器计算结果正确；否则，就要求服务器重新计算。

### 可变底数可变指数形式的外包算法

2014年，Wang等人[[38](#_ENREF_38" \o "Wang, 2014 #58)]提出了一种通用的支持可变底数可变指数形式的外包算法，其支持的基本形式为：。最终返回多个可变指数运算的乘积。该算法共包含两个阶段，预处理阶段（类似于上述离线阶段）和在线阶段。

作者共提出了两种预处理方法，生成基础预处理表（BPL）和生成标准乘法基础表（SMBL）。前者适用于大质数阶的循环群上的指数外包，为其生成统计上不可区分的指数对，后者则适用于基于椭圆曲线生成的循环群上的指数外包。不管使用哪种离线计算方案，客户端均需要计算两个表：静态表（ST）和动态表（DT）。令表示一个阶为素数的循环群而是其生成元。值得注意的是，上述两个预处理算法均计算的是底数为的指数值。

BPL的计算共包括两个步骤：

（1）ST：客户端随机选择个整数。之后将这些随机选择的数作为指数计算个指数对，并将指数对存储到ST中。

（2）DT：动态表中依然存储指数对，只是与静态表中指数对的计算方式不同。客户端选择一个随机集合并要求。之后，计算。如果不为0则计算，否则丢弃重新选择新的随机集合。在线阶段中每调用一次BPL，客户端选择一个指数对并将其从DT表中删除，之后在适当的时间计算新的指数对来更新DT表。

SMBL的计算也包括两个步骤：

（1）ST：令，为每个客户端计算，并将指数对存储在ST中。实际上，此处只需要做一次指数运算即可，因为每两个相邻的指数值存在如下关系：。

（2）DT：客户端随机选择一个值，并将其第个比特表示为。令集合包含于并只包含使得的下标的值。之后，计算。对于每一次SMBL的调用，客户端随机选择一个指数对并将其从DT中删除，之后在适当的时间计算新的指数对来更新DT表。

在线阶段共需要进行如下三步运算。

（1）客户端调用BPL或者SMBL算法获得四对指数对，，和，要求。选择一个随机值且，其中，是安全参数。令，，从整数群中随机选择多个值即。之后，进行如式计算。



（2）客户端再次调用BPL或者SMBL算法获取指数对，，⋯，。之后，客户端以随机的顺序访问服务器，使其执行式中的计算。



（3）客户端检查等式是否成立。如果成立，则根据式计算最终结果。



## Merkle哈希树

1987年，Merkle R C[[39](#_ENREF_39" \o "Merkle, 1987 #60)]首次提出了一种基于哈希函数的二叉树或多叉树结构，通常称其为Merkle树。每个叶节点代表的值为每个数据块的哈希值，每个非叶结点代表的值则是其孩子节点数据块连接后求得的哈希值。该过程自下而上一直执行，直至求出树的根节点。其中，哈希函数是密码学的基本工具。它将任意长的消息压缩为一个固定长度的摘要，这个摘要通常被称为哈希值。

Merkle树主要用途是确保在对等网络中从其他对等网络接收到的数据块未被损坏和修改，同时检查其他对等网络是否发送假块。比如在分布式环境下，从多台主机获取数据，为了验证接收到的数据未被篡改，接收者只需要验证Merkle树根的哈希值与之前保存的值是否一致。可以说，任何叶节点数值的变化，最终都会传递到根的哈希值上。而当Merkle根的哈希值与之前保存的值不一致时，接收者也可以通过Merkle树快速定位到出现错误的数据块。

除了Merkle树以外，用户还可以直接使用哈希或者哈希链表对接收到的数据进行完整性校验。然而使用直接哈希的验证效率是很低的。当数据源不稳定时，一旦数据损坏，发送方就需要重新传输全部数据。此外，当整个数据块被存储在不同的发送方，直接哈希会导致接收者无法定位出现错误的位置。与直接哈希相比，哈希链表和Merkle树都可以直接定位错误数据块的位置，并且在重新计算Merkle根值时只需要重新传输出现错误的数据块。然而哈希链表在定位错误块时，需要遍历所有数据块，Merkle树则在每一层上都可以过滤掉一半分支。可以看到，与哈希链表相比，Merkle树定位错误数据块的效率较高。

假设数据块总个数为，将一次哈希计算看做一次单位运算，则计算Merkle根值所需的时间复杂度为。当实际场景中需要定位错误数据块位置时，接收者的存储复杂度为，定位出错数据块的时间复杂度为。而当不需要定位错误数据块位置时，接收者的存储复杂度为。在本文方案中，数据拥有者在验证密文更新时不需要定位具体的没有执行更新的密文组件位置，只需要判断密文是否被更新即删除是否完成。因此数据拥有者只需要存储Merkle根值即可，不需要额外存储计算中间值。

# 数据访问控制方案整体架构

本章主要介绍面向智慧医疗场景支持全策略隐藏和数据可信删除的访问控制方案整体架构。总共包括4个部分：方案的设计目标，通过分析智慧医疗中的需求给出方案的安全目标和功能目标；系统模型，介绍系统参与者及其承担的主要功能；系统框架，主要包括系统的所有算法及其输入输出参数；安全模型，给出对敌手能力的假设和方案安全的定义。

## 设计目标

本节从安全性和功能性两个角度出发讨论适用于智慧医疗场景的数据访问控制方案应满足的主要需求。

### 安全目标

一个适用于智慧医疗场景的访问控制方案应该满足以下安全目标：

（1）数据和密钥的机密性。智慧医疗场景中的数据大多是各种传感器收集到的体征数据，如心率、体温、血压等。为了提高医治效果以及实现远程治疗，这些敏感信息需要存储至第三方云服务器中供医生、护士及病人自己随时访问。为了保证患者隐私，这些原始数据在传输至云端前需要先加密。只有拥有合法密钥的用户可以从密文中恢复原始明文。为了保证密钥机密性，这些用户的合法密钥也需要加密存储。否则，任何拿到合法密钥的用户均可以访问原始明文。为了确保数据及加密密钥的机密性，本文采用云计算模型中常用的两层加密机制。首先通过对称加密保证原始明文的机密性，之后使用CPABE技术保证密钥安全。

（2）抵抗用户共谋攻击。在对称加密技术中，不存在用户共谋攻击的危险。因为加密密钥是随机选择的，不同用户的加密密钥可以认为是独立分布的，不存在任何关联。因此，多个不知道加密密钥的用户无法重构正确的加密密钥。但在CPABE技术中，授权机构根据用户的属性生成对应的私钥。当用户使用CPABE技术加密一个密钥后，一个非授权用户一定不能访问正确密钥。但是，当多个非授权用户的属性集合的并集是一个授权集合时，这些用户根据属性集合可以构造一个授权属性集合对应的私钥，从而达到共谋攻击的效果。通常使用两层密钥的方式抵抗用户的共谋攻击，即除了为每个用户生成属性私钥之外还有一个最终的解密密钥，只有同时拥有这两个密钥才能正确解密密文。

（3）用户属性的机密性。与上述相同，对称加密技术中不使用任何公开的用户属性信息，因此不存在用户的属性隐私泄露问题。但在传统的CPABE技术中，为了正确解密，用户的访问策略需要以明文的形式与密文一起存储在云端。然而，智慧医疗中的访问策略通常由用户的隐私属性构成，例如就诊编号、就诊科室等敏感信息。因此，用户除了保证数据和密钥的机密性之外还应保证策略中属性的机密性。本文基于PSI技术实现策略隐藏从而保证用户属性的机密性。

（4）数据可信删除。在智慧医疗场景中，用户存在数据删除需求。删除操作前，要求所有非授权用户无法访问原始数据。删除操作后，则要求任何人无法获取数据明文，包括授权用户和非授权用户。本文的删除假设是用户端不能在本地存储原始明文，并且云端在删除前已经执行去冗余操作，即在执行本文的删除前，整个系统中只存在一份被删除数据，且其存储在云端。这两个假设可以通过存在性证明和去冗余技术完成。由于删除操作由第三方服务器执行，验证服务器的是否正确执行协议规定的删除操作也是数据可信删除的一个重要目标。本文通过用户访问权限撤销实现数据删除，并借助Merkle树对密文删除进行验证，确保数据可信删除。

### 功能目标

一个适用于智慧医疗场景的访问控制方案应该满足以下功能目标：

（1）细粒度的访问控制。在传统的加密方案中，只能支持一对一的访问，数据拥有者在加密数据前就确定合法的数据访问者的身份。然而在智慧医疗场景中，一份数据通常需要被多个数据访问者访问，例如一个患者的病例可被就诊科室的医生和护士合法访问。因此，该场景需要支持细粒度访问控制的加密方案，即数据拥有者在加密前无需知道访问者的具体身份，只需要根据语义设定加密结构，让那些满足语义的访问者可以正确解密。本文采用的基础加密方案CPABE可满足该目标。

（2）用户端高效。显然，在智慧医疗场景中，用户端的采集设备和计算设备大多计算及存储能力有限。同时，为了提高用户的就诊体验和保证用户及时就诊，本场景对实时性要求较高。第二个功能目标就是确保引入的隐私保护机制不为用户端带来较大的计算开销和存储开销。本文借助可验证外包技术及边缘计算模型将用户端较大的计算开销外包给多个不可信的第三方服务器。

（3）支持大域。如上所述，智慧医疗中涉及到的用户数目庞大，这意味着用户属性域也无限增长。因此，直接在系统建立阶段确定系统中属性的具体个数及其对应参数的CPABE方案不能应用于本场景。此外，策略隐藏的实现也依赖于用户属性域的大小。因此，本文除了保证基础CPABE方案支持大属性域之外还应保证引入的策略隐藏机制也能保证大属性域。

（4）准确的解密测试。为了在实现安全目标（3）中的属性隐私的同时保证方案的可用性，本文引入一个新的功能目标，支持准确的解密测试。该功能要求在访问策略全部隐藏的情况下数据访问者依然可以判断自己是否是授权用户。更重要的是，对于授权者而言，他们应更进一步的确定私钥和密文之间的对应关系才能正确解密。

## 系统模型

随着用户端对本地设备存储能力和计算能力需求的增长，云计算，一种按需提供计算资源池的服务应运而生。在该模型中，用户通过电脑、笔记本、手机等移动设备接入系统，按自己的需求借助远程服务器进行运算。然而，随着物联网设备等资源受限设备的普及，云计算模型不再适用于新型计算场景。Shi等人[[40](#_ENREF_40" \o "Shi, 2016 #51)]提出了在靠近数据源头的边缘进行数据处理的计算模型，称为边缘计算。与云计算相比，边缘计算在多源异构数据处理、带宽负载和资源浪费、资源限制以及隐私保护等方面均表现出更强的模型优越性。因此，本文选择边缘计算模型作为智慧医疗场景中的基础模型。

本文提出了一种基于边缘计算模式的智慧医疗模型，如图3.1。该模型中共包含5种参与者：授权机构（Authorized Authority，AA）、边缘服务器（Edge Server，ES）、云服务器（Cloud Server，CS）、数据拥有者（Data Owner，DO）、数据使用者（Data User，DU）。对第三方服务器可信度假设分为两种：可信与半可信。可信指服务器会严格按照协议规定的操作执行且不会与其他恶意用户合谋窃取合法用户的隐私数据。半可信指服务器会按照协议规定的操作执行但会尝试通过已知信息推测用户的敏感数据。为了提高模型的实用性，本文会尽量弱化可信服务器在协议中承担的计算，同时验证半可信服务器生成的计算结果的正确性。

（1）AA。AA是一个可信的服务器。本文假设该服务器不会与恶意用户进行共谋，即不会泄露用户私钥也不会利用已知私钥去解密用户数据。它根据用户的属性集合为用户生成属性私钥和解密私钥。

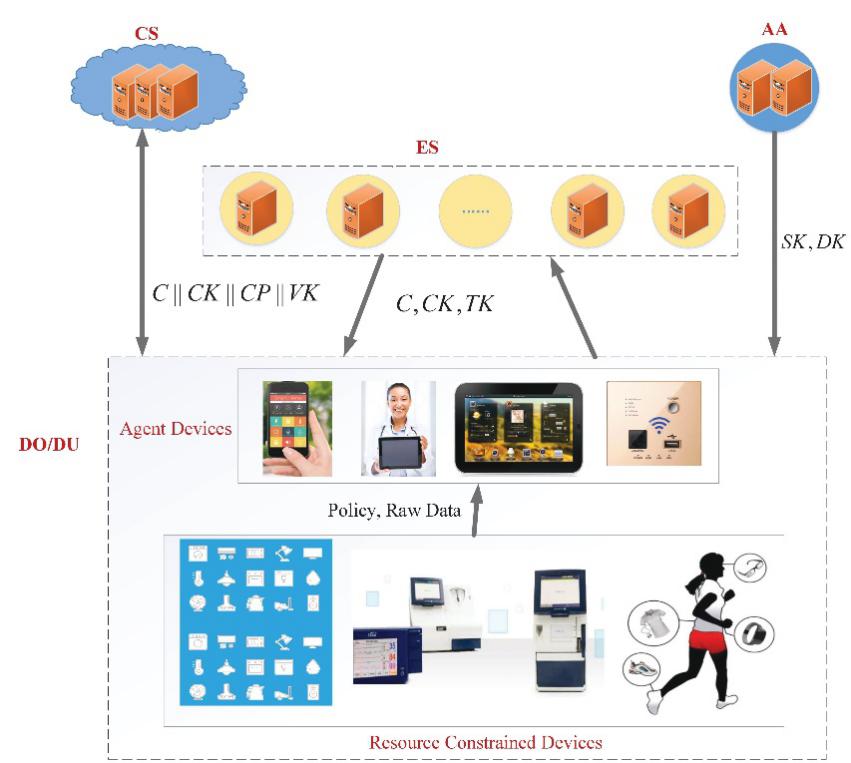
（2）CS。CS是一个半可信的服务器。本模型中，CS负责存储原始数据密文、加密密钥密文以及访问策略密文。在删除阶段，它还负责执行密文更新操作。

（3）ES。ES指靠近用户端的半可信服务器，本模型包含多个ES。它们只承担计算任务不负责存储。为了提高用户端的计算效率，DO/DS将大部分计算外包给ES。此外，当一次计算中有多个互不相关的操作时，一个用户一次可以选择多个边缘服务器进行并行计算来进一步提高用户端的计算效率。当然，在执行并行任务时还应考虑用户端的并行限制。

（4）DO。DO是指需要将电子病历等健康信息加密上传至远程服务器的用户。由于DO上传的是关于自身的健康信息，他们通常被认为是可信的。DO负责对原始数据进行对称加密，之后借助CPABE技术对加密密钥再次加密。为了保证策略中的属性隐私，DO还执行策略隐藏操作。此外，DO对原始数据发起删除请求并对删除结果进行验证。

（5）DU。DU是指访问云端密文的用户。根据CPABE的特性，被授权的DU可以得到正确的解密密钥，而非授权用户则不能。DU可能是恶意的，多个非授权的DU可能会合谋窃取用户隐私信息。DU主要进行解密测试和最终解密。通过解密测试，非授权的DU可直接得知授权关系，而授权的DU获取密文和私钥的对应关系并执行最终解密。

值得注意的是，本文的用户根据计算能力的不同，共分为两种：资源限制型设备以及正常设备。第一种类型的设备指的是传感器等只能收集数据不能处理数据的设备。第二种类型的设备指的是智能手机、笔记本、平板等具有一定存储和计算能力的设备。后者被认为是前者的代理设备。前者负责收集用户的原始健康数据，并将这些数据传输至正常设备中进行计算或者存储。



系统模型

## 系统框架

Qin等人[[41](#_ENREF_41" \o "Qin, 2015 #61)]提出了一种支持外包解密验证的CPABE方案，DU将部分解密计算外包给第三方服务器并对其计算的中间结果的正确性进行验证。但该方案仅讨论了小属性域的情况。在Rouselakis Y和Waters B[[4](#_ENREF_4" \o "Rouselakis, 2013 #62)]给出的大属性域思想的基础上，本文将上述支持外包验证的CPABE方案扩展至大属性域，并在扩展方案上实现支持大域的全策略隐藏和可信数据删除。

支持全策略隐藏和可信删除的数据安全访问控制方案共包含四个阶段：系统建立阶段、数据加密阶段、数据解密阶段、数据删除阶段。

### 系统建立

系统建立阶段的主要任务是生成系统参数和用户私钥，共包含两个算法：系统建立算法、用户私钥生成算法。

1. 。该算法由AA执行。系统建立算法的输入是安全参数，输出为一组公钥参数和主密钥。其中，被公开，被秘密存储。
2. 。该算法依然由AA执行。其输入为、、、用户的属性集合，输出用户的属性私钥和解密私钥。

### 数据加密

在数据加密阶段，DO除了加密数据和密钥以外还加密访问策略，共包含两个算法：数据加密算法、策略隐藏算法。

（1）。该算法由DO执行。其输入为、明文、加密密钥、访问策略对应的访问结构，输出为明文对应的密文、加密密钥密文、验证密钥。是用来验证服务器执行的外包解密计算产生的结果。由DO传输至CS存储。

（2）。该算法由DO执行。其输入为以及明文的访问控制策略。输出访问控制策略对应的密文以及求密钥和密文对应关系所需的辅助信息，最小授权集合标记矩阵和策略属性集合标记向量。

### 数据解密

数据解密阶段共包含三个算法：解密测试算法、中间密钥生成算法、解密算法。其中，当解密测试算法输出结果为false时，用户不再继续执行后续两个算法。

（1）。该算法由DU执行。其输入为、、、、和用户属性集合标记向量。如果是授权集合，则输出和属性私钥与密钥密文的对应关系。否则，输出。

（2）。当算法（1）输出不为才执行该算法，且由DU选择的执行。其输入为、、和。输出为中间密钥。

（3）。与上述算法（2）相同，当算法（1）输出不为才执行该算法，且由DU执行。其输入为、、、、和，输出明文。

### 数据删除

数据删除阶段共包含三个算法：删除密钥生成算法、密文更新算法、验证算法。

（1）。该算法由提起删除请求的DO和ES共同执行。其中，多个ES辅助DO生成加密密钥。其输入为、和，输出删除秘钥。

（2）。该算法由存储加密密钥密文的CS执行。其输入为、，输出更新后的密钥密文。

（3）。该算法由提起删除请求的DO执行。其输入为最新密文、并输出验证结果。

## 安全模型

根据上述安全目标，本节分别给出相应的安全模型。基础安全模型仅针对可验证外包且支持大域的CPABE方案。全策略隐藏和数据可信删除的安全模型则是根据其具体的实现给出对应的安全模型。

### 基础安全模型

本节通过下述挑战者和敌手之间的游戏定义基础安全模型。

初始化阶段。定义即将挑战的访问策略及其对应的访问结构。之后将其发送给。

建立阶段。执行算法生成一系列系统参数。之后，将公钥参数发送给。

阶段1。根据上述建立阶段生成的一系列系统参数响应的请求。首先随机地生成一些属性集合并通过提交任意属性集合获取相应的私钥。其中，下标属于整数集合。为执行并生成密钥对。当是授权集合时，直接返回。否则，返回。

挑战。向提交两个等长的明文和。之后随机选择一个明文，并使用指定的访问结构执行加密算法，得到一个挑战密文和一个验证密钥。最后，将和都发送给。

阶段2。重复执行阶段1中的操作。

猜测。根据已有信息给出选择的明文下标的猜测。

在该游戏中获胜的概率被定义为。

一个支持可验证外包解密的CPABE方案是安全的当且仅当所有的多项式时间的在上述游戏中获胜的概率是可以忽略的，即式成立。



其中，是一个可忽略函数。

### 全策略隐藏的安全模型

在支持全策略隐藏的模型中，共存在两个参与者：和。制定访问策略并且在共享数据的同时想要隐藏访问策略中包含的属性信息。令表示定义的访问策略的属性集合，表示拥有的属性集合。除了想要知道用户属性集合是否被授权外还想要知道和之间的属性对应关系。此外，还尝试通过已知信息推测访问策略中的其他属性内容。

令表示一个策略隐藏协议。令表示计算包含关系的函数。和之间的授权关系可以通过计算的任意最小授权集合和的包含关系求得。令表示的任意最小授权集合，表示中的任意属性，表示只包含的一个集合。两个参与者在不暴露和的前提下共同计算。而对的计算则是通过多次对中的任意属性计算实现的。其中，令，表示参与者需要执行的计算，表示参与者在执行协议所得到的视图，表示参与者在执行所得到的输出。保密的计算了函数等价于保密地计算了函数。

我们说协议保密地计算了函数，当且仅当存在多项式时间的模拟器和使得下式和同时成立。





其中，表示不可区分。

该定义表明任意参与者所得到的视图仅能从执行函数的输入和输出中获得，除此以外，不能得到任何额外信息。

### 数据可信删除的安全模型

本节依然通过挑战者和敌手之间的游戏定义数据可信删除的安全模型。

初始化阶段。定义将要被删除的访问策略及其对应的访问结构。之后将其发送给。

建立阶段。执行算法生成一系列系统参数。之后，挑战者将公钥参数发送给。

阶段1。根据上述建立阶段生成的一系列系统参数响应的请求。首先随机地生成一些属性集合并通过提交任意属性集合获取相应的私钥。其中，下标属于整数集合。为执行并生成密钥对。无论是否是删除策略的授权集合时，都返回。

挑战。向提交两个等长的明文和。之后随机选择一个明文，并使用指定的访问结构执行加密算法以及删除阶段的和算法，得到一个挑战密文和一个验证密钥。最后，将和都发送给。

阶段2。重复执行阶段1中的操作。

猜测。根据已有信息给出选择的明文下标的猜测。

在该游戏中获胜的概率被定义为。

一个CPABE方案能实现数据可信删除当且仅当所有的多项式时间的在上述游戏中获胜的概率是可以忽略的，即式成立。



其中，是一个可忽略函数。

# 基于秘密求集合交集的全策略隐藏方案

本章针对智慧医疗中访问策略公开传输引发的用户健康信息泄露这一问题，提出基于秘密求集合包含关系的全策略隐藏方案。同时为了减轻全策略隐藏方案给病人及医生带来的计算开销，引入可验证指数外包方案。在保证访问策略中用户健康隐私的同时确保用户诊疗过程的实时性。

## 问题定义

### 最小授权集合

为了便于描述，首先给出一些定义。集合表示一个属性集合，用户访问策略表示一组属性的逻辑组合，通常以布尔表达式的形式给出。一个访问策略可以有多个授权集合。一个属性集合是访问策略的一个最小授权集合当且仅当该集合是一个授权集合且该集合的所有真子集都是非授权的。当然，一个访问策略也可以有多个最小授权集合，这取决与构成访问策略的逻辑门的类型。例如，策略为，策略为。那么，只有一个最小授权集合，而有两个最小授权集合，分别是和。

令集合为一个访问策略的所有最小授权集合构成的一个新集合。那么，一个属性集合是该访问策略的一个授权集合当且仅当该属性集合至少包含一个的最小授权集合，即。对于属性加密中的数据访问者来说，当且仅当该用户属性集合至少包含一个访问策略的最小授权集合时，该用户是当前访问策略的一个授权用户。

### 全策略隐藏与PSI

为了保护患者的属性隐私和数据安全，本文想要设计一种支持大域和全策略隐藏的CPABE方案，也就是说在加密数据后，不直接将访问策略以明文的形式存储，而是将属性名和属性值全部隐藏。但在实现方案的过程中发现设计实现支持大域和全策略隐藏的CPABE存在如下挑战。

（1）不能引入类似哈希等需要为属性统一编码的函数。在基础的CPABE方案中支持大域意味着在系统参数生成时，不能预先为所有属性生成相关参数。除此以外，在设计全策略隐藏功能时也要注意支持大域的问题。例如，在实现全策略隐藏时，最简单的实现是使用哈希函数或者伪随机函数将系统中所有属性映射到不同的值上。直接使用映射后的值生成混淆后的访问策略。用户在解密时，直接通过公开的哈希函数判断授权关系。然而，在支持大域的情况下，引入的统一编码函数应该具有非常强的抗碰撞性并且其支持的输入和输出范围应该足够大。很显然，这个假设是不现实的。因此，本文认为在实现全策略隐藏时除了不能引入一些需要预先为所有属性生成相关参数的功能也不能引入类似哈希等需要为所有属性统一编码的函数。

（2）无法确保用户属性私钥和密文中访问策略之间的属性对应关系，导致授权用户无法正确解密。在基于LSSS的CPABE方案中，全策略隐藏除了隐藏访问控制策略外还应隐藏访问结构中的映射函数。映射函数是将访问结构的某一行映射到策略中的某一属性，其承担的主要作用是在解密时确定密文组件和用户属性私钥组件之间的匹配关系。因此，的隐藏会导致授权用户不能正确解密数据。

（3）引入的全策略隐藏方法必须在一次单向通信内完成。CPABE适用于云计算模型的一个很重要的优点是，数据拥有者只需要对共享数据执行一次加密，之后多个用户可以根据已有私钥进行解密。也就是说，在DO发送数据到授权的DU获取数据的整个过程中，一个数据拥有者和一个数据访问者之间仅发生一次单向通信。因此，引入的全策略隐藏方法也应在单向一次通信内完成。也就是说，DO加密访问策略到DU获取授权关系这个过程仅发生了一次单向通信。

由于PSI技术支持在参与者属性集合保密的情况下判断集合交集，本文拟采用PSI技术实现全策略隐藏。首先，通过图4.1和图4.2分别从理论上和计算上表示了在策略明文传输的情况下用户是如何实现解密的。可以看到，共包含两个主要判断：（1）授权关系的判断；（2）属性私钥和密文组件的映射关系的判断。为了在全策略隐藏的情况下实现上述（1）的判断，本文基于PSI技术计算用户属性集合和访问策略的每个最小授权集合的包含关系。其判断依据为，当用户的属性集合包含一个访问策略的最小授权集合时，该用户是授权用户。此外，对于（2）的计算，本文通过在加密策略后和解密前计算多种标记向量实现映射关系的判断。



策略明文时授权关系和映射关系判断说明



策略明文时授权及映射关系判断举例

图4.3给出了在全策略隐藏的情况下基于PSI技术进行上述两种判断的过程。首先，DO定义加密数据的访问策略并求出该策略对应的所有最小授权集合。之后，DU利用PSI技术计算用户属性集合是否包含一个最小授权集合。如果是，则说明DU被授权。否则，DU不执行后续计算。为了正确计算映射关系，本文假设属性之间存在一种排序关系。与此同时，在进行PSI计算时，DO和DU为每个最小授权集合和映射函数生成一些标记向量。借助标记向量，DU在计算授权关系后可以得到两个信息：（1）它包含的那个最小授权集合中的每个属性在排序后的策略属性集合中的位置；（2）最小授权集合在排序后的用户属性集合中的位置。根据上述信息，DU可以直接得出用户属性集合和策略属性集合之间的映射关系。



策略密文时授权及映射关系判断说明

### PSI模型选择

根据敌手能力的不同，PSI技术主要分为三类：半诚实模型、恶意模型以及隐蔽敌手模型。在半诚实模型中，系统的参与者将认真按照协议规定的操作执行，但每个参与者想要根据计算中获取的信息推测其他参与者除了交集以外的其他信息。在恶意模型中，系统的参与者可能拒绝按照协议执行计算过程，例如修改输入信息、不执行任何操作等。在隐蔽敌手模型中，系统会引入对参与者不诚实行为的检测与惩罚措施，因此恶意参与者会将恶意行为隐藏在正常行为中去窃取其他参与者的信息。本文中执行PSI的两个参与者分别为数据拥有者、数据访问者。前者的目的是在保证数据共享的同时隐藏策略中的属性，而修改输入内容或不按照协议执行均与其目的相违背。后者则是想要获取解密所必须的授权关系以及私钥和密文的映射。如果不按照协议执行或随意修改输入内容，会导致该用户无法正常解密。因此，本文使用基于半诚实模型的PSI技术。

根据是否引入第三方计算服务器，PSI技术分为两类：传统的PSI技术和借助外包第三方的PSI技术。前者的参与者只有需要计算交集的客户端。而后者则引入额外的服务器辅助客户端进行计算。后者更适用于本文使用的边缘计算模型。

根据基础方法的不同，PSI技术主要分为三类：基于OT的PSI技术、基于GC的PSI技术、基于公钥加密机制的PSI技术。前两种并不适用于实现CPABE中的策略隐藏。其一，OT技术和GC技术仅通过单向一次通信很难完成求集合交集。其二，OT和GC技术为客户端带来许多开销较大的操作，尤其是OT技术。因此，本文采用第三种方法实现策略隐藏。

综上所述，为了实现CPABE上的全策略隐藏，本文选择的是适用于边缘计算模型的基于公钥加密机制的两方PSI技术。

## 全策略隐藏方案设计

支持全策略隐藏的CPABE方案共包含三个阶段：系统建立、数据加密、数据解密。其中数据加密阶段包含策略隐藏函数，数据解密阶段包含解密测试功能。为了实现用户端的高效计算，本文引入可验证指数外包将上述三个阶段中用户端的指数计算外包给多个服务器。

在正式描述方案之前，首先给出下文使用到的一些参数。令和表示阶为素数的乘法循环群。令表示群到上的双线性映射。将全体属性集合定义到整数群上。整数群上的每一个元素代表一个属性。同时，为了确定私钥和密文之间的映射关系，本文定义了一个属性排序函数。其输入为一个任意属性集合，输出为排序后的属性集合。

### 系统建立

在系统建立阶段，执行算法生成一系列后续计算所需的参数。首先，从中随机选择一个指数。之后，设置使用群上的随机数计算公开参数并令秘密参数。最后，该算法输出主密钥以及公开密钥。其中，是不同的哈希函数，可以被任何合法用户访问到。

在系统主要参数生成后，每当一个合法用户请求密钥时，执行算法为合法用户生成密钥对。首先，随机选择个整数。之后计算密钥对。其具体计算过程如式所示。其中，表示用户的一个属性，是群上的元素。



最后，将生成的密钥对发送给对应用户。

### 数据加密

在加密部分，DO首先从群中选择一个随机数，将其作为哈希函数的输入生成，并将作为对称加密机制的密钥加密数据明文得到数据密文。之后，DO使用CPABE机制加密。根据指定的访问控制策略生成访问结构以及映射函数。为了生成密钥密文，DO选择个随机整数，并设置向量。之后，利用行向量计算加密所需的参数，其中。最后，DO根据式子所示计算密钥密文。



此外，为了验证外包解密，DO在加密阶段也生成一些额外参数。计算密钥标签以及验证密钥。

为了保证患者敏感属性的机密性，DO在加密阶段也对访问策略进行加密。本文采用的是基于PSI技术的全策略隐藏方案。众所周知，一个访问策略就是一个布尔表达式，可以被表示成二叉树的形式。在二叉树中，所有的属性结点都是叶子结点，而非叶结点则表示与门和或门。本文提出一个以访问策略对应的二叉树作为输入求其对应的所有最小授权集合的递归算法。

最小授权集合选择算法

|  |
| --- |
| 选择最小授权集合 |
| 输入：访问策略对应的二叉树的根节点；  输出：包含所有最小授权集合的集合；  **procedure**  **if**  **then**  **return**  **else if**  **then**  **return**  **else**        **return**  **end if**  **end procedure** |
| 求两个集合中任意两个元素的并集 |
| 输入：左右孩子结点各自的所有最小授权集合的集合和；  输出：当前结点为“&”门时，对应的所有最小授权集合的集合；  **procedure**    **for**  **do**  **for**  **do**    **end for**  **end for**  **return**  **end procedure** |

表4.1 最小授权集合选择算法（续）

|  |
| --- |
| 求两个集合的并集 |
| 输入：左右孩子结点各自的所有最小授权集合的集合和  输出：当前结点为“|”门时，对应的所有最小授权集合的集合  **procedure**    **for**  **do**    **end for**  **for**  **do**    **end for**  **return**  **end procedure** |

表4.1给出算法的所有步骤。令表示一个集合，该集合包含当前结点对应访问策略的所有最小授权集合。根据节点内容的不同，算法的输入二叉树中结点可分为三种：与门结点、或门结点和属性结点。该算法从属性结点开始，递归的计算每个结点的，根节点的即是访问策略的。当结点值是与门，即“&”时，它的是由左孩子结点的中任意元素和右孩子结点的中任意元素求并集得到的，该计算过程由算法表示。而当结点值是或门，即“|”时，它的则是和的并集，该计算过程则由表示。特别的是，对于没有左右孩子的属性结点来说，其仅包含那个属性节点自身构成的集合。

当求出该访问策略的所有最小授权集合以后，DO对每个最小授权集合进行加密。令。对于，计算的密文。令。第一步，计算对应的多项式的各项系数。根据包含的属性值将表示为点表达式的形式；第二步，根据式中的通用多项式公式求出各项系数；



第三步，利用公开参数将该函数加密，即对各项系数加密。加密方式是将各项系数分别作为指数，群上的生成元作为底数。为每个最小授权集合执行上述三步操作求出对应的多项式函数的密文。最后，访问策略的密文可表示为式所示的形式：



值得注意的是，中并不包含系数的密文。因为不论是什么样的多项式，它的系数总是1。还应包含最小授权集合的阶，在解密测试时使用。

然而，根据访问策略的所有最小授权集合的密文，数据访问者只能求出自己的属性集合与每个最小授权集合的交集。也就是说，数据访问者在根据上述信息只能得到授权关系，而对解密所需的密钥和密文对应关系一无所知。因此，在计算策略密文时，数据访问者还需计算两个标记值，标记行向量和标记矩阵。标记行向量表示的是访问结构中的每个属性对应的行与排序后的属性位置之间的关系。令表示排序后的访问策略的属性集合。对于，的值为。标记矩阵中的每一行表示一个最小授权集合与之间的关系。令每一行表示为，则表示属性在最小授权集合中是否存在，用0,1表示。其中，1表示存在，而0表示不存在。最后，DO将计算到的所有密文和辅助信息传输至CS端。

为了更清晰地表示访问策略密文中标记值的计算，本文在这里给出一个简单的例子。要加密的访问策略为。为了描述方便，假设排序后的属性集合为。在实际计算中，排序后的属性集合与系统给出的函数有关。根据算法求出的所有最小授权集合为。映射函数表示的属性与访问结构中行的关系为。之后，标记行向量和标记矩阵的计算过程如图4.4所示。



举例说明标记矩阵的生成及使用

### 数据解密

在全策略隐藏中，DU解密前需要先执行进行授权关系和映射关系判断。在进行上述判断两个之前，令DU排序后的属性集合为。根据排序前后属性顺序的变化，DU可以得到一个标记行向量。对于，表示排序后第个属性在排序前的集合中的位置，也就是属性对应的密钥组件下标。首先，求DU的授权关系。DU从CS中得到策略密文和辅助信息。DU执行表4.2和表4.3中的算法判断用户集合是否包含一个最小授权集合并通过已知的辅助信息计算映射关系。

属性包含关系判断

|  |
| --- |
| 属性是否属于最小授权集合 |
| 输入：的密文、属性、生成元；  输出：若属于，输出；若不属于，输出；  **procedure**  **for**  **do**    **end for**    **if**  **then**  **return true**  **else**  **return false**  **end if**  **end procedure** |

在算法中，DU判断一个用户属性是否被包含在访问策略的一个最小授权集合中。将属性作为多项式函数的输入，如果该属性使得函数值为0说明属性属于对应的最小授权集合。由于在数据加密阶段被加密存储，DU不能直接计算的值。根据加密阶段使用的指数计算的同态性质，将的计算转化到指数上，从求的值变为求。算法是判断属性集合是否能包含一个最小授权集合。遍历所有的最小授权集合，直到找到一个能被包含的最小授权集合或者发现没有这样的集合。此外，算法还计算一个辅助向量和最小授权集合对应在矩阵中的行数，记录排序后的属性集合中的每个属性是否属于最小授权集合。如果不存在这样的最小授权集合，说明DU是非授权用户，DU将不再执行之后的所有计算。否则，DU将利用得到的进行之后的映射关系判断。DU根据和的第行可以知道排序后的属性集合和排序后的访问策略的属性集合的对应关系。之后，再借助向量和向量便可求得映射关系。

授权关系判断

|  |
| --- |
| 属性集合是否被授权 |
| 输入：访问策略密文、DU的属性集合、生成元；  输出：若是，则输出，否则输出；  **procedure**  **for**  **do**    **for**  **do**  **if**  **then**    **end if**  **end for**  **if**  **then**  **return**  **end if**  **end for**  **return**  **end procedure** |

与加密阶段一样，为了更清楚描述映射关系的计算，本文在上文例子的基础上给出解密测试的举例，如图4.4的右半部分所示。令用户的属性集合为，排序后的属性集合为。根据上述定义，向量为。根据表4.2和4.3中的算法，可以知道自己是授权用户，并且对应的的值为以及所包含的最小授权集合在矩阵中的行位置。根据，可知属性集合包含一个最小授权集合。并且根据向量，可知访问矩阵中的第0行对应属性，第2行对应属性。再根据向量，可得原始属性集合中的第0个属性对应访问矩阵中的第0行，第1个属性对应访问矩阵中的第2行。至此，得到用户的私钥和访问策略中密文组件之间的映射关系，如式所示：



在得到授权关系之后，DU选择一个服务器执行解密。根据得到的进行如所示的计算：



得到中间密钥并将其发送给DU。

DU得到中间密钥之后，利用本地的求加密密钥。先计算。在得到之后，DU需要验证的计算结果。当值与验证密钥相等时，DU才执行最终解密。否则，重新选择一个并对计算错误的进行标记，可根据系统规定对不能正确执行外包解密的执行惩罚措施。在最终解密中，DU利用加密密钥解密对称密文得到明文。

### 指数外包

根据上述方案可以看到，本文提出的支持全策略隐藏的CPABE虽然保证了患者的数据和敏感属性的机密性，但是用户端需要执行很多次指数操作。尤其是在策略隐藏和解密测试阶段。因此，为了减轻患者的加密效率以及医生和护士的访问效率，本文将上述方案中用户端的指数操作外包给多个边缘服务器。

分析发现，上述策略隐藏方案在算法、、以及中有需要DO/DU执行的指数运算。总结下来共有两种形式：、和。特别的是，这两种指数计算中保密形式均是指数保密、底数公开、计算结果公开。在和中，密文计算需要执行形如和的指数加密，最终解密需要执行形如的指数运算。其中，是第二种指数计算形式的特殊情况。在中，计算多项式密文时执行多次形如的指数加密。在中，DU则需要计算的值判断属性与多项式的关系。本文改进第二章介绍的两种指数算法辅助用户端进行可验证指数外包计算。主要的改进部分在于将两个不相关的指数外包计算的离线阶段合并到一起。此外，基于本文中使用的边缘计算模型以及本文中指数计算的特殊情况将多个指数运算变为并行计算。

改进后的指数外包方案共包含两个阶段：离线阶段和在线阶段。在本文的系统模型中，包括两种用户，资源受限型设备和正常设备。而后者通常是前者的代理计算设备。因此在执行指数计算外包时，正常设备被认为是客户端，边缘服务器ES被认为是服务器。也就是说，离线阶段中所需要的计算和存储均在正常设备上执行。

离线阶段：客户端随机选择个整数。之后将这些随机选择的数作为指数计算个指数对，并将指数对存储起来以备后续使用。在线阶段中每选择一对指数对，客户端将将其从存储域中删除，之后在适当的时间计算新的指数对。

在线阶段：根据所需计算的指数计算形式的不同，客户端选择以下两个过程中的一个进行计算。令客户端想要计算的指数值共有个。令客户端支持的最大并行数为。计算可得，客户端需要执行的指数外包次数为。也就是说，客户端选择个同时执行计算。在本文中，不同的指数计算并行执行是可行的。因为在上述多个存在指数计算的算法中，这些指数计算之间不存在计算上的前后关系，互相参数之间也没有关联。在表4.4中给出每个算法中可并行执行的指数计算。在客户端想要执行的是第二种形式的指数计算时，令表示该指数计算中因子的个数。

可并行的指数操作分析

|  |  |  |
| --- | --- | --- |
| 算法 | 指数形式 | 指数计算 |
|  | 一 |  |
| 二 |  |
|  | 一 |  |
|  | 二 |  |

过程A

|  |  |  |
| --- | --- | --- |
| **步骤** | **设备** | **计算** |
| 1 | 客户端 | 客户端选择个服务器，并从整数群上随机选择个整数，并对每个计算两个值和。其中，，，为指数计算中的指数。之后，客户端将每对和发送给不同的服务器。 |
| 2 | 服务器 | 每个服务器接收到客户端发送的数据后，计算，，并将和发送给客户端。 |

表4.5 过程A（续）

|  |  |  |
| --- | --- | --- |
| **步骤** | **设备** | **计算** |
| 3 | 客户端 | 接收到数据之后，客户端执行如下判断。当指数为0时，客户端直接将计算结果设置为。在这里，0和1均指各自群上的零元和幺元。否则，客户端计算。之后，客户端通过判断与 是否相等来验证服务器的计算结果。如果相等，表示服务器计算结果正确；否则，就重新选择服务器进行计算并对计算错误的服务器执行一定的惩罚措施。 |

过程B

|  |  |  |
| --- | --- | --- |
| **步骤** | **设备** | **计算** |
| 1 | 客户端 | 客户端选择个服务器，并从离线阶段所计算的指数对中选择组指数对。之后，并行执行下述过程，并行数为。选择一个随机值，并令。其中，是设定好的安全参数。令，从整数群中随机选择个整数，即。之后的计算如2.3.2中的式。 |
| 2 | 客户端 | 客户端再次从离线计算计算的指数对中选择，，。同样，该过程也需并行执行。之后，客户端将组值分别发送给服务器。要发送一组数据为，，，以及，，。 |
| 3 | 服务器 | 接收到数据之后，每个服务器均进行如2.3.2中的式的计算。计算结束后，将数据发回客户端。最后，每个服务器将计算得到的，，和 发送回客户端。 |
| 4 | 客户端 | 客户端检查等式是否成立。如果成立，则执行步骤5。否则，客户端重新选择服务器对计算错误的指数值重新计算。 |
| 5 | 客户端 | 客户端执行计算求得最终的指数值。 |

## 正确性证明

上述方案中的有三种计算需要进行正确性证明，包括解密计算、全策略隐藏以及指数外包计算。由于指数外包计算只是在计算模型和离线阶段的指数计算上有所改变，在线阶段的计算与参考算法基本一致。因此本节只给出前两种计算的正确性证明。

### 解密正确性

在解密阶段，授权用户在解密测试外还需执行两个解密算法。一个是，利用ES生成中间密钥；一个是，DU执行最终解密。在中，用户先执行式：



获取分子的值。之后，执行式计算分母。



最后，使用式正确求出密钥种子。





### 全策略隐藏正确性

在本文中，全策略隐藏的正确性包括授权关系和映射关系的计算正确性。映射关系的正确性在上节的举例中可以被证明。授权关系的正确性在于对用户属性集合和一个最小授权集合的包含关系的正确判断。包含关系的判断则是遍历用户属性集合并求每个属性是否在最小授权集合上。因此，全策略隐藏中授权关系的计算正确性在于用户属性存在性的判断。

假设访问策略中一个最小授权集合中包含个属性，对应的多项式函数为。其密文形式为。当存在一个用户属性，DU判断等式是否成立来计算属性是否在最小授权集合内。其中该等式可以被化简为。当属性在集合中时，；否则。可以说，对属性存在性的判断就是对多项式的解的判断。

## 安全证明

本节对第三章给出的安全模型中的前两个进行安全证明。第一个是对可验证外包且支持大域的CPABE方案的安全性证明。第二个则是基于PSI技术的全策略隐藏的安全性证明。

### 基础方案的安全性证明

如果假设成立，则所有具有大小为（其中）的挑战矩阵的PPT对手在选择性地破坏基础CPABE方案时具有可忽略的优势。

具体的证明如下。

初始化：敌手将挑战矩阵发送给挑战者。其中矩阵是一个大小的矩阵。其中并且。

建立：挑战者将系统公开参数发送给敌手。之后将主密钥设置为。其中，和在假设阶段设置而是由挑战者随机选择的已知指数。并且挑战者随机选择指数 之后根据假设将式中的公开参数发送给敌手。



阶段1和2：挑战者为敌手生成的那些非授权集合生成私钥。两个阶段上的操作一样。令属性集合表示敌手生成的属性集合。

由于是的非授权集合，因此对于所有的存在向量使得并且。之后，挑战者选择，并设置：



之后，计算私钥的各项组件：



其中，通用组件的计算如下：



部分可以由挑战者计算，剩余部分可以被约掉。因此，对于每个属性挑战者进行如下设置：



其中。中的部分计算如下：



是参数的剩余部分。 的第二个部分与的一部分可以抵消。因此，挑战者对于计算和，并将私钥发送给敌手。

挑战：敌手输出等长的一对明文。在该阶段，挑战者随机选择并以如下方式构造密文：



其中是挑战项，是假设的相应项。

挑战者设置向量：



其中，。之后，计算向量，对于每行，计算:



值得注意的是，挑战者可以得到参数。对于每一行，挑战者计算。最后，挑战者以如下方式计算密文：



最后，挑战者将密文发送给敌手。

猜测：在阶段2的查询后，挑战者根据上述内容构造私钥，敌手对的值给出一个猜测。如果，挑战者输出0；否则，挑战者输出1。如果是群上的一个随机项，则消息的所有信息都将丢失在质询密文中。因此，敌手的获胜优势恰好为0。可以说，如果敌手以不可忽略的优势破坏了安全博弈，则挑战者在打破安全假设时具有不可忽略的优势。

### 全策略隐藏的安全性证明

全策略隐藏中DU借助多个标记向量计算映射关系。其他用户在得到标记矩阵和向量时可知属性和密文组件的映射关系，而这在不支持策略隐藏的方案中则由公开的映射函数表示。特别的是，用户还能从标记向量中得到属性之间的顺序关系，但利用这些顺序关系并不能推测出属性的明文信息。因此可以说，DO给出的这些标记向量不会泄露属性的明文信息。

对于授权关系判断的安全性证明则是通过对用户属性集合和最小授权集合之间的包含关系的安全性证明实现的。

定理4-2. 协议保密的计算了两个参与方之间的包含关系。

证明. 通过构造满足式子的模拟器来证明定理4-2。在中，存在以下两种情况：





假如式成立，构造模拟器。接收作为输入，工作步骤如下：

（1）接收输入。首先它随机选择集合，其中。之后，它为集合构造多项式。令表示多项式系数的集合即。

（2）从整数群中选择较大的随机数，计算多项式系数对应的承诺。之后模拟器根据集合计算向量。参与者也选择较大的随机数，计算多项式系数对应的承诺。

（3）计算式，



从而得到判断结果。参与者计算式并得到结果。



根据上述模拟过程，可知，，。根据假设可得。由于，可得，因此，式成立。



而在本文场景中，DO是参与者而DU是参与者，前者负责计算模拟器中的相关参数，后者则是计算最终结果，假设的情况不会发生在本场景中。因此不需要构造。

# 基于访问权限撤销的用户数据可信删除方案

针对患者无法验证云端是否确实执行了健康数据的删除这一问题，本章在上述方案的基础上实现云端数据可信删除方法。首先给出策略图和关键属性集合定义以及属性撤销和策略删除的关系。之后，给出数据可信删除方案的具体细节。最后，给出数据可信删除的安全证明。

## 问题定义

### 策略图

在传统的CPABE中，一个文件通常是一个最小访问单元，也就是说，不同的文件用不同的加密密钥加密。然而，随着企业文件的增多，用户需要管理大量的密钥。为了减轻用户端的密钥管理负担，本文将具有同一种访问控制策略的文件作为一个最小访问单元。也就是说，使用相同的密钥加密这些文件。本节定义保护类来描述一组具有相同访问策略的文件。一个保护类对应一个访问策略，不同的保护类对应的访问策略不同。因此，一个保护类保护的一组文件是访问、删除的最小单位。也就是说，用户通过删除密钥发起删除操作时，删除的是某个策略加密的所有文件。当然，本文的访问机制也可以退化为一个文件对应一个加密密钥，只要DO人为的将相同访问策略或上一个时间属性使每个策略各不相同即可。在本文中，通过加密密钥的删除实现文件删除。而加密密钥的删除依赖于对应访问策略的删除。

首先，本文给出一个策略图来描述用户、属性、文件和策略之间的关系，如图5.1所示。令表示一个策略图。是节点集合，每个节点都是一个布尔值。它共包含两种节点：源节点和内节点。源节点就是属性节点，只有出度没有入度。内节点就是保护类节点，既有入度也有出度。是边的集合，它共有两种边，从属性节点到保护类节点、从保护类节点到保护类节点。每个属性节点与一组用户有关，表示这些用户都包含该属性，也就是说，对于这组用户来说该节点的值为真。每个保护类节点与一组文件有关，表示这组文件的访问策略就是该保护类节点表示的访问策略。当一个用户是一个访问策略的授权用户时，该策略对应的保护类节点的值为真。例如，保护类节点的访问策略为。对于和而言，为真。而对于和，为假。保护类节点的值与用户是否被授权有关。当保护类节点为真时，表明该用户是保护类对应访问策略的授权者，可以正确解密加密密钥。当保护类节点为假时，说明用户是非授权用户。



策略图

### 关键属性集合

为了实现用户访问权限撤销，本文为每个访问策略定义一个关键属性集合。一个属性集合是一个访问策略的关键属性集合当且仅当该集合中全部属性为假时，该策略的值总为假。令表示一个访问策略，是策略对应的属性集合。令表示的一个关键属性集合。不属于的属性不会影响策略的真值，因此本文只研究包含于的关键属性集合。也就是说，每个关键属性集合必然包含于策略的属性集合。

一个访问策略会有多个关键属性集合，这由构成策略的门类型决定。一定是的一个关键属性集合。如图5.2所示，一个策略为，另一个为。前者共有3个关键属性集合，，和。后者只有1个关键属性集合，。对于纯与门策略来说，将其任意孩子节点置为false其值总是fasle。对于纯或门策略来说，只有将所有孩子节点置为false其值才总是fasle。任意策略均可以表示为二叉树的形式，每个非叶节点都可以抽象为一个如图5.2所示的简单策略。通过递归的求每个节点的关键属性集合可以得到一个策略的关键属性集合。为了实现高效删除，本文总是选择最小的关键属性集合。



简单策略的关键属性集合

### 属性撤销与策略删除

为了实现策略删除，本节讨论属性撤销与策略删除之间的关系，在Yang等人[[30](#_ENREF_30" \o "Yang, 2013 #49)]提出的属性删除思想上扩展本文的策略删除方案。属性撤销意味着用户权限的变更。当一个用户的属性被撤销时，该用户对所有访问策略的权限被更新，而其他用户的访问权限不受任何影响。在图5.1中，当我们撤销用户的属性，**对**来说的值由变为，而与有关但未发生撤销的用户和不受任何影响。Yang等人[[30](#_ENREF_30" \o "Yang, 2013 #49)]提出了一种高效的属性撤销方案，如图5.3所示。当一个用户的属性被撤销时，其他包含该属性但未发生撤销的用户需要更新被撤销属性对应的私钥组件，包含被撤销属性的密文同样也需要更新该属性对应的密文组件。这样，被撤销某个属性的用户将不能解密包含该属性的密文，而其他包含该属性的用户的解密不受影响。

在本文中，通过策略删除实现加密密钥的删除。本文的删除思想是当所有用户对要删除的访问策略的权限变为非授权时，我们就认为该访问策略被删除了。当一个策略被删除后，对所有用户来说，该策略的值都是。也就是说，所有授权用户对该策略的访问权限被改变了。该访问策略对应的保护类保护的所有文件被删除了。根据Yang等人的属性撤销的实现，本文通过仅更新被删除策略的密钥密文实现策略删除，如图5.3所示。在这里，密文的更新可以是更新任意一个关键属性集合中所有属性对应的密文组件。在图5.1中，为。集合是的一个关键属性集合，只要更新属性的密文组件即可达到密钥删除的目的。



属性撤销和策略隐藏示例

## 数据可信删除方案设计

本文中的可信删除方案由DO和CS共同实现。其中，DO负责发起删除请求并生成删除密钥，最后验证CS的密文删除。而CS负责根据删除密钥更新密文。

### 删除密钥生成

在算法中，DO先为删除策略选择最小关键属性集合，算法具体过程如表5.1所示。

最小关键属性集合选择算法

|  |
| --- |
| 最小关键属性集合选择算法 |
| 输入：访问策略对应的二叉树的根节点；  输出：包含所有最小授权集合的集合；  **procedure**    **if**  **then**    **return**  **end if**      **if**  **then**  **if**  **then**    **else**    **end if**  **else**  **if**  **then**    **end if**  **end if**  **return**  **end procedure** |

为什么需要执行额外的操作选择最小关键属性集合而不是直接使用所有的属性构成的关键属性集合？原因在于，需要更新的密文组件数与关键属性集合中的属性个数一致。因此，选择的关键属性集合越小，要更新的密文组件数越少。而更新密文所消耗的时间远大于执行该算法所消耗的时间。在中，从叶子节点开始递归的计算每个节点的最小关键属性集合。对于非叶节点来说，如果当前节点是与门，则其最小关键属性集合为左右孩子节点的最小关键属性集合中较小的一个，如果当前节点是或门，则其最小关键属性集合为左右孩子节点的最小关键属性集合的并集。

之后，为中的每个属性生成它的删除密钥，其计算方式如下：



其中，。由于基础方案中包含了策略隐藏功能，在生成删除密钥时为了明确删除密钥与密文组件之间的关系，还生成一个行标记向量。向量来标记中的属性在访问结构中的行，当的值为真时，说明该分量表示的属性在选择的关键属性集合中。最后，将生成的删除密钥发送给CS。

值得注意的是，本算法在DO端产生了一些指数计算开销。当要删除的策略的关键属性集合个数比较多时，这会导致用户端开销较大。但是分析发现删除密钥的生成时间可以在策略生成后和密文删除前之间的任意一个时间点。因此，根据智慧医疗系统的不同需求，在实际实现中可以采用两种方法减轻用户端开销。一种是DO在离线阶段生成删除密钥。另一种是DO在即将删除密文时通过指数外包计算生成删除密钥。

### 数据删除

在中，CS根据更新标记向量中为的值对应的密文组件，其更新方式如式所示：





可以看到，CS只更新集合中的属性的密文组件。在密文更新后，所有用户都变为该策略的非授权用户，都不能正确解密密文。也就是说，该策略对应的文件组被删除。

### 删除验证

由于密文更新在不可信的第三方执行，所以DO需要通过Merkle树对CS上的密文更新进行验证。图5.4给出删除策略中属性个数为4时的Merkle根生成示例。在DO计算初始密文时，为密文组件计算一个Merkle根，。而在验证密文更新时，DO从CS处获取最新的密文，并执行如式所示计算：



之后，为密文组件计算一个Merkle根，。如果等式成立，说明CS诚实地执行了更新。否则，重新向CS发送删除请求直到验证成功。



密文更新验证示例

## 数据可信删除的安全性证明

数据可信删除的安全性证明与基础安全性证明基本类似，其区别在于敌手定义的属性集合可以是任意集合以及挑战者在密文构造时的构造方式不同。具体的证明思想如图5.5所示。在基础方案的安全证明中，其构造思想为非授权的用户无法正确解密挑战密文。而在可信删除的安全证明中，通过密文更新将所有用户变为非授权用户。值得注意的是，当系统中属性较多时，密文更新后可能会导致原来某个非授权用户凑巧可以解密密文。但是，由于删除密钥中包含随机值，任何人没办法确定更新后的密文对应的授权用户。当非法用户尝试解密更新后的密文时，并不能得到密文组件和属性私钥之间的对应关系，从而无法正确解密，所以可以认为通过更新确实达到了可信删除的目的。



可信删除的安全证明思路

如果假设成立，则所有具有大小为（其中）的挑战矩阵的PPT对手在选择性地破坏基础CPABE方案时具有可忽略的优势。

具体的证明如下。

初始化：敌手将挑战矩阵发送给挑战者，该挑战矩阵也是要删除的策略对应的矩阵。其中矩阵是一个大小的矩阵。其中并且。同时，敌手给出该挑战矩阵对应的最小关键属性集合的行标记向量。

建立：挑战者将系统公开参数发送给敌手。之后将主密钥设置为。其中，和在假设阶段设置而是由挑战者随机选择的已知指数。并且挑战者随机选择指数之后根据假设将下述公开参数发送给敌手：



阶段1和2：挑战者为敌手生成的那些非授权集合生成私钥。两个阶段上的操作一样。令属性集合表示敌手生成的属性集合。

是任意属性集合。挑战者对于计算和，并将私钥发送给敌手。私钥组件中参数的构造方法与基础方案的安全性证明中一致。

挑战：敌手输出等长的一对明文。在该阶段，挑战者随机选择并构造密文。当是时，密文各个组件构造方法与4.4.1中一致。而当是时，除了组件之外，其余参数也与4.4.1中一致。当是时，挑战者以的方式计算：



其中，。而当是时，挑战者以的方式计算：



最后，挑战者将密文发送给敌手。

猜测：在阶段2的查询后，挑战者根据上述内容构造私钥，敌手对的值给出一个猜测。如果，挑战者输出0；否则，挑战者输出1。如果是群上的一个随机项，则消息的所有信息都将丢失在质询密文中。因此，敌手的获胜优势恰好为0。可以说，如果敌手以不可忽略的优势破坏了安全博弈，则挑战者在打破安全假设时具有不可忽略的优势。

# 性能对比和实验分析

本章从计算开销、存储开销和通信开销这三个方面出发对本方案中指数外包、全策略隐藏和数据可信删除的性能进行分析，并与文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中的策略隐藏、文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中的数据可信删除进行对比。之后，在已有的公开库的基础上对本方案以及文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]、文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]进行仿真实现并对实验结果进行具体分析。

## 性能对比

按照功能的不同，本文的内容可以分为四个部分：基础加密、全策略隐藏、数据可信删除以及指数外包。而本方案中全策略隐藏的实现以及数据可信删除可以被扩展至任意支持LSSS的CPABE方案。也就是说，本文方案中的基础加密部分是可变的。因此，本节不需要对基础的加密解密开销进行性能分析。对于其他三个部分，本节给出在不同算法中每个参与方的计算开销和通信双方间的通信开销。其中，将本文给出的全策略隐藏方案与Yang等人[[15](#_ENREF_15" \o "Yang, 2016 #31)]提出的全策略隐藏方案的进行对比，将数据可信删除方案与Yu等人[[28](#_ENREF_28" \o "Yu, 2018 #43)]提出的可验证数据删除方案进行对比。由于表中区域限制，本文在给出开销函数时只保留最高次项，并且使用缩写表示算法。

表6.1给出后续说明中使用的参数。其中，表示执行一次群上的操作需要消耗的时间并在表6.1中给出了不同的操作标识符。对于一个授权用户来说，使用本文中的算法找到的是包含于用户属性集合且满足策略的一个最小属性集合，而文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中的算法找到的则是用户属性集合和策略属性集合的交集。前者必然包含于后者，也就是说本文找到的集合的阶小于等于文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]找到的集合的阶，即。

计算参数描述

|  |  |
| --- | --- |
| **参数** | **含义** |
|  | 群上的运行一次运算需要的时间 |
|  | 乘法运算 |
| ， | 两个不同的指数操作，前者指数长度远小于后者 |
|  | 一次双线性对映射 |
|  | 一次哈希操作 |
|  | 授权用户对一个策略的一个最小授权集合， |
|  | 访问策略的属性集合 |

表6.1 计算参数描述（续）

|  |  |
| --- | --- |
| **参数** | **含义** |
|  | 用户属性集合 |
|  | 文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中执行一次策略隐藏所需的哈希函数个数 |
|  | 策略的属性集合与用户属性集合交集 |
|  | 访问策略的所有最小授权集合阶的和 |
|  | 访问策略的所有最小授权集合的个数 |
|  | 每个客户端需要存储的指数对的平均个数 |
|  | 访问策略的一个最小授权集合 |
|  | 文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中属性的一个共享分量长度 |
| |U| | 文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中属性总个数 |
|  | 文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中个哈希函数对属性域中每个属性作映射后所有值构成的集合的阶 |

在方案中，当要执行的多个指数计算之间不存在顺序和调用关系时，本文假设客户端可以选择多个边缘服务器并行的执行外包计算。本文在确定并行的边缘设备个数时主要考虑三个限制因素：（1）用户端支持的最大并行数；（2）用户端支持的最大带宽与单次指数外包需要传输的数据大小的商；（3）边缘设备的部署限制。其中的最小值决定了最终的并行个数。随着物联网设备、5G等技术的发展，边缘设备会逐渐成为像云服务器那样的基础设施而传输带宽也会增大。因此，可以说用户端支持的最大并行数会逐渐成为主要限制因素。

### 计算开销

本节主要分析在执行不同算法时参与者要执行的具体开销。

指数外包操作由DO/DU和ES共同完成。表6.2中给出在执行形如和的指数计算时，客户端未进行指数外包和使用指数外包算法分别所需要的计算开销。表6.2中只给出在线阶段的计算开销，因为指数外包的离线阶段会在客户端空闲时执行，一般认为离线阶段的计算开销不会影响系统实时性。显然，对于，客户端的一次指数运算被转化为一次短指数运算和一次加法及三次乘法运算。而对于，客户端的次指数运算被转化为一次指数运算和次加法及次乘法运算。可以说，本文使用的两个可验证指数外包算法有效地减轻了用户端的计算开销。

指数外包前后的计算开销对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | 外包前 | 外包后 | |
| DO/DU | DO/DU | ES |
|  |  |  |  |
|  |  |  |  |

全策略隐藏阶段包含两个算法和，其计算开销如表6.3所示。其中，表示，表示。文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中这两个算法的参与者只有用户自己，而在本文的方案中，由于用到了指数外包计算，参与者还包括多个边缘服务器。在中，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]的计算开销与策略中属性个数和对每个属性使用的哈希函数个数正相关。而本方案则与策略的最小授权集合个数以及所有最小授权集合的阶的和有关，因为在本方案中DO需要求访问策略的每个最小授权集合的密文。而在中，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]的计算开销与用户属性个数和对每个属性使用的哈希函数个数正相关。本方案则与查找用户包含的最小授权集合这一算法的具体执行有关。本文假定在发送策略密文前，按照最小授权集合的阶的大小进行排序。那么，用户包含的最小授权集合的阶越小，该算法就会越快找到，相应的，计算开销就越少。当不包含这样的集合时，该算法的开销达到最大值，因为它要遍历访问策略的所有最小授权集合。表6.3中给出当算法遍历完所有最小授权集合所需的计算开销，其大于或等于任意用户执行该算法时的计算开销。文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中的值是由用户设置的，且受用户属性总个数的影响。当属性总个数越大时，值越大才能保证策略隐藏的安全。因此，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]不适用于大属性域。而本文中策略隐藏方案不受属性总个数的影响，仅与用户访问策略有关。

策略隐藏中的计算开销

|  |  |  |  |
| --- | --- | --- | --- |
|  | [[15](#_ENREF_15" \o "Yang, 2016 #31)] | 本方案 | |
| DO/DU | DO/DU | ES |
|  |  |  |  |
|  |  |  |  |

数据可信删除阶段包括、和三个算法，其计算开销如表6.4所示。其中，表示，表示，表示。在删除部分，本方案及文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中的参与者均涉及到DO和CS。本方案在可信删除中的计算开销与策略对应的最小关键属性集合的阶有关。而在文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中，DO在和算法中和CS协商一个删除密钥并在验证阶段通过再次解密验证密文的更新。从表6.4中可以看出，在验证阶段，文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中的DO需要执行多次双线性对以及指数操作。从整个删除阶段来看，本方案在删除部分的高效性更加明显。

数据删除中的计算开销

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方案  算法 | 本方案 | | [[28](#_ENREF_28" \o "Yu, 2018 #43)] | |
| DO | CS | DO | CS |
|  |  |  |  |  |
|  |  |  |  | 3 |
|  |  |  |  |  |

### 通信开销

本节给出指数外包、全策略隐藏以及数据可信删除中的通信开销。在指数外包部分，给出执行一次指数操作时，客户端和服务器分别所需的指数开销。在策略隐藏部分，仅分析和算法带来的通信开销，不重复分析引入指数外包带来的通信开销。并且上述两个算法的通信开销一致的，通信方向相反，因此表中只给出一个算法的通信开销。在可信删除部分，算法和会产生通信开销，由于通信双方不变，而传输内容发生变化，因此分别给出其通信开销。

指数外包中的通信开销如表6.5所示。其中，箭头表示数据传输方向。第一种指数计算的通信开销可以忽略，因为其只需要传输两个群中的元素。第二种指数计算的通信开销与要计算的指数对的个数成正比。

指数外包中的通信开销

|  |  |  |
| --- | --- | --- |
|  | DO/DU→CS | DO/DU←CS |
|  |  |  |
|  |  |  |

全策略隐藏中的通信开销就是策略密文的传输，如表6.6所示。其中，算法不同，其传输方向不同。在中，是DO发送数据而在中则是DO接收数据。文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中策略密文长度为。当个哈希函数中任意函数都能将属性集合中每个属性映射到不同的值时，为极限取值。此时，。而本文中的策略密文长度仅与策略中最小授权集合个数以及每个最小授权的阶有关，属性集合的大小不会影响密文长度。可以看到，当属性集合越大时，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中的策略密文长度也随之增大，这再一次表明该方案不适用于大域。

全策略隐藏中的通信开销

|  |  |  |
| --- | --- | --- |
|  | [[15](#_ENREF_15" \o "Yang, 2016 #31)] | Ours |
| DO/DU↔CS | DO/DU↔ES |
|  |  |  |

数据可信删除的通信开销就是删除密钥和更新密文的传输，如表6.7所示。像表6.6中一样，双向箭头表示任意传输方向，根据算法不同，其传输方向不同。从表6.7中可以看出，密文删除阶段的传输开销与删除策略的最小关键集合有关，而文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]仅需要传输一个删除密钥和少量其他相关参数。但本文在验证时只需要传输少部分密文，而文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]要传输全部的密文。

数据删除中的通信开销

|  |  |  |
| --- | --- | --- |
|  | [[28](#_ENREF_28" \o "Yu, 2018 #43)] | Ours |
| DO/DU↔CS | DO/DU↔CS |
|  |  |  |
|  |  |  |

### 存储开销

表6.8中给出文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]、文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]及本文方案中各个参与者的存储开销。AA只负责用户密钥生成并不承担存储任务。ES只负责辅助计算，也不产生存储开销。从表6.8中可以看出，DO/DU和CS在三个方案中均需承担存储任务。其中，DO/DU存储用户私钥，CS存储数据密文和密钥密文。对于本方案和文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]来说，CS还需存储访问策略密文。与通信开销的分析一样，当属性集合越大时，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]生成的策略密文长度越大。与文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]相比，本方案在CS端额外产生了策略密文的存储开销。其原因在于本方案实现了全策略隐藏来保证属性隐私而文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中并未实现该功能。与文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]和文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]相比，本方案的密钥密文长度最长。这是因为本方案采用的基础CPABE方案为了能够支持大域需要存储更多的信息。当然，本方案中的全策略隐藏和数据可信删除也可以在文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]和文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]的基础方案上实现。此外，为了实现指数外包，负责执行运算的DO需要额外存储多对指数对。但这保证了用户端的高效性。从现实角度来讲，牺牲部分存储开销来提高计算效率是合理的。

存储开销

|  |  |  |
| --- | --- | --- |
|  | DO/DU | CS |
| [[15](#_ENREF_15" \o "Yang, 2016 #31)] |  |  |
| [[28](#_ENREF_28" \o "Yu, 2018 #43)] |  |  |
| Ours |  |  |

## 实验分析

本节首先给出方案实现时用到的具体实验环境，之后，对每个算法的仿真结果进行分析说明。

### 实验环境

表6.9给出本文实验测试的基本配置信息。在De Caro A和Iovino V给出的JPBC库[[42](#_ENREF_42" \o "De Caro, 2011 #65)]和Liu给出的CloudCrypto库[[43](#_ENREF_43" \o "Liu, 2014 #66)]基础上使用Java语言在win系统上实现本方案、文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]和文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中的所有算法。其中，代码中使用的群参数以及哈希函数如表6.9所示。在基础加密部分和策略隐藏部分与文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]进行对比，而在删除部分与文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]进行对比。在实验测试中，本文假设客户端支持的最大并行数为4。

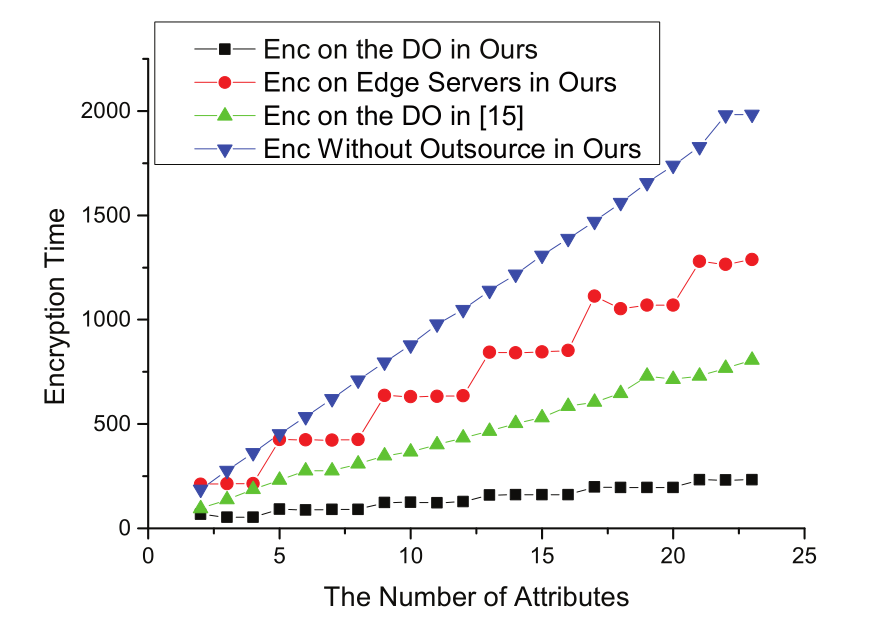
实验参数设置

|  |  |
| --- | --- |
| 项目 | 参数 |
| PC配置 | Windows x64, Intel Core i7 3.40GHZ and 8.00GB RAM |
| 编程语言 | Java |
| 库 | jpbc, https://github.com/liuweiran900217/CloudCrypto |
| 群参数 | ：160-bit，：512-bit，哈希：SHA-256 |
| 测试方案 | 本方案，[[15](#_ENREF_15" \o "Yang, 2016 #31)]、[[28](#_ENREF_28" \o "Yu, 2018 #43)] |

### 实验结果及分析

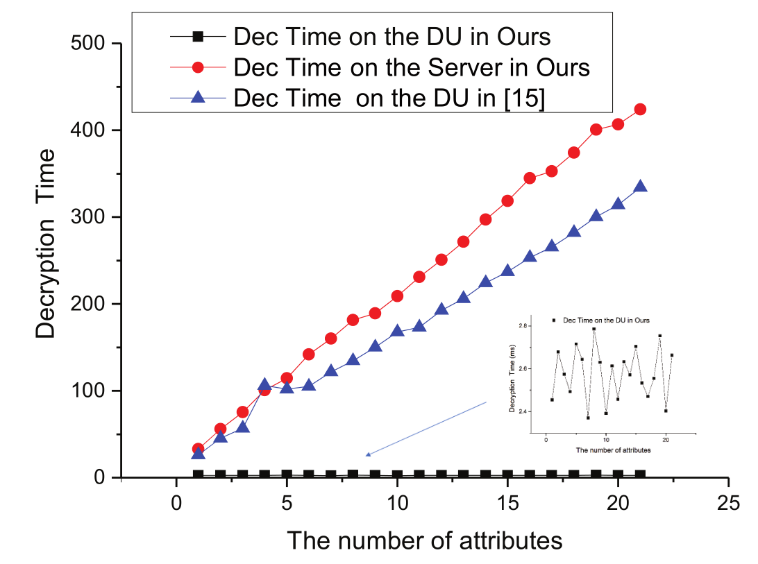
本文的全策略隐藏方案和数据可信删除方案可以被扩展至任意基于LSSS的大属性域CPABE方案中，因此比较本方案与对比方案的加密和解密开销没有实际意义。但是为了说明在加密阶段使用指数外包算法可以有效提高DO的加密效率，本实验中仍然对基础加密和解密部分进行测试。测试的算法共包括：文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中的加密、解密、策略隐藏和解密测试，文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中的删除密钥生成、数据删除和删除验证，以及本方案中的加密、解密、策略隐藏、解密测试、删除密钥生成、数据删除和删除验证这14个算法。测试结果中所有时间的单位为毫秒（ms）。

加密算法的测试共包含以下4种情况：执行指数外包时的DO端加密开销、执行指数外包时的ES端加密开销、文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中的DO端加密开销和不执行解密外包的DO端加密开销，如图6.1所示。图6.1中，横轴为访问策略中包含的属性个数，纵轴为加密时间。可以看到，在外包前，本方案的加密开销高于文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]。而当加密计算中的指数运算被并行的外包给多个ES后，本方案的加密开销低于文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]。显然，指数外包计算有效地减轻了用户端的加密开销。



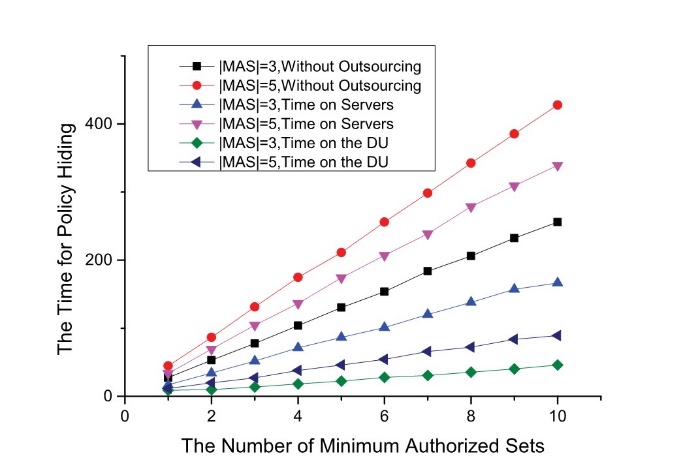
加密计算开销

解密算法的测试共包含以下3种情况：如图6.2所示，DU端解密开销、ES端解密开销、文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中的DU端解密开销。其中，横轴为用户属性集合中的属性个数，纵轴为解密时间。可以看到，本文的解密计算在用户端产生的开销远低于文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]，大部分解密开销被卸载至ES端，DU的解密时间几乎为0。



解密计算开销

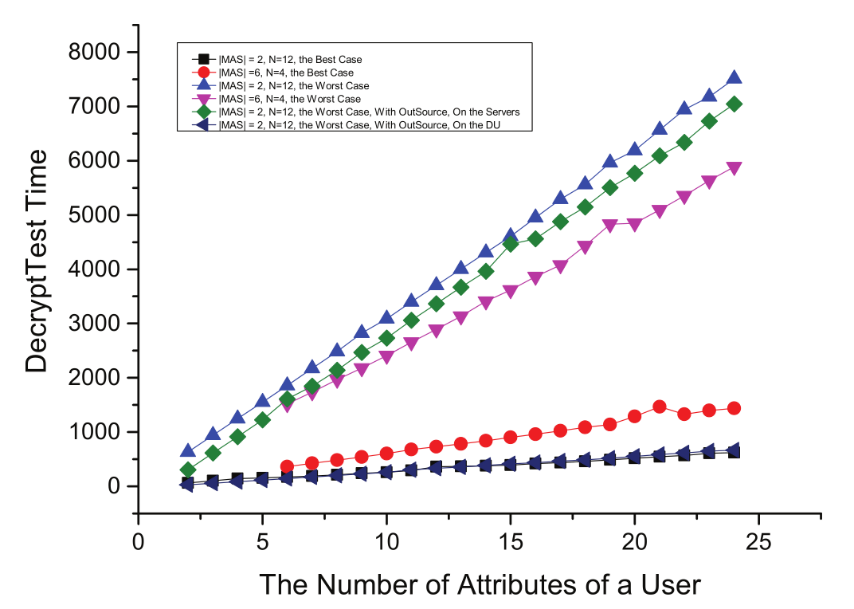
策略隐藏算法的计算开销包含多项式的计算以及策略密文的计算，其中多项式计算主要是乘法运算，在访问策略确定后可以在离线时计算多项式系数。分析发现，策略隐藏算法开销与策略中每个最小授权集合的阶及最小授权集合的个数相关。实验结果如图6.3所示，其中，横轴为策略包含的最小授权集合的个数，纵轴为计算时间，表示一个策略的所有最小授权集合的阶的平均值。可以看到，随着最小授权集合个数的增加，策略隐藏所需的开销也逐渐增加。除了最小授权集合个数影响策略隐藏开销外，最小授权集合的平均阶也会影响计算开销。从图6.3可以很明显地看到，最小授权集合的平均阶越大，开销越大。此外，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中策略隐藏算法的计算开销为多次哈希函数的计算，与访问策略中的属性个数和的值成正比。由于本文的策略隐藏开销和文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]的影响因素不一致，无法在一个图中进行比较。作为参考，本节测试了SHA256算法对一个属性字符串进行一次哈希的时间，为10ms左右。当访问策略中属性个数为20，取值为3时，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]进行一次策略隐藏的时间为600ms。而这只是在为3时的计算开销。随着用户数目的增长，属性域势必要增大，为了确保方案的安全性，值也要增加。相比而言，本文的策略隐藏开销与属性域和策略中属性个数无关，只与策略复杂度有关，其变化比较稳定且造成的客户端开销较少。



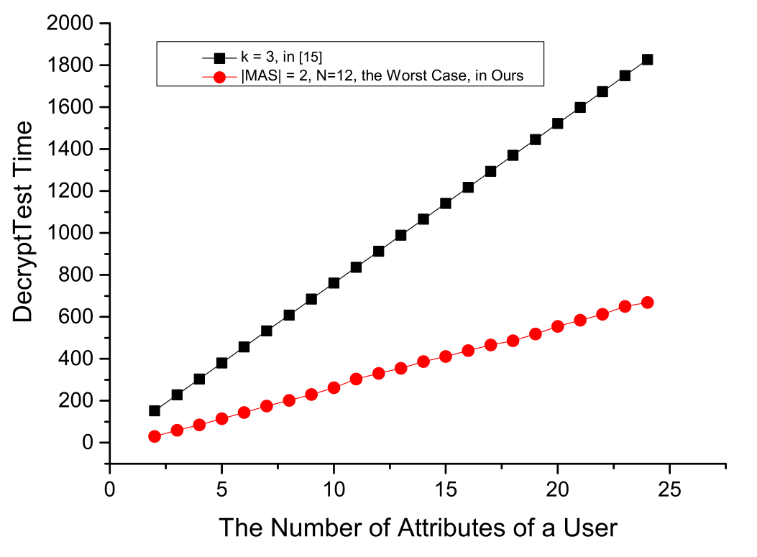
策略隐藏开销测试

在两个方案中，解密测试的结果不同，用户需要执行的操作不同。对于授权者，本方案及文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]均需执行解密测试和解密。因此，只对比两个方案中授权用户的解密测试开销。而对于非授权者，本文只需执行解密测试，而文献[15]根据解密测试求出的交集是否为空决定是否执行解密。当交集为空时，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]也不需要执行解密。因此，对于交集非空的非授权者，文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]将其解密时间也作为其解密测试开销的一部分。而交集为空时的非授权者的解密测试对比与授权用户最坏情况时的对比一致。因此，本文对解密测试算法开销的测试分为授权用户测试、交集非空的非授权用户测试。

授权用户的测试考虑最好情况、最坏情况、是否外包以及策略的复杂度。最好情况指该用户在遍历最小授权集合寻找具有包含关系的特定集合时，处理的首个集合就是该用户包含的一个最小授权集合。最坏情况则是在遍历结束时才找到该集合。外包则是当用户执行查找该集合的算法时将指数计算外包给多个ES。由于解密测试会受到对应策略复杂度的影响，但策略复杂度又不能采用一个具体的数学公式对其进行量化。因此，本文简单地使用所有最小授权集合的平均阶和策略中最小授权集合的个数这两个变量共同表示策略的复杂度。显然，当的值越大，策略越复杂。当值相同时，这两个值不同，策略的复杂度也不同。本节测试了当相同但和各自不同时对解密测试的影响，如图6.4所示。图6.4中，横轴表示授权用户包含的属性个数，纵轴表示计算开销。从图6.4中可以看出：1）指数外包可以有效减轻用户端的解密测试开销；2）用户集合中属性个数与解密测试开销正相关；3）策略越复杂，解密开销越大。此外，本文还测试了当使用相同的访问策略时，本方案在最坏情况与文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]的一般情况的计算开销，如图6.5所示。根据文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中给出的实验设定，本节测试时假设为3。显然，为3时本方案中的用户端解密测试开销远小于需要执行多次哈希算法的文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]。

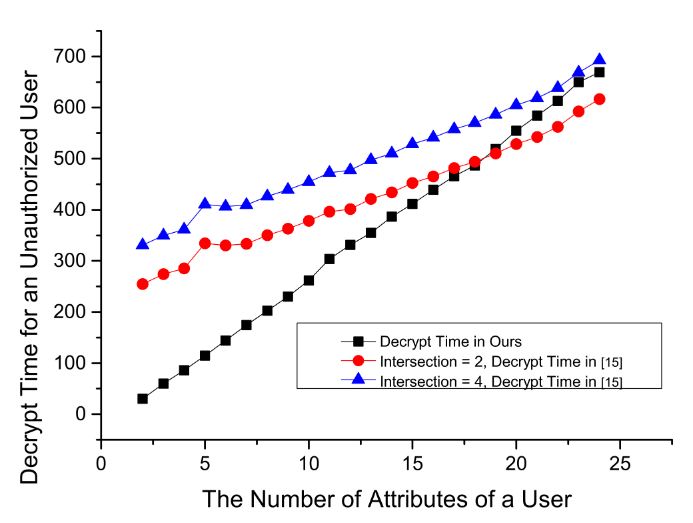


授权用户的解密测试开销



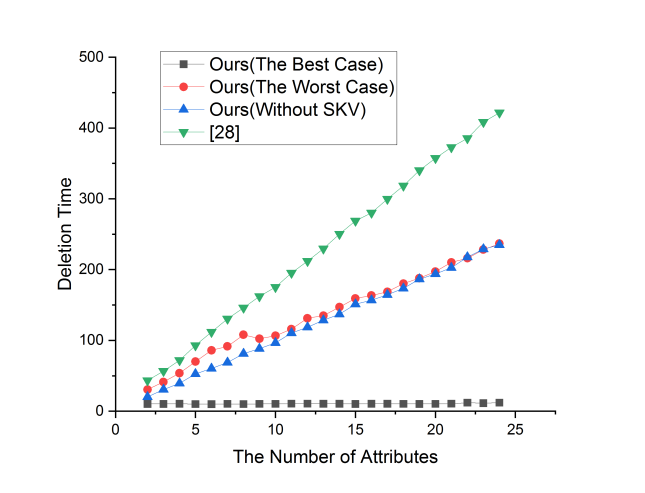
授权用户解密测试开销对比

而交集非空的非授权用户的解密测试如图6.6所示。本文测试了在文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]中，当为3时，交集为2和交集为4的解密开销。可以看到，即使是在交集为2的情况下本方案的解密测试开销也小于文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]。这是由于文献[[15](#_ENREF_15" \o "Yang, 2016 #31)]还需要执行解密计算即多次双线性对操作。



交集非空的非授权用户的解密测试开销对比

在数据删除部分，本文将删除中的3个算法的总时间与对比算法进行比较，如图6.7所示。图6.7中，横轴为删除策略的属性集合中属性的个数，纵轴为删除时间。其中，共测试了四种情况：最好情况、最坏情况、不执行关键属性集合选择算法的情况以及文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]中的删除时间。最好情况指关键属性集合中只包含一个属性，而最坏情况则指关键属性集合中包含策略中的所有属性。从图6.7中可以看到，执行关键属性集合选择算法时删除所消耗的时间基本与最坏情况一致，可见选择算法能有效地提高删除效率。此外，图6.7也表明在没有采取5.2.1给出的两种优化手段的情况下，本方案的最坏情况的删除时间也远小于文献[[28](#_ENREF_28" \o "Yu, 2018 #43)]。



删除阶段总开销对比

综上所述，在基础加密部分，本方案借助指数外包和解密外包技术有效地减轻了加密和解密时用户端的计算开销；在策略隐藏阶段，本方案在实现支持大域的策略隐藏的同时也保证了用户端高效；在删除阶段，本方案实现了高效地可信删除。

# 总结与展望

## 研究工作总结

随着5G、物联网、边缘计算等技术的发展，智慧医疗逐渐被认为是未来患者就医的主要服务形式。与传统的就诊模式相比，智慧医疗在减轻医务人员负担的同时，能够获取更丰富的数据，给出更精确的治疗方案。然而患者健康数据作为智慧医疗中处理的主要信息，面临着被泄露风险。因此，如何设计保证患者隐私的数据访问控制方案成为智慧医疗发展中的主要问题。

已有的支持细粒度的访问控制方案不能支持数据可信删除和全策略隐藏，而部分属性隐藏或者不能支持大域的全策略隐藏不适用于智慧医疗场景。前者可能会导致用户敏感信息泄露，后者则严重影响方案的实用性。本文研究一种既支持全策略隐藏又能保证数据可信删除的适用于智慧医疗场景的访问控制方案。研究内容可总结为如下几点：

（1）针对系统研究目标，本文研究基于PSI技术实现在全策略隐藏情况下判断用户授权关系及私钥密文映射关系。通过在指数上构造多项式计算用户属性集合和访问策略的所有最小授权集合的包含关系，从而确定用户是否被授权。并设计多个辅助向量使得授权用户在判断授权关系的同时明确私钥密文的映射关系，从而在策略隐藏的情况下可以正确解密。此外，设计的全策略隐藏方案还能满足支持大域这一研究目标。

（2）针对现有CPABE方案不能支持数据可信删除这一问题，本文研究适用于CPABE方案的通用数据可信删除方法。本文通过构建策略图将文件删除转化为策略撤销，并分析属性撤销和策略撤销之间的关系，通过密文更新达到所有用户不能正确解密删除文件的目的。最后基于Merkle树判断CS是否诚实地执行了更新操作。

（3）为了确保系统实时性，本文借助已有的指数外包方案将用户端中特定形式的指数计算外包给边缘服务器，从而实现用户端高效。本文给出安全分析证明设计的方案能够同时保证数据的细粒度访问控制、用户的属性隐私以及数据可信删除。此外，性能对比和实验分析也表明，本方案在实现全策略隐藏和数据可信删除的同时并没有为用户端带来非常大的开销。

## 研究工作展望

本文设计并测试了适用于智慧医疗场景中的访问控制方案。本方案提出了基于PSI的全策略隐藏方法以及基于密文更新的通用数据可信删除方案。在上述方案的基础上，本节针对方案的实时性以及数据的安全性，给出以下两个研究点。

1. 策略复杂度与最小授权集合的阶及个数之间的定量分析。由于本文在测试中将最小授权集合的阶及个数作为两个变量，分别分析其对策略隐藏效率的影响。不能得到一个策略复杂度与隐藏效率之间的一个定量关系。之后应该通过引入数学模型对策略复杂度、最小授权集合的阶及个数这三个值建立关系，定义策略复杂度函数，从而可以定量的分析访问策略复杂度的改变对策略隐藏效率的影响。同时，也能进一步指导用户在加密前设计出更高效的访问策略。
2. 轻量级代理计算协议研究。在本文的模型中，资源受限的数据拥有者需要将数据传输至代理客户端进行加密。但在实际环境中，在传输数据时，不一定存在一个可靠的通道防止原始数据泄露，同时代理客户端并不一定是完全可信的。因此，为了提高方案的实用性，应该为资源受限的设备设计一种轻量级代理计算协议，使得这种类型的设备能够使用简单的计算将明文模糊处理并在代理客户端进行相应计算，保证数据隐私的同时又不为资源受限的用户带来较重的计算负担。

参考文献

[1] Catarinucci L, De Donno D, Mainetti L, et al. An IoT-aware architecture for smart healthcare systems[J]. IEEE Internet of Things Journal, 2015, 2(6): 515-526.

[2] February 2019 healthcare data breach report[EB/OL]. https://[www.hipaajournal.com/february-2019-healthcare-data-breach-report/](http://www.hipaajournal.com/february-2019-healthcare-data-breach-report/).

[3] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]. 2007 IEEE symposium on security and privacy (SP'07), 2007: 321-334.

[4] Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption[C]. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013: 463-474.

[5] Sahai A, Waters B. Fuzzy identity-based encryption[C]. Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2005: 457-473.

[6] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]. Proceedings of the 13th ACM conference on Computer and communications security, 2006: 89-98.

[7] Winsborough W H, Seamons K E, Jones V E. Automated trust negotiation[C]. Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, 2000: 88-102.

[8] Yu T, Winslett M, Seamons K E. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation[J]. ACM Transactions on Information and System Security (TISSEC), 2003, 6(1): 1-42.

[9] Boneh D, Waters B. Conjunctive, subset, and range queries on encrypted data[C]. Theory of Cryptography Conference, 2007: 535-554.

[10] Katz J, Sahai A, Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products[C]. annual international conference on the theory and applications of cryptographic techniques, 2008: 146-162.

[11] Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures[C]. International conference on applied cryptography and network security, 2008: 111-129.

[12] Lai J, Deng R H, Li Y. Expressive CP-ABE with partially hidden access structures.[C]. Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, May 2-4, Seoul, Korea, 2012: 18-19.

[13] Cui H, Wan Z, Deng R H, et al. Efficient and expressive keyword search over encrypted data in cloud[J]. IEEE Transactions on Dependable and Secure Computing, 2016, 15(3): 409-422.

[14] Zhang Y, Zheng D, Deng R H. Security and privacy in smart health: Efficient policy-hiding attribute-based access control[J]. IEEE Internet of Things Journal, 2018, 5(3): 2130-2145.

[15] Yang K, Han Q, Li H, et al. An efficient and fine-grained big data access control scheme with privacy-preserving policy[J]. IEEE Internet of Things Journal, 2016, 4(2): 563-571.

[16] Yang K, Jia X, Ren K. Attribute-based fine-grained access control with efficient revocation in cloud storage systems[C]. Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, 2013: 523-528.

[17] Feng X, Li C, Li D, et al. Fully Secure Hidden Ciphertext-Policy Attribute-Based Proxy Re-encryption[C]. International Conference on Information and Communications Security, 2017: 192-204.

[18] Li J, Ren K, Zhu B, et al. Privacy-aware attribute-based encryption with user accountability[C]. International Conference on Information Security, 2009: 347-362.

[19] Zhang Y, Chen X, Li J, et al. Anonymous attribute-based encryption supporting efficient decryption test[C]. Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, 2013: 511-516.

[20] Joukov N, Zadok E. Adding secure deletion to your favorite file system[C]. Third IEEE International Security in Storage Workshop (SISW'05), 2005: 8 pp.-70.

[21] Joukov N, Papaxenopoulos H, Zadok E. Secure deletion myths, issues, and solutions[C]. Proceedings of the second ACM workshop on Storage security and survivability, 2006: 61-66.

[22] Reardon J, Basin D, Capkun S. Sok: Secure data deletion[C]. 2013 IEEE symposium on security and privacy, 2013: 301-315.

[23] Boneh D, Lipton R J. A Revocable Backup System[C]. USENIX Security Symposium, 1996: 91-96.

[24] Mitra S, Winslett M. Secure deletion from inverted indexes on compliance storage[C]. Proceedings of the second ACM workshop on Storage security and survivability, 2006: 67-72.

[25] Perlman R. File system design with assured delete[C]. Third IEEE International Security in Storage Workshop (SISW'05), 2005: 6 pp.-88.

[26] Cachin C, Haralambiev K, Hsiao H-C, et al. Policy-based secure deletion[C]. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013: 259-270.

[27] Tang Y, Lee P P, Lui J C, et al. Secure overlay cloud storage with access control and assured deletion[J]. IEEE Transactions on dependable and secure computing, 2012, 9(6): 903-916.

[28] Yu Y, Xue L, Li Y, et al. Assured data deletion with fine-grained access control for fog-based industrial applications[J]. IEEE Transactions on Industrial Informatics, 2018, 14(10): 4538-4547.

[29] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization[C]. International Workshop on Public Key Cryptography, 2011: 53-70.

[30] Yang K, Jia X, Ren K, et al. DAC-MACS: Effective data access control for multiauthority cloud storage systems[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(11): 1790-1801.

[31] Liu Z, Duan S, Zhou P, et al. Traceable-then-revocable ciphertext-policy attribute-based encryption scheme[J]. Future Generation Computer Systems, 2019, 93: 903-913.

[32] Chen Y, Sun W, Zhang N, et al. A secure remote monitoring framework supporting efficient fine-grained access control and data processing in IoT[C]. International Conference on Security and Privacy in Communication Systems, 2018: 3-21.

[33] Ning J, Cao Z, Dong X, et al. Auditable $\sigma $-Time Outsourced Attribute-Based Encryption for Access Control in Cloud Computing[J]. IEEE Transactions on Information Forensics and Security, 2017, 13(1): 94-105.

[34] Cui H, Deng R H, Li Y, et al. Attribute-based storage supporting secure deduplication of encrypted data in cloud[J]. IEEE Transactions on Big Data, 2017.

[35] Freedman M J, Nissim K, Pinkas B. Efficient private matching and set intersection[C]. International conference on the theory and applications of cryptographic techniques, 2004: 1-19.

[36] Rivest R, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms[J]. Foundations of secure computation, 1978, 4(11): 169-180.

[37] Di Crescenzo G, Khodjaeva M, Kahrobaei D, et al. Practical and secure outsourcing of discrete log group exponentiation to a single malicious server[C]. Proceedings of the 2017 on Cloud Computing Security Workshop, 2017: 17-28.

[38] Wang Y, Wu Q, Wong D S, et al. Securely outsourcing exponentiations with single untrusted program for cloud storage[C]. European Symposium on Research in Computer Security, 2014: 326-343.

[39] Merkle R C. A digital signature based on a conventional encryption function[C]. Conference on the theory and application of cryptographic techniques, 1987: 369-378.

[40] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.

[41] Qin B, Deng R H, Liu S, et al. Attribute-based encryption with efficient verifiable outsourced decryption[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(7): 1384-1393.

[42] De Caro A, Iovino V. jPBC: Java pairing based cryptography[C]. 2011 IEEE symposium on computers and communications (ISCC), 2011: 850-855.

[43] CloudCrypto[EB/OL]. https://github.com/liuweiran900217/ CloudCrypto.

1. T. V. X. Phuong, G. Yang and W. Susilo: Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions[J].IEEE Transactions on Information Forensics and Security, 2016, 11(1): 35-45.
2. 王悦, 樊凯. 隐藏访问策略的高效CP-ABE方案[J]. 计算机研究与发展, 2019, 56(10): 2151-2159.
3. 王光波, 刘海涛, 王晨露, et al. 云存储环境下可撤销属性加密[J]. 计算机研究与发展, 2018, 55(6): 1190-1200.
4. 赵志远, 王建华, 朱智强, et al. 面向物联网数据安全共享的属性基加密方案[J]. 计算机研究与发展, 2019, 56(6): 1290-1301.

致谢

在2020年这个特殊的年份里，我迎来了我的毕业季。由于疫情形势严重，学校不能正常开学，我们只能在家里完成毕设论文。在整个论文的完成过程中，我的导师杨力老师给了我很大的帮助。从论文题目确定、应用场景选择、论文整体架构设计到具体的方案细节描述，他都悉心指导。

回顾这三年的研究生生活，我由衷地感谢我的导师。在我还不知道科研为何物时，他领我入门，教我们如何有效地阅读论文、如何查找相关文献以及在众多的参考文献中如何找到最有用以及质量最高的文章。感谢杨老师不断地给我机会，让我去参加国际会议，增长见识，开拓眼界。在这三年里，我取得的每一点进步都离不开老师指导。感谢老师在我懈怠时，能及时地督促我和鼓励我。不论是在研究上还是在生活中，杨老师总能在我遇到困难的时候，给予及时的帮助。能够成为杨老师的学生，我感到无比的幸运和感激。

感谢309实验室的每一位同学，感谢他们能在我遇到困难的时候不断地鼓励我。感谢韩孜怡师姐树立好榜样，激励我不断前进。感谢师兄韦腾、司杨涛、周亚胜、王焱济、张程辉、庞晓健和师姐王佳雪，感谢他们能及时回应我的各种疑问。感谢309电脑达人王宇晨大师弟帮我修电脑，使我的科研工作能得以持续。感谢排版神人李成师兄在我word排版遇到百度没法都解决的问题时给我有用的建议。感谢万人迷方方姐督促我参加企业活动，使我能够获得面试机会。感谢能吃妹妹马卓茹带我出去玩。也感谢其他所有可爱的师弟师妹们，在大家共同努力下，我们拥有一个良好的学习氛围。

最后，感谢我的家人，为我提供了一个良好的生活环境，让我没有后顾之忧，是他们无微不至的关怀支撑着我不断前进。

作者简介

##### 基本情况

程昱婷，女，陕西渭南人，1995年9月出生，西安电子科技大学网络与信息安全学院网络空间安全专业2017级硕士研究生。

##### 教育背景

2013.08～2017.07 西安电子科技大学，本科，专业：计算机科学与技术

2017.08～ 西安电子科技大学，硕士研究生，专业：网络空间安全

##### 攻读硕士学位期间的研究成果

###### 发表学术论文

1. Yang L, Cheng Y, Wang X, et al. Online handwritten signature verification using feature weighting algorithm relief[J]. Soft Computing, 2018, 22(23): 7811-7823.(SCI: 000451472000013)
2. Cheng Y, Yang L, Yu S, et al. Achieving Efficient and Verifiable Assured Deletion for Outsourced Data Based on Access Right Revocation[C]. International Conference on Cryptology and Network Security. Springer, Cham, 2019: 392-411. (EI: 20194807769616)

###### 申请（授权）专利

无

###### 参与科研项目及获奖

1. 国家自然科学基金项目，移动云计算中用户敏感数据的可信删除研究， 2017.01-2020.12，在研，负责云端数据可信删除协议设计与实现。