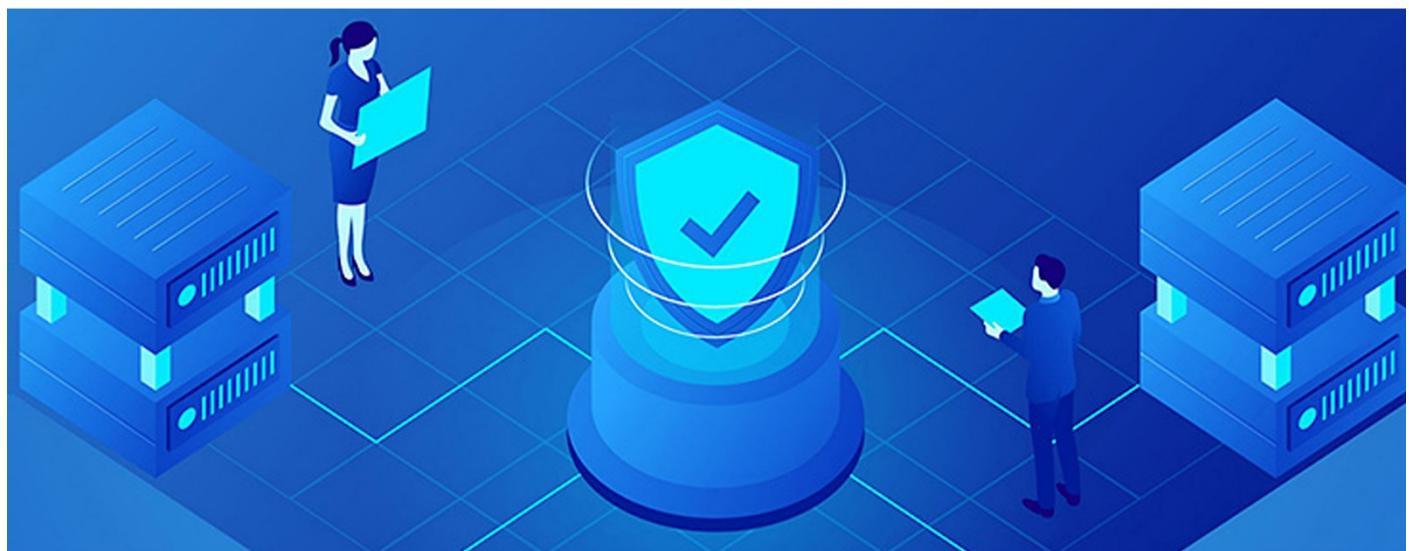




基于区块链的跨域认证与公平审计云存储系统

Cross-domain authentication and fair audit cloud storage system based on blockchain



项目分类：科技发明制作A类

所属领域：信息技术

参赛队员：曹寅峰 李艺扬 谢意 薛晨

指导教师：陈晓峰

摘要

习总书记在 4·19 讲话中强调，没有网络安全就没有国家安全。网络安全被提升到国家安全的战略新高度。如今，云计算是个人企业乃至政府机关网络服务的主要范式，其安全的重要程度不言而喻。云计算目前面临着诸多安全问题，一方面，广泛使用的传统跨域认证方案 PKI 运行效率十分低下，且证书服务器易受攻击，如 2011 年证书服务提供商 DigiNotar 的服务器遭黑客入侵，包括 Google、微软、中情局在内的重要网站网络流量数据被监听长达一年。另一方面，云服务器为了保护数据的隐私性会选择加密数据，但却不可避免地增加了数据的冗余程度与存储负担，从而更加难以保证数据的完整性。每年云存储服务商都会因数据冗余报废大量硬盘，不可避免的造成数据丢失。因此，如何设计兼顾安全隐私与效率的云存储底层解决方案，是云计算安全的关键所在。

本作品从上述问题出发，提出了基于区块链的跨域认证与公平审计云存储系统。系统主要结合了区块链与密码学技术，包括**区块链证书认证模块**，**公平完整性审计模块**与**密文去重模块**全方位地为云计算安全提供解决方案。

在认证模块中，通过使用我们新型认证模型与认证协议，可以从原理上避免传统认证中心（CA）层级式验证证书时产生的巨大开销与安全隐患。

在公平完整性审计模块中，我们结合数据拥有协议（Proof of Data Possession）与区块链智能合约技术，高效快速验证数据完整性，如果验证失败则通过智能合约自动补偿用户。

在密文去重模块中，我们使用收敛加密技术（Convergent Encryption），在保护用户隐私的同时减少云服务提供商的存储负担，达到密文去重的效果。

本作品能切实满足企业的需求指标，在认证模块中，我们的认证网络流量开销相比传统认证方案减少 70%，同时认证算法的时间复杂度变为 $O(1)$ ；在公平完整性审计模块中，如果服务器丢失 1% 的数据块，我们可以通过仅审计不超过 5% 的数据块，就能以 99% 以上的概率发现数据丢失，所需审计时间相比现有方案减少 90%；在密文去重模块中，可以避免相同文件使用不同密钥加密所产生的冗余。与传统加密相比， N 个用户加密相同文件，存储空间占用变为原来的 $1/N$ ，大大减少存储成本。

关键词： 跨域认证 完整性审计 密文去重 区块链 数据安全

Abstract

Cyber security is an important part of national security, but nowadays, cloud computing is the main paradigm of application services for individuals, enterprises and even government agencies, and the importance of its security is self-evident. The certificate authentication center that the traditional cross-domain authentication scheme relies on in the cloud environment is extremely vulnerable, which makes the traditional cross-domain authentication scheme not only inefficient but also huge in the verification process. Security risks. On the other hand, due to various software and hardware failures of the cloud server and the presence of malicious adversaries, the user's data may be maliciously tampered with or deleted. If cloud users still need to pay expensive storage fees to the cloud server if the data is damaged, this is extremely unfair to cloud users. Therefore, how to design a cross-domain authentication and fair audit cloud storage system that supports deduplication, ensure that user authentication is not restricted by a single certificate authentication center, and punish the server when the user's data is destroyed, charge a certain penalty and compensate for the damage. The user is an urgent problem to be solved.

This work starts from the user's immediate interests, and designs a cross-domain authentication and fair audit cloud storage system based on blockchain for the problem that the single certificate authentication center in the cloud environment is difficult to trust and the user data is vulnerable. The system solves the problem of the traditional cross-domain certificate relying on the single certificate authentication center, and uses the blockchain to realize the effective verification of the user certificate. At the same time, the system can punish the malicious server when the user data is destroyed, and collect a certain penalty and compensate. Users with impaired interests effectively protect the interests of users.

The system can meet the requirements of cross-domain authentication and fair data auditing, and effectively solves the problems of excessive trust of a single certificate authentication center and easy destruction of user data. Cloud servers with higher security will also be favored by users who need higher security. Therefore, the cross-domain authentication and fair audit cloud storage system based on blockchain has broad application prospects and is of great significance for building a more fair cloud storage system.

Keywords: PKI Cross-domain Authentication; Integrity Audit; Ciphertext Deduplication; Blockchain; Penalty Mechanism

目录

1 第一章 作品设计背景与功能介绍	1
1.1 设计背景	1
1.2 作品主要功能	2
1.3 系统组成	3
1.4 系统创新	4
2 第二章 设计与实现方案	5
2.1 实现原理	5
2.1.1 跨域认证	5
2.1.2 收敛加密(CE)与密文去重	7
2.1.3 数据完整性审计	8
2.1.4 智能合约	10
2.2 方案设计	11
2.2.1 认证模型	11
2.2.2 认证协议	13
2.2.3 用户数据加密与上传	14
2.2.4 数据完整性挑战	15
2.2.5 智能合约验证	15
2.2.6 用户数据下载与解密	16
2.3 软件流程	17
2.4 技术指标	20
3 第三章 系统测试与结果	21
3.1 测试方案	21
3.1.1 测试环境	21
3.1.2 测试方案	21
3.2 功能测试	22
3.2.1 认证功能	22
3.2.2 加密功能	27

3.2.3 上传功能	28
3.2.4 去重功能	29
3.2.5 下载功能	30
3.2.6 完整性审计功能	30
3.2.7 服务器惩罚功能	32
3.3 系统性能测试	34
4 第四章 应用前景与结论	36
5 第五章 附录	38
5.1 中国航天二院 706 所研究员 密码学前沿国防项目带头人 徐志华推荐信	39
5.2 区块链头部公司（专利数国内 TOP3），杭州复杂美公司副总裁 曹竞推荐信	40
5.3 项目团队成员曾在在知名企安全部门工作	41
5.4 项目成员的国家机构云服务安全漏洞证明	42
5.5 测试数据与结果	43

第一章 作品设计背景与功能介绍

1.1 设计背景

习总书记强调，没有网络安全就没有国家安全，而云计算作为目前个人企业乃国家机构应用服务的主要范式，其安全重要程度不言而喻。随着云计算的飞速发展，越来越多的用户和企业倾向于将数据外包存储在云服务器上同时使用云与其他用户交换分享数据。由于用户在交换数据之前往往需要认证身份，在云环境下传统的跨域认证方案所依赖的证书认证中心极易遭到破坏，这使得传统的跨域认证方案在验证过程中不仅效率低下而且存在巨大的安全隐患。另一方面，由于云服务器的各种软硬件故障和恶意敌手的存在，用户的数据可能被恶意篡改或者删除。

根据国际数据公司（IDC）的最新统计分析，全球产生和复制的数据以每 2 年翻一番的速度激增，到 2025 年，全球数据总量将达到 160ZB（1ZB = 230TB）。这些大量的数据将会给云服务器带来了前所未有的挑战。然而据报道，云服务器中存储的数据有高达 60% 是重复的，且数据冗余率随时间推移不断上升。

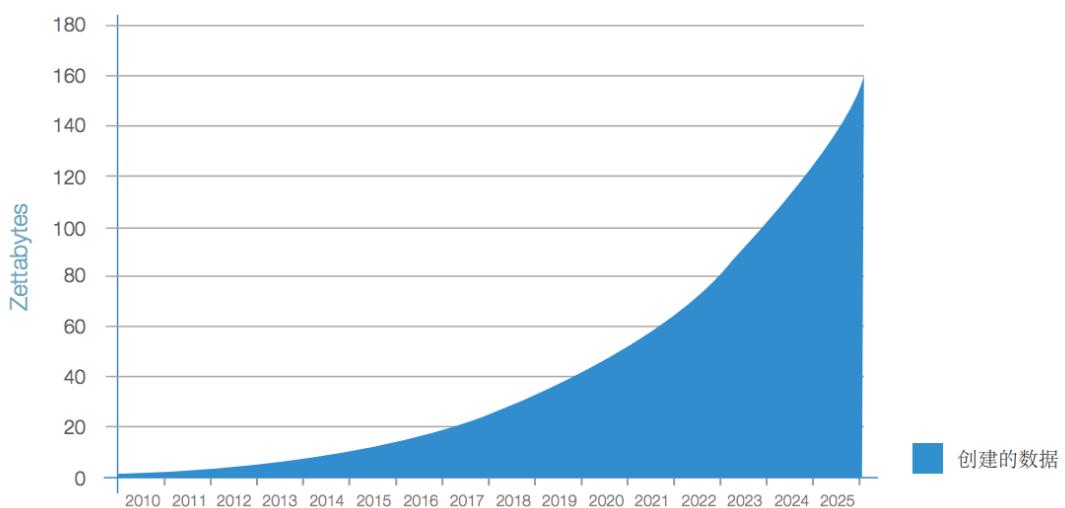


图 1.1 IDC 预测的全球数据总量发展趋势图

激增的数据经常成为黑客攻击的目标。每年全球范围都会发生多起严重的数据泄漏事件，严重威胁着公众与国家安全。



图 1.2 云安全事件频发

这一方面说明了目前网络犯罪十分猖獗，另一方面也说明了云存储的防护措施仍存在着很多不足，其中的认证安全，存储安全问题尤为严重。因此，如何设计兼顾安全隐私与效率的认证与数据存储管理方案，是云计算安全的关键所在。

1.2 作品主要功能

1. 实现基于区块链的跨域认证模型，通过本地生成证书，区块链存储的方式，有效解决传统跨域认证方案证书认证中心过度信赖的问题，同时大幅提高证书认证的效率。
2. 实现公平数据完整性审计系统的设计，通过批量审计功能，可以同时审计多个数据块的完整性，并使用概率性验证算法，可以通过验证少量数据块以较高的概率验证所有数据块的完整性。当存储在云服务器的用户数据被破坏时，可以惩罚已经收取高昂存储费用的云服务器，并补偿利益被损害的云用户。
3. 实现数据密文去重功能，防止云服务器存储大量的重复数据，避免用户和云服务器消耗大量的计算开销和存储开销。

1.3 系统组成

基于区块链的跨域认证与公平审计云存储系统由客户端、服务器端、第三方审计者（TPA）三部分组成。该系统包括了跨域认证、公平完整性审计、密文去重三个模块。在认证模块中，通过用户邮箱等标识字段，客户端自签名生成证书，经认证服务器验证后，证书指纹存入区块链网络，验证客户端则比对证书指纹等区块信息，验证证书是否有效。在审计模块中，主要实现的功能是快速审计用户数据的完整性，当用户存储在云服务器上的数据被破坏时，可以惩罚存储用户数据的云服务器，收取一定的罚金，并补偿给利益受损害的云用户。此外，本系统还支持密文去重功能，可以在保证用户数据隐私性的基础上避免存储重复数据，减轻用户与云服务器的计算开销和存储开销。

账号管理功能：通过本地客户端，利用账号密码生成证书，为个人或机构提供身份证明，用于身份验证，并提供常见的申请、修改、吊销、认证等基本操作。

查询验证功能：在区块链上查询一个证书的真伪，保证信任状态的实时透明。

回溯功能：查询一个证书或实体全部的历史操作，保障记录日志的真实有效，无法篡改。

加密功能：用户使用基于收敛加密的技术对上传的文件进行加密，在保证数据隐私性的同时为数据去重提供了保证，此外用户还生成用于数据完整性审计的辅助信息并存储于区块链中，使得第三方审计者可以在不需要下载原始数据的前提下对用户数据的完整性进行审计。

上传功能：用户使用网页登录账户，可以从计算机上任意选取文件进行上传，上传的文件将被存储在星际文件系统（IPFS）中。

去重功能：通过收敛加密的使用，相同数据被加密成相同的密文，因此云服务器可以判断用户数据是否已经存储在云服务器上，如果用户数据已经存储在云服务器上则不再需要用户上传数据，这将大大减轻云服务器的存储开销。

下载功能：用户的下载不再受时间、地域的限制，用户可以随时随地通过网页登录到 IPFS 下载需要的文件。

完整性审计功能：第三方审计者（TPA）将执行数据的完整性审计工作，服务器生成验证标签并提交，由第三方审计者（TPA）进行验证。

服务器惩罚功能：如果存储在云服务器上的用户数据遭到破坏时，智能合约将惩罚云服务器收取一定的罚金并补偿给数据被破坏的用户。



图 1.3 系统架构示意图

1.4 系统创新

本作品深度结合了区块链技术与多种复杂密码学技术的特点，不仅实现区块链项目的真正落地，而且充分发挥其不可篡改，去中心化等特性，为云存储安全提供自上而下的整体解决方案，研究深入底层，设计了全新的区块链证书认证模型与认证协议，数据完整性审计方式等，且各项性能指标超越以往传统方案，用户体验更加优良，减少用户成本。

第二章 设计与实现方案

2.1 实现原理

2.1.1 跨域认证

在云计算环境中，不同域间主体跨域请求服务时（如 A 公司想使用 B 公司的部分数据）的认证方式主要有两种：(1) 传统 PKI 方式，通过多级 CA 向上检索，直到请求到目标证书；(2) 两个主体共同信任一个可信的第三方。前一种方式中，CA 中心为不同用户的公钥与身份，建立了一一对应的关系，并使用自己的私钥进行签名，为实体颁发数字身份证件（CA 证书）。犹如一个数据库，记录了各种实体的身份信息。这样以来，两个实体之间请求认证时，无需实时交换公钥，仅需查看对方的 CA 证书，并到 CA 中心去查询真伪（或者本地策略默认信任该 CA 中心），就可以验证对方身份。但每个 CA 中心都有一定的作用域，不同域之间的用户无法直接认证。当 A 跨域认证 B 时，需要域 A 的用户请求到与本域 CA 有信任关系的 B 的证书，而 CA 中心的信任域没有囊括 B，所以需要到其他 CA 中心逐级查询验证，产生证书链，直到查询到可为 B 提供担保的 CA。

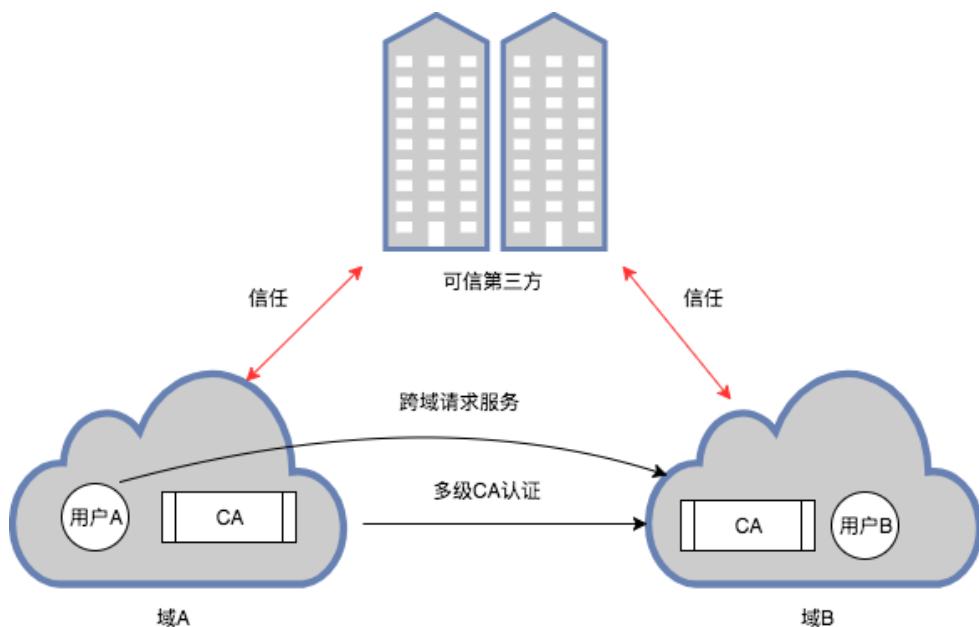


图 2.1 传统认证跨域关系图

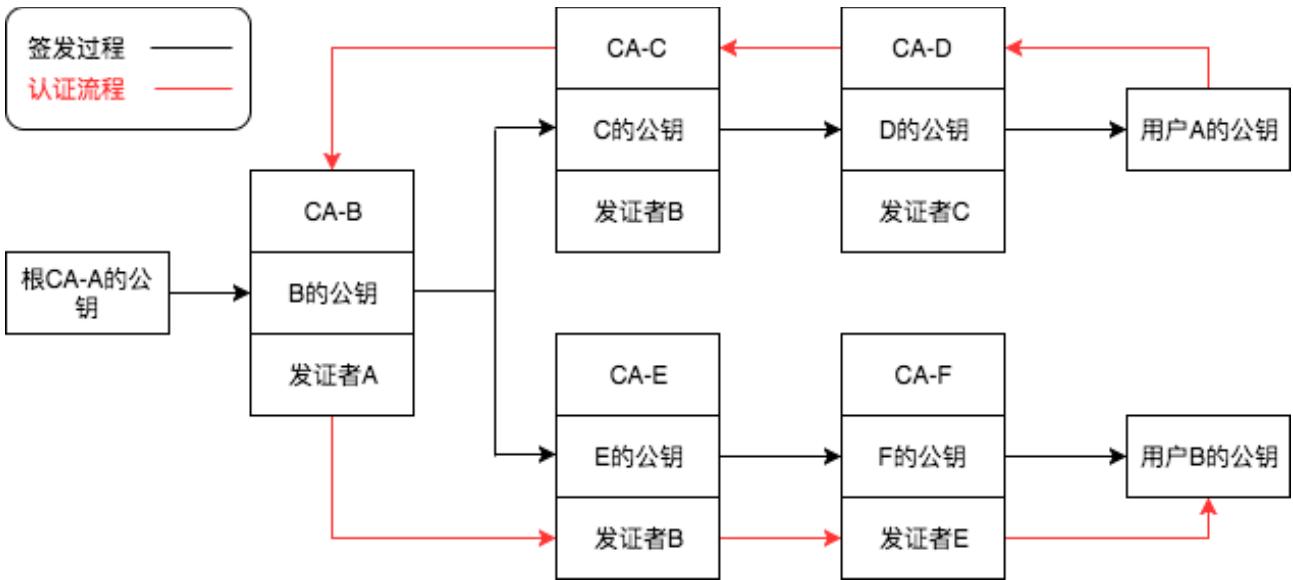


图 2.2 PKI 信任链

如图，每查询一次，就需要进行一次以上的非对称加解密与保密传输，从而大幅降低的认证效率。而证书链越长，系统风险则越大，任何一环（某个 CA 中心）都有被攻击的可能，从而使信任无法传递，最终导致信任错误，产生中间人攻击。此外，每级 CA 中心信任状态不会实时共享，比如 A 为 B 认证，B 为 C 认证，当 A 变为恶意节点时，下游的用户 C 无法获悉状态，仍然由 B 担保，但此时的 B 也是不可信的。而后一种方式使用较少，虽然避免的证书链问题，但信任的第三方机构仍然可能直接被入侵，导致认证失效。

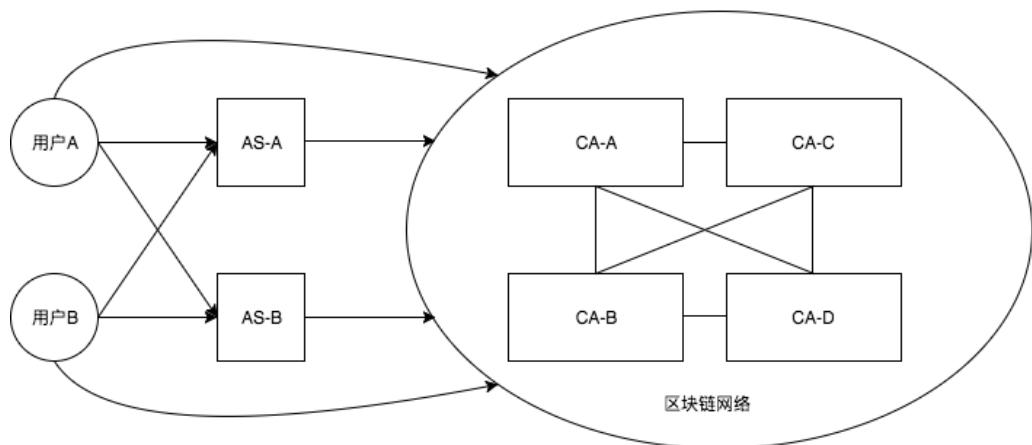


图 2.3 基于区块链的跨域认证模型

在我们的系统中，通过利用区块链分布式存储证书指纹，并通过比对证书完成验证。在系统中有以下角色：

AS (authentication server) : AS 为用户做身份验证功能，并审查用户通过客户端生成的证书是否真实有效，有效则将证书指纹提交给 CA。

CA (certificate authority): CA 即系统中的 CA 中心，负责管理证书指纹，检查证书信任状态，并存储在区块链网络中。

用户：需要认证服务的主体，可以是个人，私有云，机构等

区块链网络：组织起各个 CA 中心，共同维持统一的证书记录，是系统的核心所在，具体服务建立在区块链网络的顶层应用之上（以太坊智能合约），便于更新拓展与移植。

2.1.2 收敛加密(CE)与密文去重

为了节省存储空间，商业云服务提供商需要对云服务器存储的文件进行去重。举例来说，如果 Alice 想存储一个文件 M ，而 Bob 同时也请求存储一个相同的文件，云服务器对于 Bob 的请求不再存储第二份 M ，而是在数据库之中写入 Alice 与 Bob 都存储了文件 M 。这样一个被 n 个用户存储的文件的空间开销就由 $O(n \cdot |M|)$ 变为 $O(n + |M|)$ 。

用户希望云服务器能够完整的存储用户数据，但是由于服务器是不可信的：(1)云服务器忠实的执行去重复或审计操作，却对用户数据表示好奇；(2)云服务器会无意（如：硬件、软件故障）或者有意（如：进行数据挖掘）将用户数据透露给其他用户。所以用户上传至服务器的文件需在客户端进行加密，这给云服务器端的去重带来了挑战。

由此，为了在保护用户隐私的同时实现密文去重，收敛加密 (Convergent encryption, CE) 技术被 Douceur 等人提出。收敛加密从本质上讲是一种特殊的对称加密方案，它使用消息的哈希值作为加密密钥，从而可以保证不同用户加密相同消息总能得到相同密文。基于上述良好的性质，收敛加密已被广泛应用于安全数据去重等研究中。

过程中用到的密码学知识如下：

Hash 函数：又称为散列函数，它是一种单向密码体制，即它是一个从明文到密文的不可逆映射，可以将任意长度的输入经过变换以后得到固定长的输出。Hash 函数的安全性依赖于 3 个条件：单向性、抗弱碰撞性、抗强碰撞性。

1. 对任意给定的 Hash 值 z ，找到满足 $h(x)=z$ 的 x 在计算上是不可行的，这一性质也称为函数的单向性。
2. 已知 x ，找到 $y(y \neq x)$ 满足 $h(y)=h(x)$ 在计算上是不可行的，这一性质也称为抗弱碰撞性。

3. 找到任意两个不同的输入 x, y , $h(y) = h(x)$ 在计算上是不可行的, 这一性质也称为抗强碰撞性。

抗碰撞性 Hash 函数: 在抗碰撞 Hash 函数中, 要找出两个输入映射到一个输出上是困难和不可行的, 并且函数的输出比输入短。

收敛加密方案: 定义收敛加密方案 $CE = (CE.KeyGen, CE.Enc, CE.Dec, CE.Tag)$ 由以下四个算法组成。

1. $CE.KeyGen(M) \rightarrow K$: 密钥生成算法, 生成消息 M 的收敛密钥 $K = H(M)$ 。

其中, $H(\cdot)$ 表示密码学哈希函数。

2. $CE.Enc(K, M) \rightarrow C$: 确定性对称加密算法, 输入收敛密钥 K 和消息 M , 输出密文 C 。

3. $CE.Dec(K, C) \rightarrow M$: 对称解密算法, 输入密文 C 和收敛密钥 K , 输出对应明文 M 。

4. $CE.Tag(C) \rightarrow T_M$: 标签生成算法, 输入密文 C , 计算 T_M , 可用作文件去重标签。

在收敛加密方案之中, 原文 M 被由原文本身产生的密钥 K (如原文的哈希 $K = H(M)$) 加密, 加密结果将是唯一映射于原文的密文 $C = E(K, M) = E(H(M), M)$ 。原文 M 被加密之后, 客户端上传其至服务器, 并保留原文的哈希 $K = H(M)$ 用于之后的解密。虽然 CE 方案是一种对称加密算法, 但是大多数对称加密算法都不是确定性的, 它们大多数加入随机系数进行运算, 且每次加密得到不同的密文。CE 方案是加密算法为重复数据删除技术做出的一种“让步”, 这是因为数据加密和去重复是相互矛盾的。一方面加密使得数据呈现出随机性, 而另一方面重复数据删除建立在数据的相似性上。通过收敛加密, 相同明文能够加密得到相同密文, 从而为去重提供了可能。

假如两个用户刚好上传了相同的文件 M , 这样服务器可以在两个用户不需要相互协商的前提下知晓两份密文是相同的。基于云服务器对于每个加密文件都有唯一的文件标签与其对应, 因此通过将欲上传文件的文件标签与库中已有的文件标签进行查找与匹配就可以判断文件是否重复, 服务器就可以识别相同的密文, 仅保留一份进行存储, 既而进一步实现数据去重。

2. 1. 3 数据完整性审计

云计算环境下, 用户端的计算和存储能力都非常有限, 无法完成复杂的科学计算或密码

运算。外包计算通过将大部分复杂运算交由云服务器来完成，用户端只做少量的计算的方式，大大提高了计算效率，从而为用户提供多种多样的数据服务。目前，国内外许多公司和科研机构如 IBM、Google、Amazon 等都提供安全数据存储和外包计算服务。

作为其中一个重要的分支，外包数据库（ODB, outsourced database）近年来越来越吸引学术界的广泛关注。早在 2002 年，Hakan Hacigumus 等人就含蓄地引入了外包数据库的概念。在外包数据库模型中，为了节省昂贵的数据库管理成本，数据拥有者将数据库在本地进行加密运算并将密文数据库外包给云服务器来管理。云服务器提供数据库访问所需的一切软硬件资源，确保在收到用户数据访问请求后，执行数据库检索并返回相应的结果给用户。这种模式下，既降低了数据拥有者的数据库维护开销，又可以为用户提供高质量的数据访问服务。

然而，外包数据和外包数据库在为人们带来诸多益处的同时，也不可避免地面临着一些新的安全挑战。首先，在云计算环境下，找到一个完全可信的云服务器几乎是不可能的。而外包数据往往包括一些不能泄露给云服务器的敏感信息。因此，一个主要安全挑战是外包数据的秘密性：即云服务器不能知道存储的数据的内容。其次，由于云服务器是不完全可信的，出于自身经济利益（节省网络带宽和计算量）的驱动或者软硬件运行故障，它可能会破坏或者删除用户不经常访问的数据。然而如果需要下载所有的数据然后逐一验证其完整性，这将带来大量的通信开销和计算开销，这是用户无法接受的。因此，另一个重要的安全挑战是如何在不下载用户数据的前提下审计用户数据的完整性。

由此，为了验证用户数据的完整性，可证明数据拥有（Provable Data Possession, PDP）技术被 Ateniese 等人提出。PDP 技术可以有效的在不下载原始数据的前提下验证数据的完整性，此外，如果每一次都验证所有的用户数据来保证数据的完整性则将会消耗大量的计算资源。PDP 方案设计了概率性验证算法，它通过检测随机的数据块从而以较高的概率验证全部数据的完整性，通过 PDP 技术的使用，这将大大提高数据完整性验证的效率。

可证明数据拥有方案：定义可证明数据拥有方案 $PDP = (PDP.KeyGen, PDP.TagBlock, PDP.GenProof, PDP.CheckProof)$ 由以下四个算法组成。

1. $PDP.KeyGen(1^k) \rightarrow (pk, sk)$: 密钥生成算法，用于生成用户的公钥和私钥。
2. $PDP.TagBlock(pk, sk, m) \rightarrow T_m$: 标签生成算法，输入公钥 pk , 私钥 sk 和一个文件 m , 生成用于验证的标签 T_m 。
3. $PDP.GenProof(pk, F, chal, \Sigma) \rightarrow V$: 证据生成算法，输入公钥 pk , 文件 F , 挑战信息

$chal$ 和序列集合 Σ ，输出证据 V 。

4. $PDP.CheckProof(pk, sk, chal, V) \rightarrow \{\text{"success"}, \text{"failure"}\}$: 验证算法，输入公钥 pk ，私钥 sk ，挑战信息 $chal$ 和证据 V ，验证数据完整性，若果数据完整性验证通过输出"success"，失败输出"failure"。与

2.1.4 智能合约

智能合约的概念最先由密码学家尼克·萨博提出，其定义为“一个智能合约是一套以数字形式定义的承诺，包括合约参与方可以在上面执行这些承诺的协议”。它是运行在可复制、共享的账本上的计算机程序，同时它自己也是一个系统参与者，它对接收到的信息进行回应，它可以接收和储存价值，也可以向外发送信息和价值。随着比特币、以太坊等数字货币的兴起，智能合约被广泛讨论和研究。基于以太坊的智能合约由事件驱动，具有状态，运行在一个可复制可分享但不可篡改的账本（区块链）之上，并且能够保管和转移账本上的资产。

基于区块链的安全智能合约包括事务处理和保存的机制，以及一个完备的状态机，用于接受和处理各种智能合约；并且事务的保存和状态处理都在区块链上完成。事务主要包含需要发送的数据；而事件则是对这些数据的描述信息。事务及事件信息传入智能合约后，合约资源集合中的资源状态会被更新，进而触发智能合约进行状态机判断。如果自动状态机中某个或某几个动作的触发条件满足，则由状态机根据预设信息选择合约动作自动执行。

智能合约系统根据事件描述信息中包含的触发条件，当触发条件满足时，从智能合约自动发出预设的数据资源，以及包括触发条件的事件；整个智能合约系统的核心就在于智能合约以事务和事件的方式经过智能合约模块的处理，出去还是一组事务和事件；智能合约只是一个事务处理模块和状态机构成的系统，它不产生智能合约，也不会修改智能合约；它的存在只是为了让一组复杂的、带有触发条件的数字化承诺能够按照参与者的意志，正确执行。

智能合约由编程语言而不是法定语言记录，当被发布到区块链上之后，会存储在区块链上的一个特定地址，不能被篡改，由以太坊虚拟机解释执行。在我们的方案模型中，用户与云服务器是智能合约的参与方，在云服务器给用户提供数据存储服务之前，双方约定存储协议并以智能合约的形式编写出来，然后双方都仔细检查和测试代码，确信不存在后门或者恶意漏洞，最后部署到区块链上。在我们的方案设计中，用户数据的完整性验证结果取决于云服务器提供的数据拥有性证明和第三方审计者的验证，并且由第三方审计者反馈验证结果给智能合约，智能合约根据约定承诺对云服务器账户做出相应处理并且记录用户向云服务器挑

战的验证结果。例如，如果用户的数据完整性验证失败，则会自动从云服务器账户向用户账户转入一定金额的数字货币，实现对云服务器的惩罚和对用户的经济补偿。

2.2 方案设计

2.2.1 认证模型

与传统证书类似，本作品的对区块链证书的操作有注册，验证，颁发，更新，注销。

区块链证书的参数与 X.509v3 标准基本相同，但附加了证书 ID，用作身份标示，从而与传统证书区分开。过程中用到的符号说明如下：

- $\text{sig}(sk, \mu)$: 通过消息 μ 和私钥 sk 生成签名 σ
- $\text{Hash}(\mu) \rightarrow \theta$: 生成消息 μ 的哈希值 θ
- $A \rightarrow B$: A 发送给 B 一个请求
- $\text{Func_Gen}() \rightarrow Bcert$: 客户端生成区块链证书（使用更改之后的openssl证书工具）
- $\text{ver}(pk, \sigma, \mu) \rightarrow b \in \{0,1\}$: 利用公钥 pk 验证消息 μ 签名 σ ，验证通过返回 0，否则返回 1

进程一：证书申请

用户通过客户端本地生成区块链证书，然后发送到CA处

1. 用户本地生成证书: $\text{Func_Gen}() \rightarrow Bcert$
2. 用户向CA发送信息: $User \rightarrow CA: \{\theta, application, info, values = (pk, \theta)\}$

其中 $\theta = \text{sig}(sk, \text{Hash}(Bcert) \parallel info)$ ， σ 是作者用私钥对 $\text{Hash}(Bcert \parallel info)$ 的签名

3. CA对时效性，签名等验证验证通过后，从证书中取出邮箱做hash运算， $m = \text{Hash}(email)$ ，生成一个唯一的ID，连同ID, m, Bcert一同存入区块链。

进程二：证书验证

CA接收到用户的请求后，验证证书的真实性与完整性，不通过则返回error，通过即写入区块链。

- 1.查询区块，确定本证书没有被注册过
2. $\text{ver}(\text{pk}, \sigma) = 1$ ，通过验证，通过共识算法，生成区块，在区块中写入证书指纹与证书状态等信息。最后广播到区块链网络中。

由于各个CA是区块链网络中的节点，共同维持统一的状态，证书信息同步成功。

进程三：证书更新

在这种情况下，用户更新证书即需要更换密钥对，而不改变证书 ID，这里我们通过签名验证的方式，更换密钥对。类似证书申请，用户同样在客户端生成一个区块链证书。

1. $\text{Func_Gen}() \rightarrow Bcert_{new}$
2. $\text{Hash}(Bcert_{new} \parallel pk^{new})$
3. $User \rightarrow CA : \{ID_{old}, Bcert_{new}, update, values = (pk^{new}, pk^{old}, \sigma_1, \sigma_2)\}$

其中：

$$\sigma_1 = \text{sig}(sk^{old}, \text{Hash}(Bcert_{new} \parallel pk^{new}))$$

$$\sigma_2 = \text{sig}(sk^{new}, \text{Hash}(Bcert_{new}))$$

σ_1 保证了证书拥有者知道与旧公钥 pk^{old} 对应的旧私钥

$\text{Hash}(Bcert_{new} \parallel pk^{new})$ 保证新的证书与私钥不被篡改

随后，重复进行进程二的证书验证，更新证书。

进程四：证书注销

已有证书的用户，当需要注销证书时，向CA发送注销信息

$$User \rightarrow CA : \{ID, revocation, info, values = (pk, \sigma)\}$$

$$\sigma = \text{sig}(sk, \text{Hash}(Bcert \parallel info))$$

重复进行进程二证书验证，通过后写入区块链网络，将证书信息改为已经注销。由于区块链的不可篡改，单向增长的特点，系统将以最新的证书状态为准，从而避免冒充身份。

2.2.2 认证协议

基于上述认证模型，我们设计了特有的协议来完成具体的认证功能，流程如下图所示。

其中主要流程说明如下：

$U_A \rightarrow AS_B$: 用户 U_A 请求 AS_B 为其认证，以访问域 B 的服务。

$AS_B \rightarrow U_A : \{N\}$: AS_B 返回一个随机数 N ，防止重放攻击。

$U_A \rightarrow AS_B : \{Cert, sig_{sk}(N), N\}$: 用户 U_A 发送给 AS_B 自己在域 A 的证书，对随机数的签名，用于 AS_B 进行验证。

$AS_B \rightarrow U_A : \{CertB, sig(CertB)\}$: 当验证证书操作完成后 U_A 会得到 AS_B 颁布的证书 $CertB$ 。

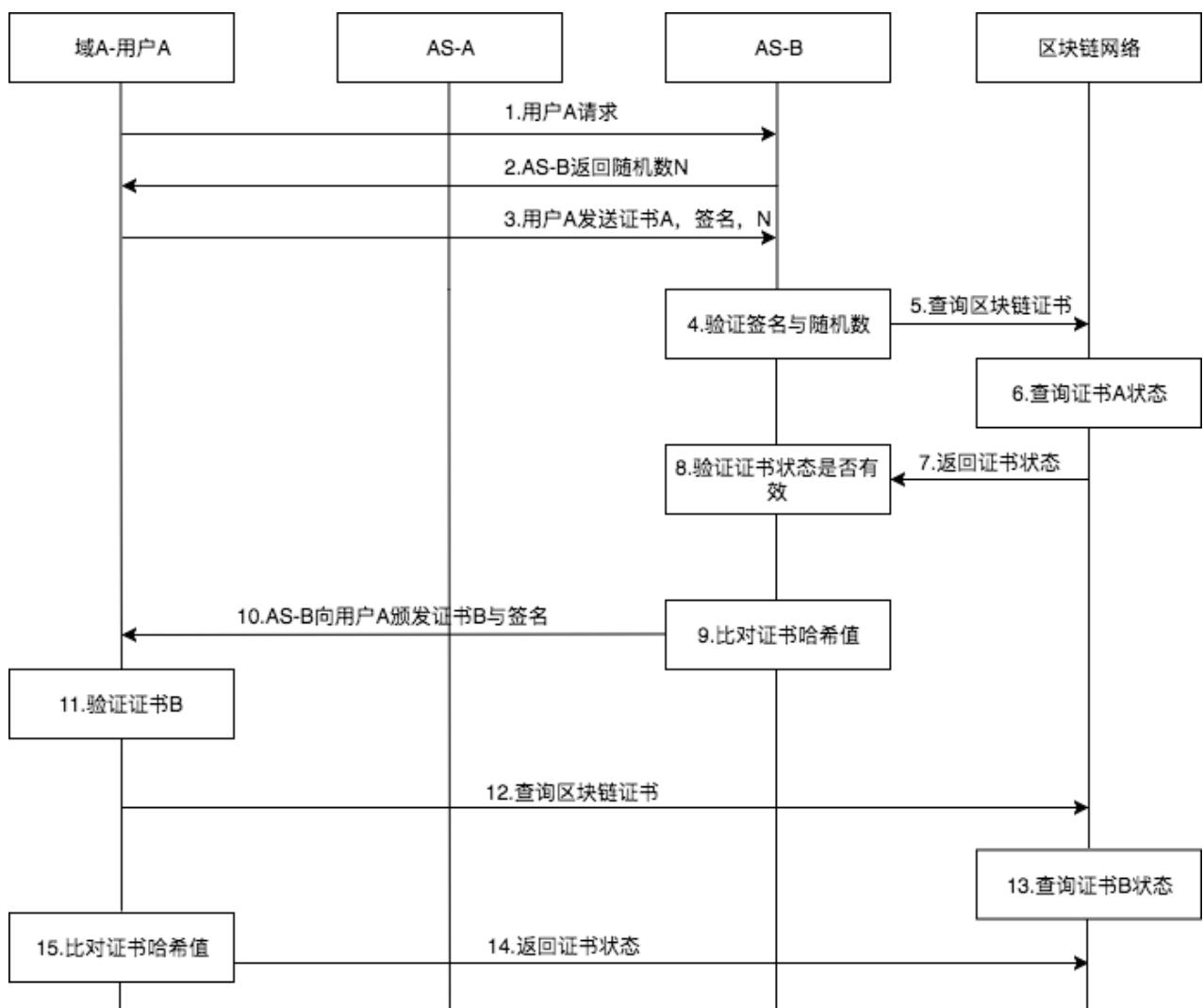


图 2.4 认证协议

2.2.3 用户数据加密与上传

用户数据加密与上传模块包含数据加密与上传两部分。其中，数据加密部分采用收敛加密技术实现了对用户上传数据的加密，其实现方案主要由 MD5 和 AES 算法构成；文件上传到云服务器后，由云服务器对密文进行比较去重，不需要用户再次上传重复的数据，这样可减少由于重复文件带来不必要的存储开销和通信开销。

当用户 Alice 拥有某一本地文件 M ，并希望将其上传至云服务器时，系统首先利用 MD5 求得其哈希值作为密钥 $K = H(M)$ ，再用密钥 K 对文件进行加密得到密文 $C = E(K, M)$ 。上传到云服务器上后，云服务器对密文 C 生成标签 $tag = H(c)$ ，如果云服务器已经存在标签 tag 说明服务器已经拥有数据 C ，此时就不再需要用户再次上传重复的数据。整个系统中，文件标签 tag 承担着标识、区分文件的作用，每一个文件的标签 tag 都是独一无二，且都唯一对应着相应的文件。

加密上传过程中，客户端首先计算文件的密钥 K 及其密文 M ，然后将密文分割为 n 个文件： $F = (m_1, m_2, \dots, m_n)$ ，同时生成公私钥对 (pk, sk) ，其中 $pk = (N, g)$, $sk = (e, d, v)$ 。再利用 $TagBlock(pk, (d, v), m_i, i)$ 生成文件完整性验证标签 (T_i, W_i) 。

其中 $W_i = v||i$, $T_i = (h(W_i) \times g^{m_i})^d \bmod N$ ，并将 $pk, F, \Sigma = (T_1, T_2, \dots, T_n)$ 发送至云服务器端进行文件上传请求。服务器对上传文件及已存储文件的标签进行比较，若云服务器已存储了相同文件标签，说明服务器已拥有相同的数据，不需要用户再次上传，从而实现去重功能；若不存在相同的文件标签，则需要用户上传数据。

用户数据加密上传示意图如图 2.5 所示。

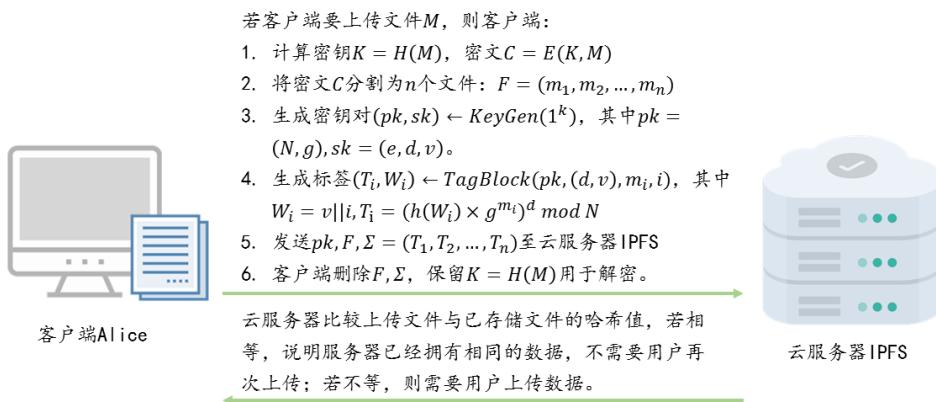


图 2.5 用户数据加密与上传

2.2.4 数据完整性挑战

数据完整性挑战模块主要由可证明数据拥有(PDP)技术实现，可有效地在不下载原始数据的前提下验证数据的完整性，通过检测随机的数据块，能以较高的概率验证全部数据的完整性，这将大大的提高数据完整性验证的效率。

当用户 Alice 指定对一文件进行完整性挑战时，首先随机输入几个验证块编号 $chal = [j_1, j_2, \dots, j_c]$ ，并随机产生一个生成元 s ，计算 $g_s = g^s \bmod N$ ，然后将 $chal, g_s$ 发送至服务器。服务器接受挑战后，查找读取该文件对应的密文 $F = (m_1, m_2, \dots, m_n)$ 及标签 $\Sigma = (T_1, T_2, \dots, T_n)$ ，并计算 $T = T_{j_1} \times T_{j_2} \times \dots \times T_{j_c}$, $\rho = H(g_s^{m_{j_1} + m_{j_2} + \dots + m_{j_c}} \bmod N)$ ，生成证据 $V = (T, \rho)$ $V = (T, \rho)$ ，并发送至第三方审计者进行验证。数据完整性挑战过程示意图如图 2.6 所示。

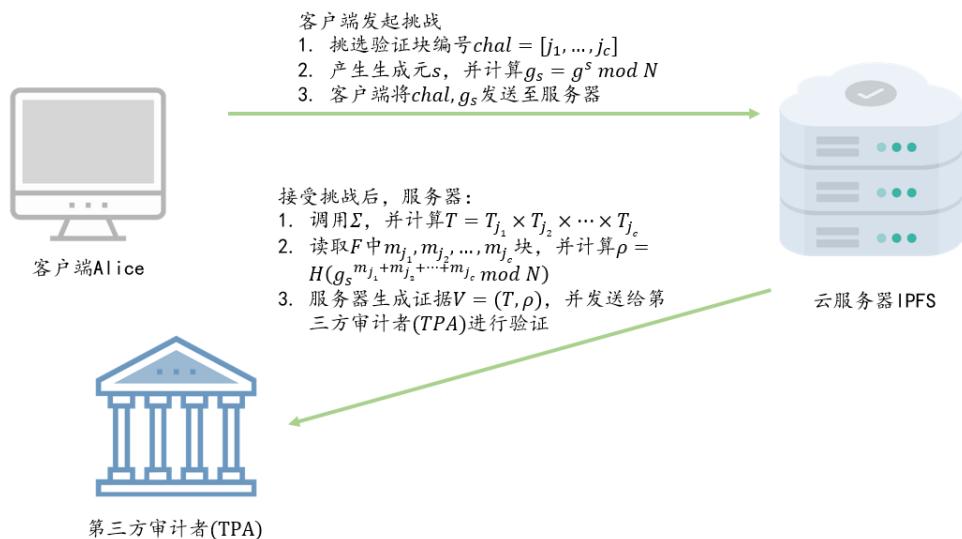


图 2.6 数据完整性挑战

2.2.5 智能合约验证

验证模块及惩罚处理机制由智能合约实现，用户与云服务器是智能合约的参与方，在云服务器给用户提供数据存储服务之前，双方约定存储协议并以智能合约的形式编写出来，然后双方都仔细检查和测试代码，确信不存在后门或者恶意漏洞，最后部署到区块链上。一经部署，不能被篡改，由以太坊虚拟机解释执行。

进行智能合约验证之前，要求第三方审计者完成完整性审计工作，计算

$$\tau = \frac{T^e}{h(W_{j_1}) \times h(W_{j_2}) \times \dots \times h(W_{j_c})} \bmod N$$

若 $H(\tau^s \bmod N) = \rho$ ，则完整性验证通过；若二者不

相等，则验证不通过，并将验证结果反馈给智能合约，智能合约根据约定承诺对云服务器账户做出相应处理并且记录用户向云服务器挑战的验证结果。若完整性验证不通过，则对服务器进行惩罚，自动从云服务器向用户转账一定金额的数字货币，实现对云服务器的惩罚和对用户的经济补偿。智能合约验证模块示意图如图 2.7 所示。

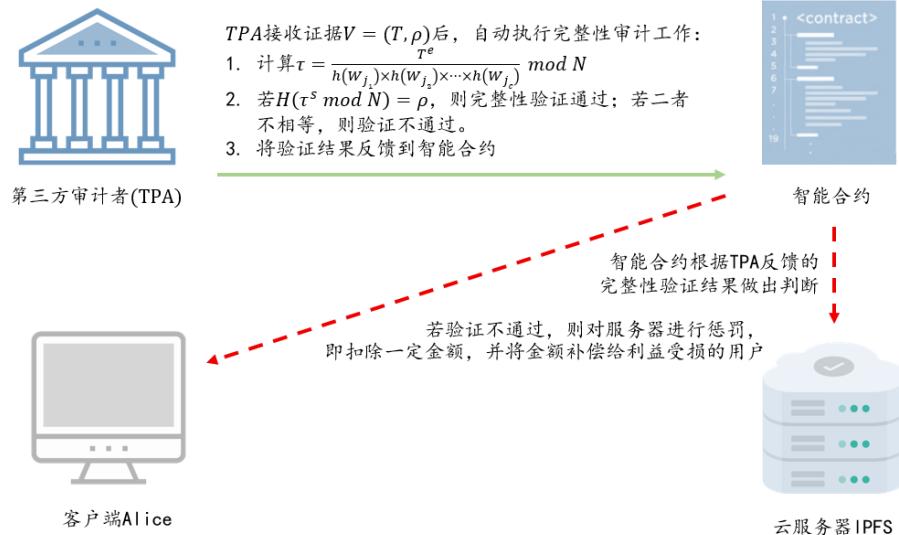


图 2.7 智能合约验证

2.2.6 用户数据下载与解密

用户数据下载与解密模块对应于用户数据上传与加密模块。当用户指定请求下载一文件时，服务器根据请求查找并返回分割后的密文块 $F = (m_1, m_2, \dots, m_n)$ ，客户端接收 n 个密文块并合并为完整密文 C ，再调用存储在本地的密钥 K ，解密得明文 $M = D(K, C)$ 。

用户数据下载与解密模块示意图如图 2.8 所示。

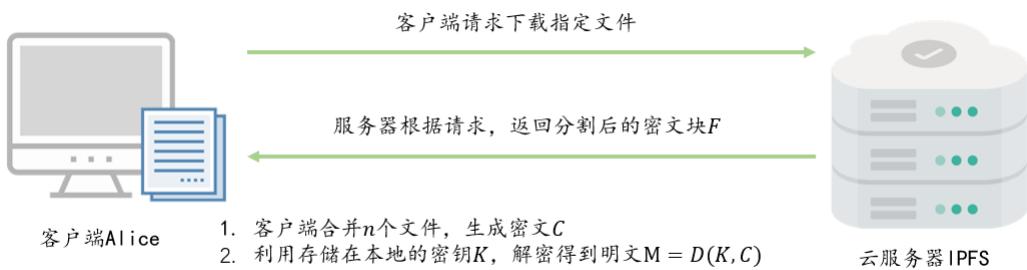


图 2.8 用户数据下载与解密

2.3 软件流程

本系统将认证功能、上传功能、去重功能、下载功能、完整性审计功能和服务器惩罚功能实现在证书验证、数据加密与标签生成、数据完整性挑战与审计、智能合约验证、数据解密五个流程中。

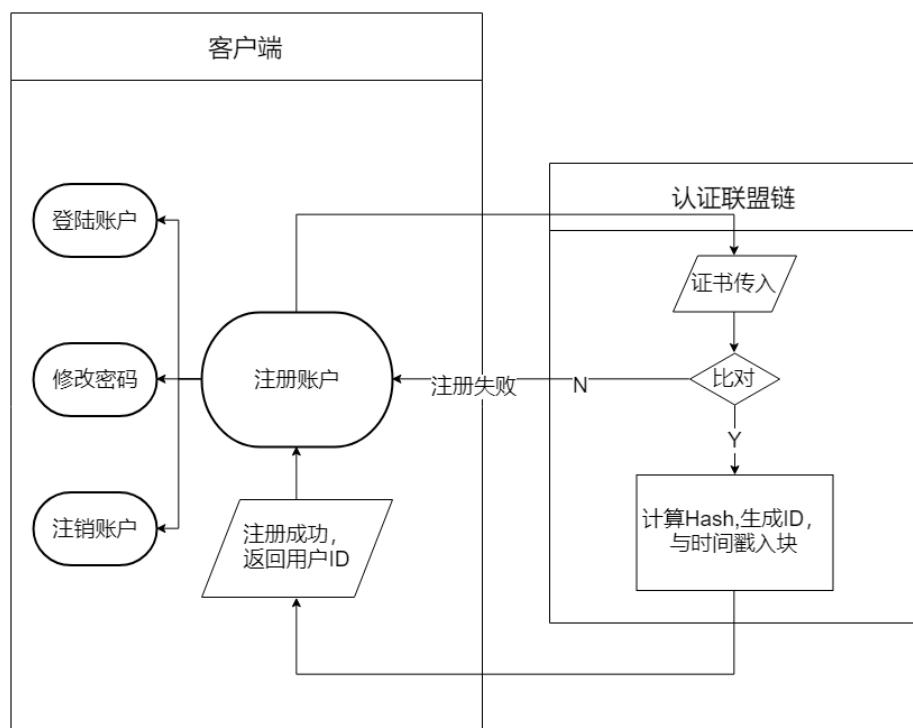


图 2.9 用户注册与认证

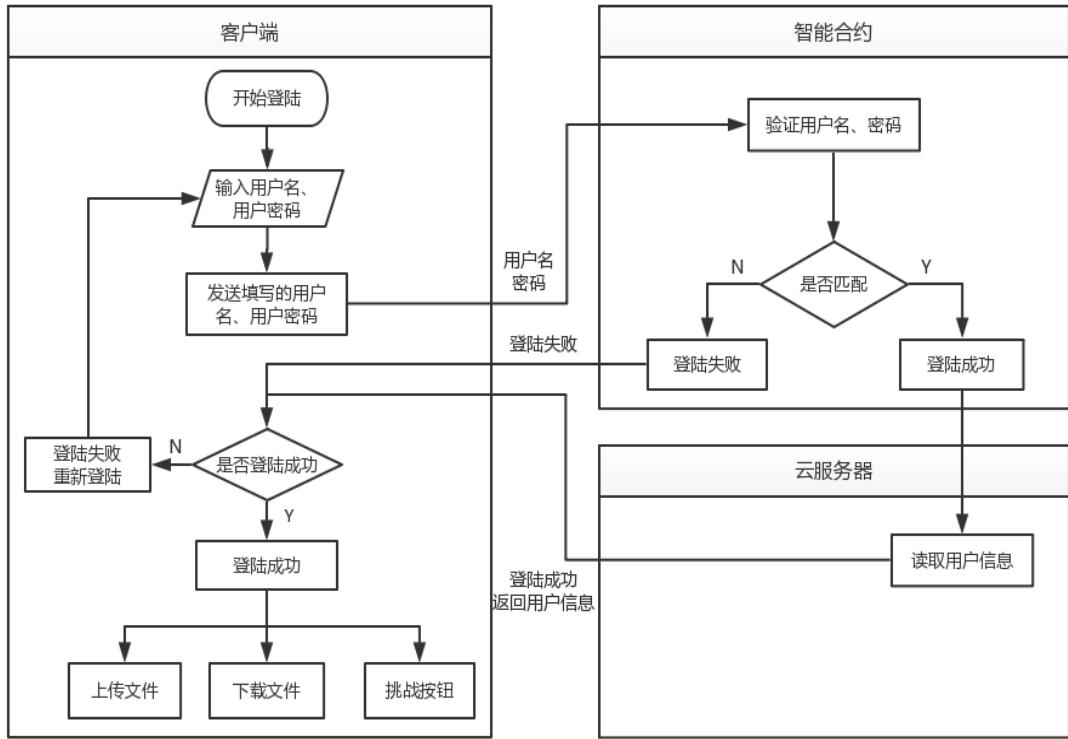


图 2.10 用户登录

数据加密与重复检测流程图如图 2.11 所示。

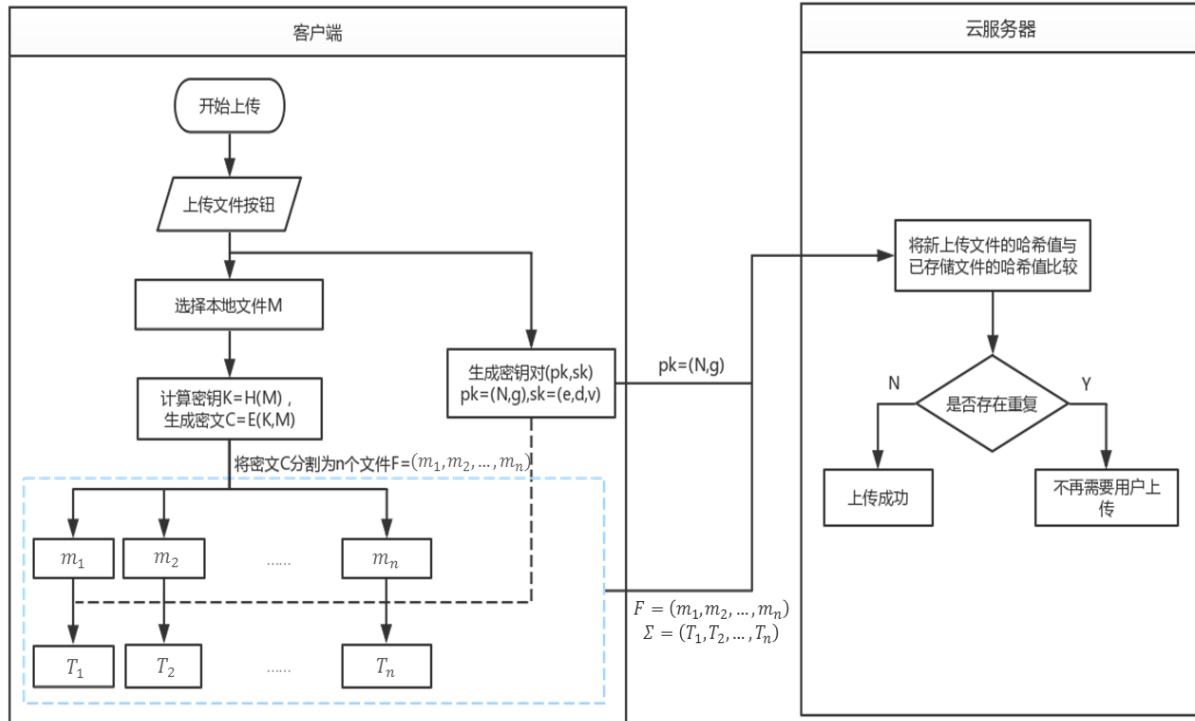


图 2.11 数据加密与重复检测

数据完整性审计与智能合约验证流程图如图 2.12 所示。

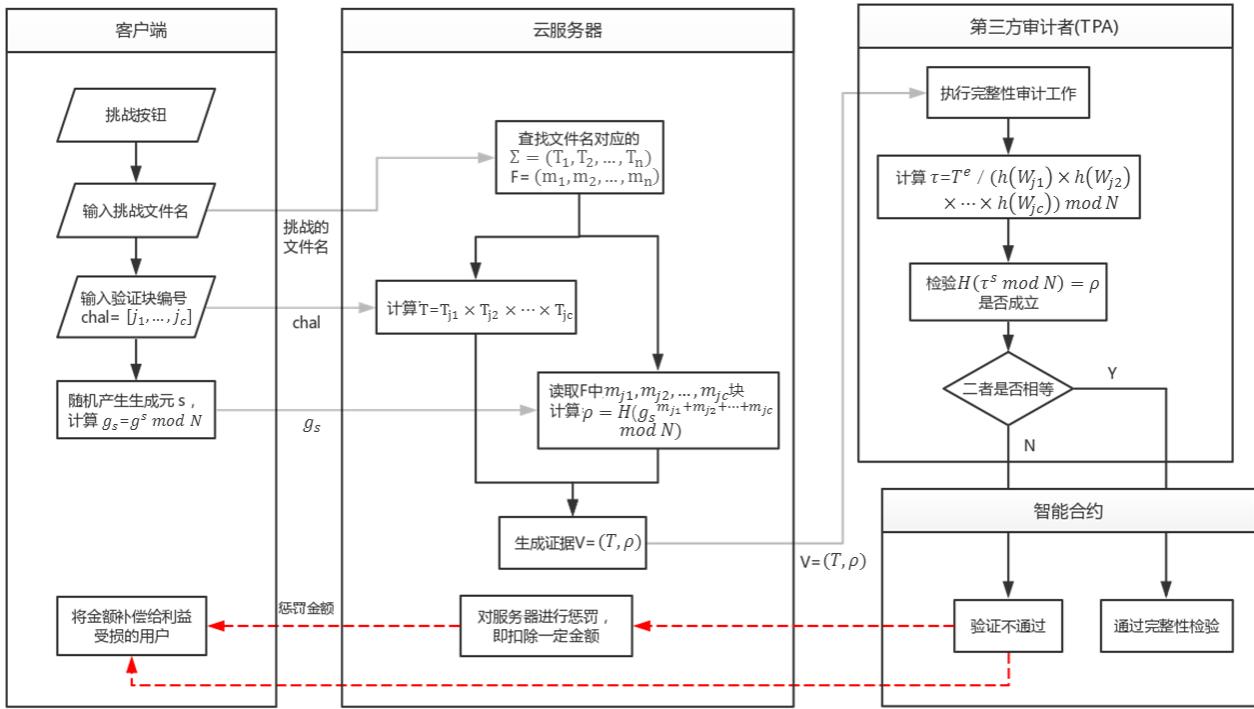


图 2.12 数据完整性审计与智能合约验证

数据下载与解密流程图如图 2.13 所示。

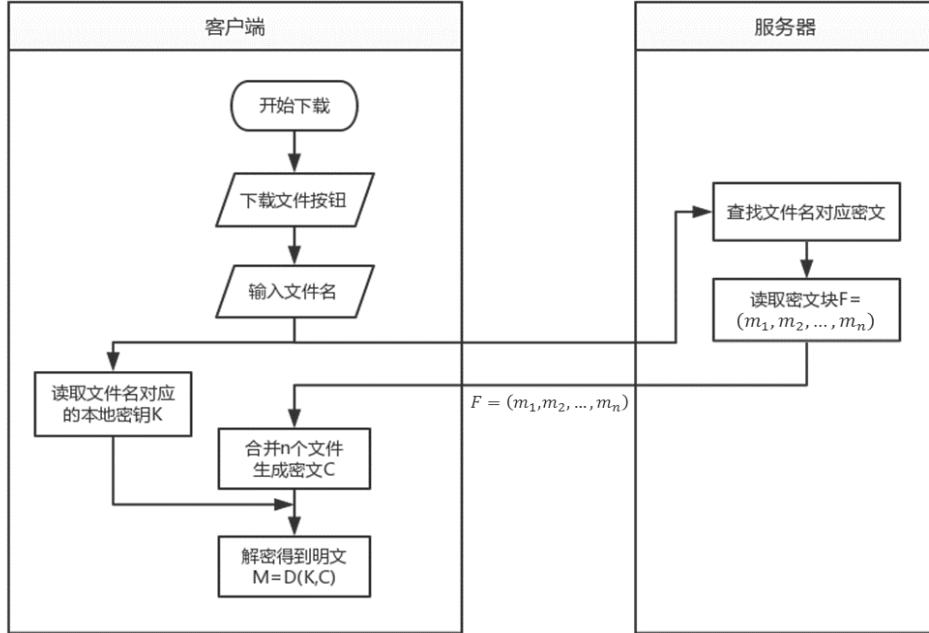


图 2.13 数据下载与解密

2.4 技术指标

1.认证效率

在一定的认证次数中，系统网络流量大小，在测试环境中，开销约为 0.0012Mbytes/次

2.数据加密速度

数据加密速度，就是用户数据在本地使用收敛加密技术执行加密操作的时间，在测试环境下目标 36Mbps。

3.服务器生成证据速度

服务器在生成证据速度，就是在服务器接收到用户的完整性验证挑战之后计算证据所需的时间，在测试环境下目标 0.18Mbps。

4.完整性审计速度

完整性审计速度，就是在服务器生成证据之后由第三方审计者（TPA）验证数据完整性的
时间，在测试环境下的目标为 0.002s。

5.数据解密时间

数据解密速度，就是用户数据在本地执行解密操作的时间，在测试环境下目标为 35Mbps。

6.完整性审计概率

完整性审计概率，由于本系统完整性审计采用概率性验证算法，即通过验证部分数据的
完整性以较高的概率保证所有数据的完整性，验证概率目标为 99%。

第三章 系统测试与结果

3.1 测试方案

3.1.1 测试环境

客户端	第三方审计者	服务器
操作系统: Microsoft Windows [版本 10.0.14393]	操作系统: Microsoft Windows [版本 10.0.14393]	操作系统: Linux [x86 64,kernel version:4.4.0-31-generic]
处理器: Intel(R) Core(TM) i3- 2310M CPU @ 2.10GHz	处理器: Intel(R) Core(TM) i3- 2310M CPU @ 2.10GHz	处理器: Intel(R) Core(TM) CPU i3-3240 @3.40GHz, 4 core
内存: 4G	内存: 4G	内存: 4G
硬盘: Fixed hard disk media	硬盘: Fixed hard disk media	硬盘: Toshiba Dt01aca050
编程语言: Html,Css,Javascript	编程语言: Python	编程语言: Python, Solidity

3.1.2 测试方案

选取三台 PC,一台作为客户端, 一台作为服务器, 一台作为第三方审计者 (TPA)。用户数据加解密算法和标签生成算法均使用 Python 语言编写; AS, CA 服务器生成验证证据算法和 TPA 验证算法也都使用 Python 语言编写; 用户界面和服务器界面均使用 JavaScript 实现; 惩罚机制与区块链网络通过基于区块链的安全智能合约实现。我们在 3.2 节进行功能测试, 主要测试系统的各个功能是否可以实现, 3.3 节主要进行性能测试, 主要测试数据加密时间、服务器生成证据时间、完整性审计时间和数据解密时间。在 3.4 节我们给出测试的数据和结果。

3.2 功能测试

3.2.1 认证功能

对于需要认证的用户，先通过注册功能，输入自己的相关信息，生成证书。

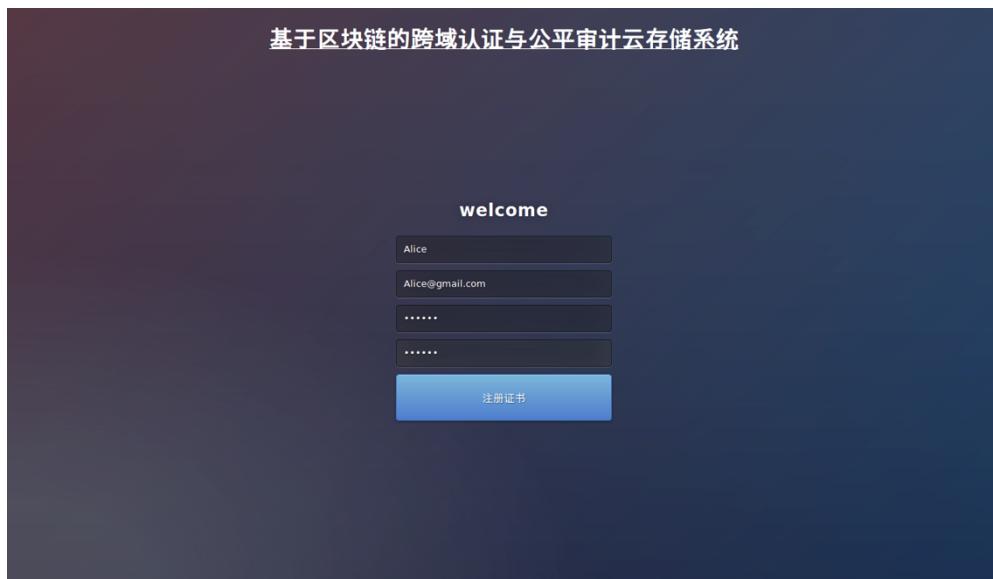


图 3.1 数据下载与解密



图 3.2 生成证书

查看本地证书详情信息，账户 ID



The screenshot shows a certificate management interface. On the left, there is a detailed view of a certificate for "Alice". It includes fields for Subject Name (CN: Alice, ST: Some-State, O: Internet Widgits Pty Ltd), Issuer Name (CN: Alice), and Issued Certificate details (Version: 3, Serial Number: 00 8B 3F 29 B6 03 32 84 90, Not Valid Before: 2018-10-23, Not Valid After: 2019-10-23). It also lists Certificate Fingerprints (SHA1: E8 C9 E9 66 3C CB 65 E7 24 21 E7 B7 DA 93 9C 48 AB 9F 6A 99, MD5: B7 3E FB A6 89 24 AF 40 F7 B2 F1 6A AE 51 38 99) and Public Key Info (Key Algorithm: RSA, Key Parameters: 05 00, Key Size: 2048, Key SHA1 Fingerprint: 2E 0F 9E 89 BB 02 D7 D9 14 FF DB 4C FE 33 1D 2E 11 8B 86 A7, Public Key: long hex string).

On the right, there is a file manager interface showing two files: "certificate.crt" (represented by a document icon) and "private.key" (represented by a key icon).

图 3.3 基于 x.509v3 的区块链证书

证书生成后，通过服务器校验后，写入区块链中。



图 3.4 区块链网络中证书信息

当客户需要更新证书时，需要生成新私钥，更改密码。



图 3.5 通过更改密码生成新私钥

更新证书后，用户身份 ID 不变，更新证书其他信息。

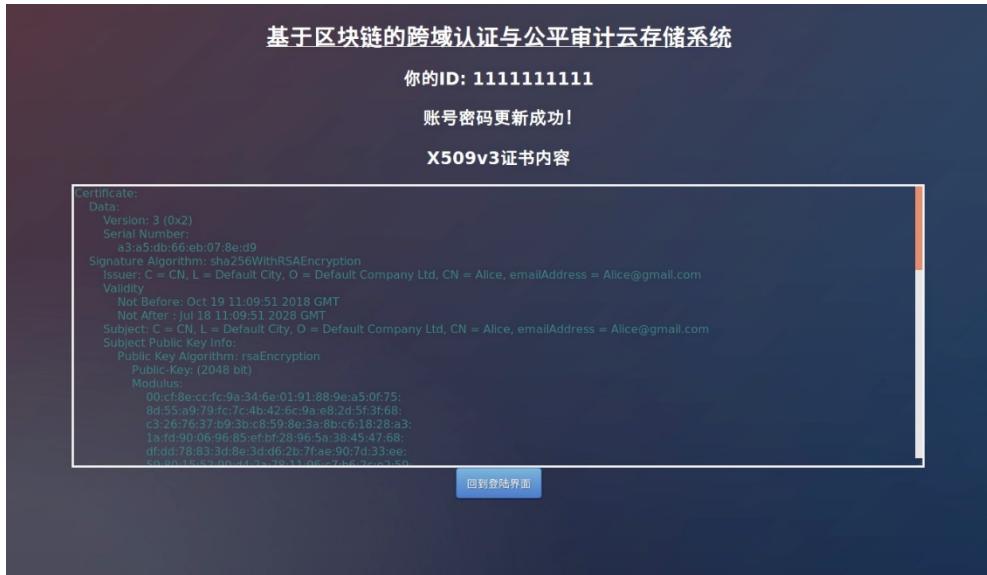


图 3.6 更新后的证书信息

可以看到，在区块链网络中，证书已经更新。



图 3.7 更新后的区块链状态

最后，当用户由于某些原因，需要注销证书时，通过验证 ID 密码，向 CA 发送注销请求。

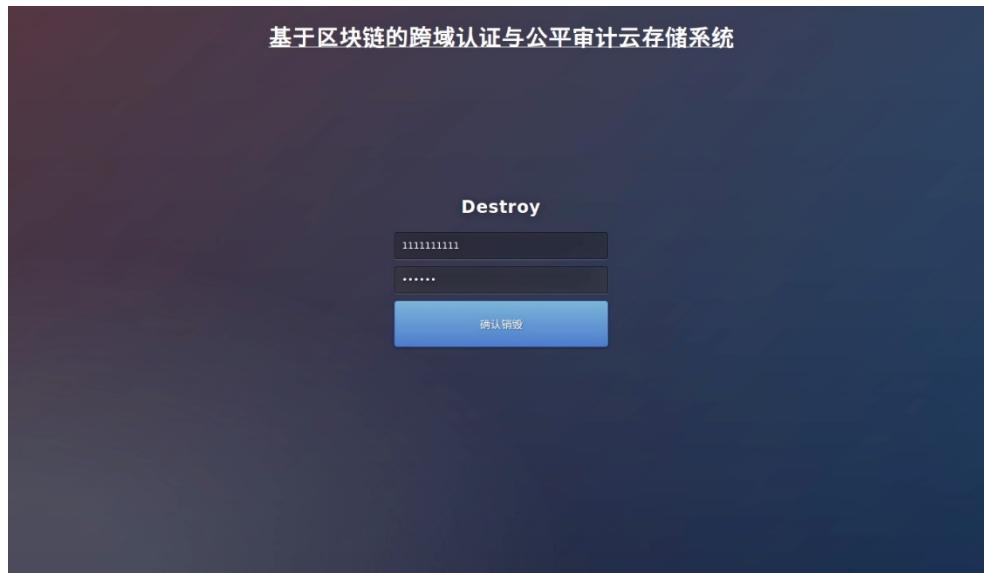


图 3.8 用户注销选项

最终，该证书在区块链中的状态变为 destroy，成功注销。

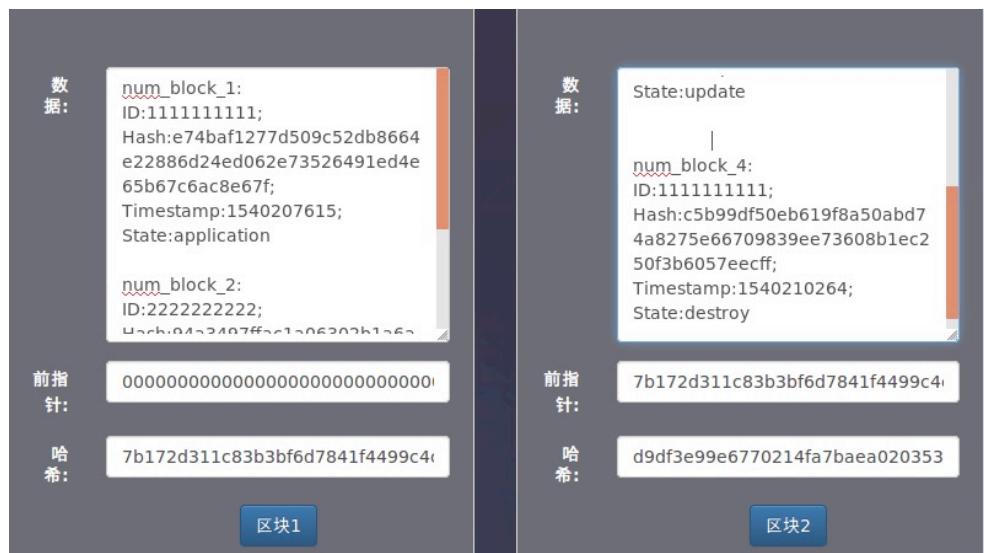


图 3.9 被注销的证书

3.2.2 加密功能

为了保护外包数据的隐私性，用户在将数据外包存储到云服务器之前，使用收敛加密技术对文件进行加密。测试加密功能所使用的明文数据如图 3.10 所示。

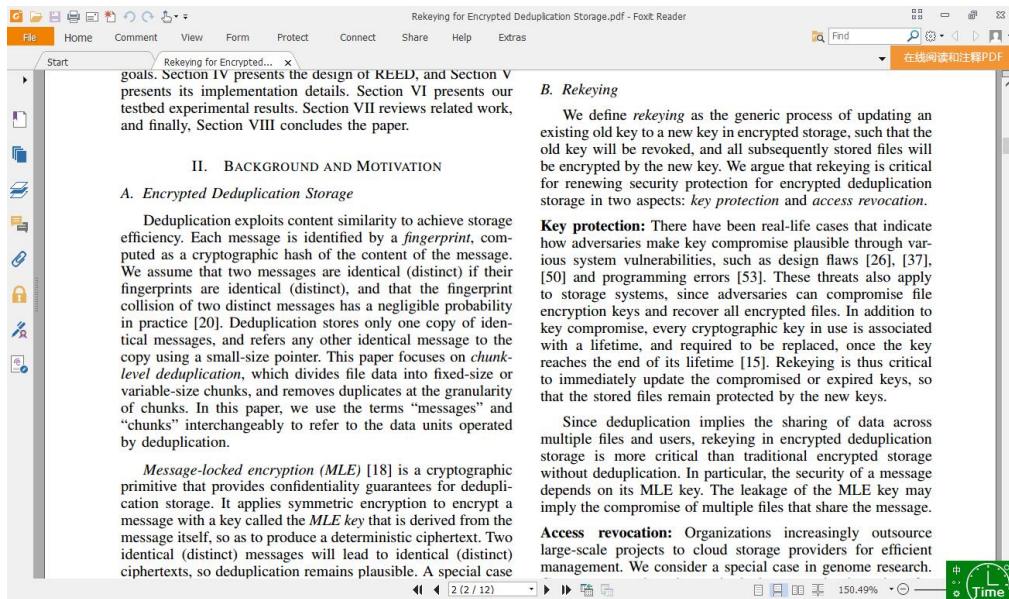


图 3.10 加密前的明文数据

收敛加密在保证用户数据隐私性的同时，为密文数据去重提供了保障，它使得相同的明文可以被加密成相同的密文，这给云服务器执行去重操作提供了可能。用户在上传文件前执行加密操作，所得密文数据如图 3.11 所示。

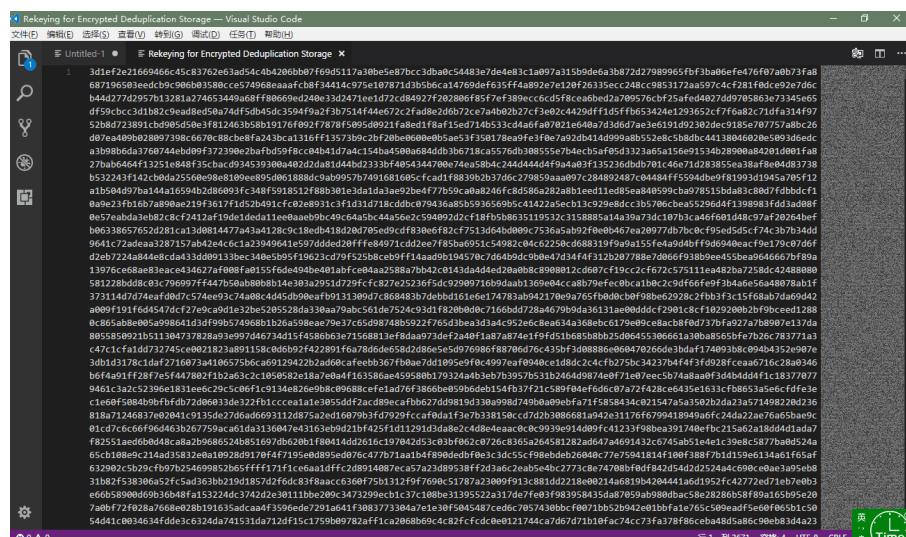


图 3.11 加密后的密文数据

3.2.3 上传功能

用户上传文件之前，必须先登录到可去重的数据完整性系统中。首先用户打开网页进入登录界面，如图 3.12 所示。



图 3.12 登录界面

经智能合约验证，输入正确的账户名和密码之后，用户进入到可去重的数据完整性审计系统中，并返回账户、账户余额、文件等当前用户的信息，用户主界面如图 3.13 所示。

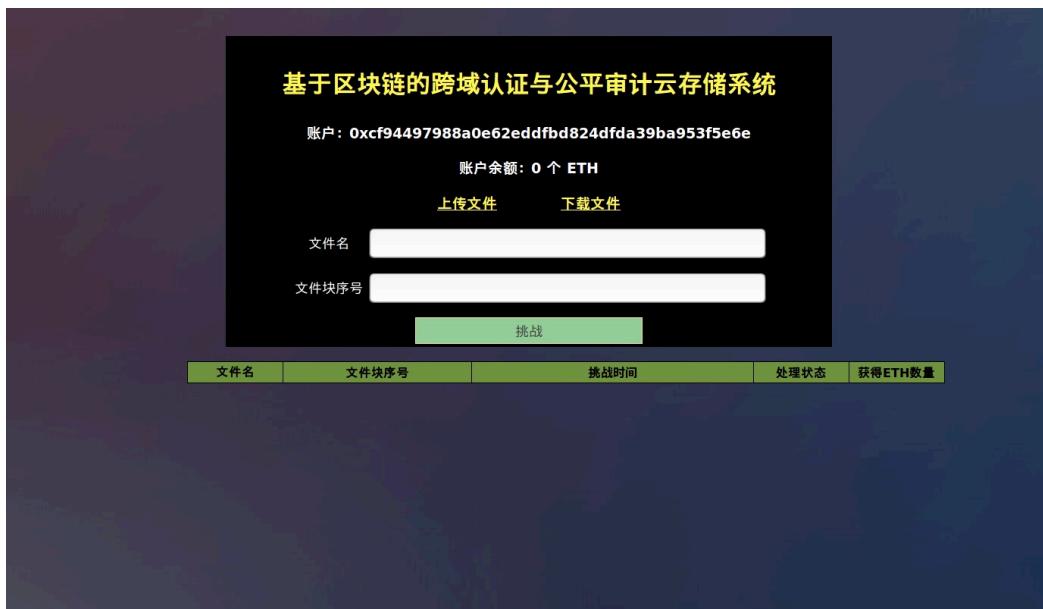


图 3.13 用户主界面

用户登录系统后，点击“上传文件”按钮将进入星际文件系统（IPFS）上传界面。此时用户可以从计算机上任意选取文件进行上传，上传的文件将被存储在 IPFS 中，上传界面如图 3.14 所示。

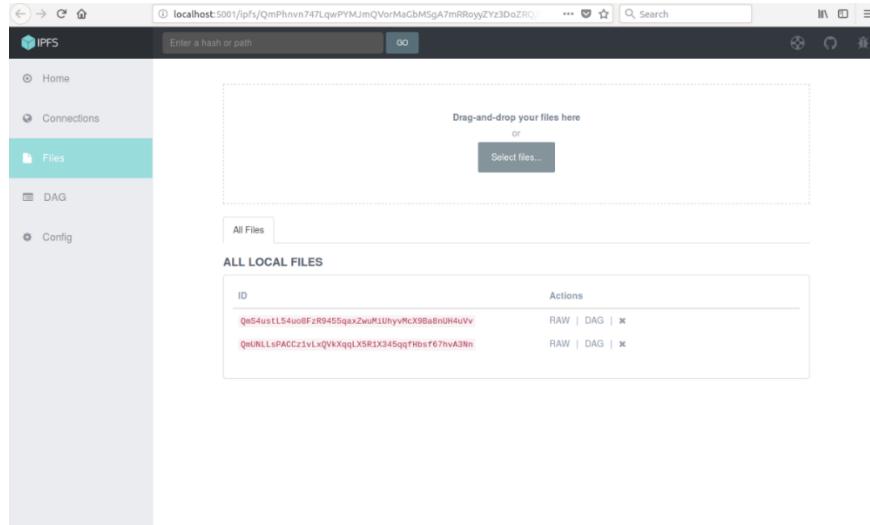


图 3.14 上传文件界面

3.2.4 去重功能

由于用户数据使用收敛加密算法进行加密，这使得相同的明文被加密成相同的密文。IPFS 将检测数据标签 tag 是否已经存储在 IPFS 服务器，如果 IPFS 服务器中已经存在相同的数据，则不再存储重复的数据。这将节约用户和云服务器大量的计算开销和存储开销。重复文件检测界面如图 3.15 所示。

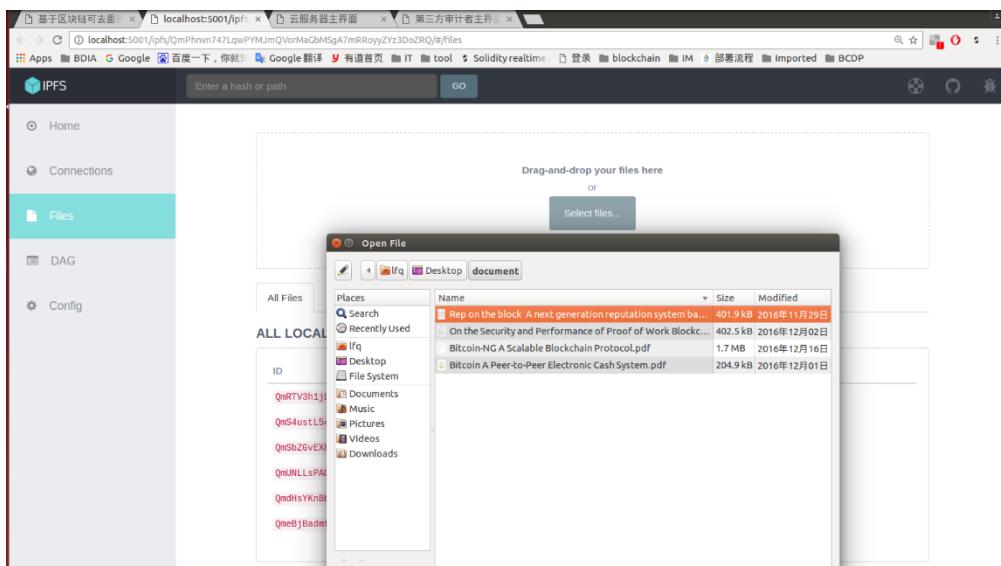


图 3.15 重复文件检测

3.2.5 下载功能

用户在主界面点击“下载文件”按钮，进入IPFS服务器，选择需要下载的数据就可以成功下载希望下载的数据。成功下载所得文件如图3.16所示。

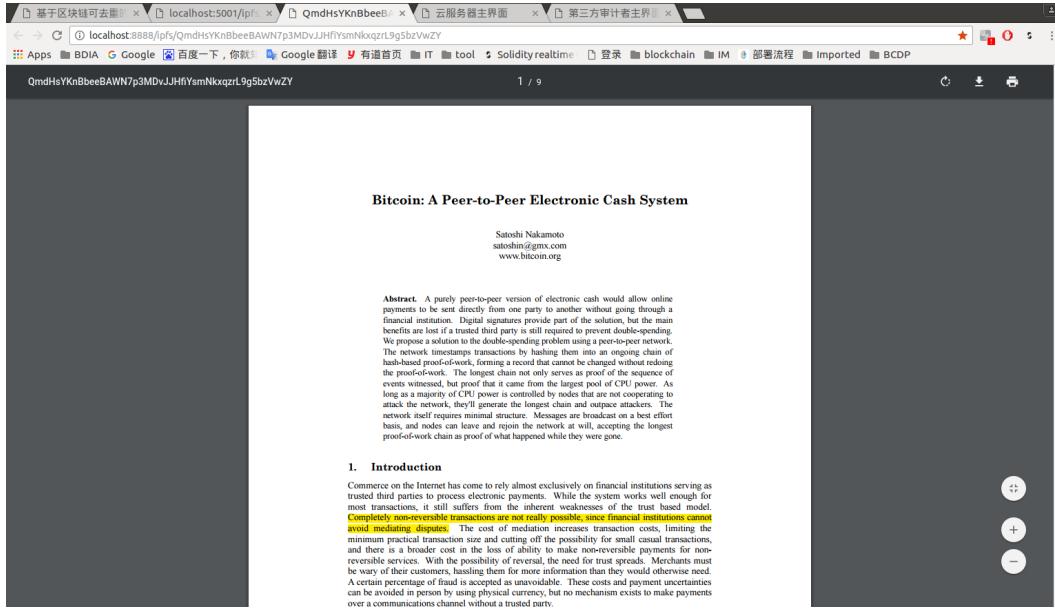


图3.16 下载的文件

3.2.6 完整性审计功能

如果用户希望对存储在云服务器上的数据进行完整性审计，首先需要输入文件名、选择文件块序号，向云服务器发起挑战，发起挑战界面如图3.17所示。



图 3.17 用户发起挑战

用户发起挑战后，将产生三种处理状态：未处理、验证成功和验证失败。挑战状况如图 3.18 所示。

文件名	文件块序号	挑战时间	处理状态	获得ETH数量
Bitcoin	3	2018-10-23 09:46:22	验证失败	1
Bitcoin	1	2018-10-23 09:35:50	验证成功	0
Bitcoin	2	2018-10-23 09:46:10	未处理	0

图 3.18 用户挑战状态

云服务器在接收到用户的挑战之后，需要对用户的挑战进行相应，并根据用户的挑战块的信息生成完整性审计的证据，提交给第三方审计者（TPA）。服务器生成证据界面如图 3.19 所示。



图 3.19 服务器生成证据界面

第三方审计者（TPA）在接收到云服务器产生的完整性证据之后，使用云服务器产生的证

据对存储在云服务器上的数据进行完整性审计工作，验证其用户数据的完整性。TPA 验证界面如图 3.20 所示。



图 3.20 TPA 验证界面

经 TPA 审计后，若完整性验证通过，则显示“验证成功”，如图 3.21 所示。

Bitcoin	1	2018-10-23 09:35:50	T 1922998455693012338298216379 p 2959322125190860000303666562	验证成功
---------	---	---------------------	--	------

图 3.21 TPA 验证成功

经 TPA 审计后，若用户数据的完整性验证不通过，则显示“验证失败”，如图 3.22 所示。

Bitcoin	3	2018-10-23 09:46:22	T 1922998455693012338298216379 p 2959322125190860000303666562	验证失败
---------	---	---------------------	--	------

图 3.22 TPA 验证失败

3.2.7 服务器惩罚功能

在云服务器提交完整性证据之后，TPA 执行完整性审计工作并将执行结果提交给智能合约。根据事先智能合约协定，当用户存储在云服务器上的数据的完整性被破坏时，扣除云服务器一定的金额。服务器被惩罚界面如图 3.23 所示，由于一个文件验证失败，账户余额由 100ETH 扣至 99ETH。

The screenshot shows a web-based challenge interface. At the top, it displays the account information: 账户: 0xcf94497988a0e62eddfbd824dfda39ba953f5e6e and 账户余额: 1 个 ETH. Below this, there are two buttons: 上传文件 (Upload File) and 下载文件 (Download File). A red input field labeled '文件名' (File Name) is present. Another red input field labeled '文件块序号' (File Block Sequence Number) is also present. A green button labeled '挑战' (Challenge) is at the bottom. Below the interface is a table showing the challenge history:

文件名	文件块序号	挑战时间	处理状态	获得ETH数量
Bitcoin	4	2018-10-23 09:48:46	未处理	0
Bitcoin	3	2018-10-23 09:46:22	验证失败	1

图 3.23 云服务器被惩罚

而被惩罚服务器的金额将补偿给利益被损害的用户。用户获得赔偿界面如图 3.24 用户获得补偿所示，由于一个文件完整性验证未通过，因此用户获得 1ETH 作为补偿。

The screenshot shows a web-based compensation interface. At the top, it displays the account information: 基于区块链的跨域认证与公平审计云存储系统, 账户: 0x450758353af24ffa469f5101e38327908208c35d, and 账户余额: 100 个 ETH. Below this, there is a '下载文件' (Download File) button. There are two input fields: 'T的值' (Value of T) and 'p的值' (Value of p). A search bar contains the placeholder '文件名或处理状态' (File Name or Processing Status) and a '查询挑战' (Query Challenge) button. Below the search bar is a table showing the compensation history:

文件名	文件块序号	挑战时间	完整性审计	证据
Bitcoin	1	2018-10-24 09:03:12	验证成功	已提交

图 3.24 用户获得补偿

3.3 系统性能测试

为了测试 2.3 节中前五个系统性能的技术指标。在认证部分，我们使用 Docker 容器技术模拟网络测试环境，使用 2500 个节点进行认证，每次增加 500 个节点，记录网络开销，并对比了传统 PKI, Kerberos（另一种主流跨域认证模型）。而在密文去重部分我们选取了 1MB、2MB、4MB、6MB、8MB 和 10MB 共 6 个文件分别对系统数据加密速度、服务器生成证据速度、完整性审计速度和数据解密速度五个项目进行测试，记录它们的运行时间，并与文件大小相除计算得到速度指标，最后计算各个指标的平均值。认证效率折线图如图 3.25 所示。

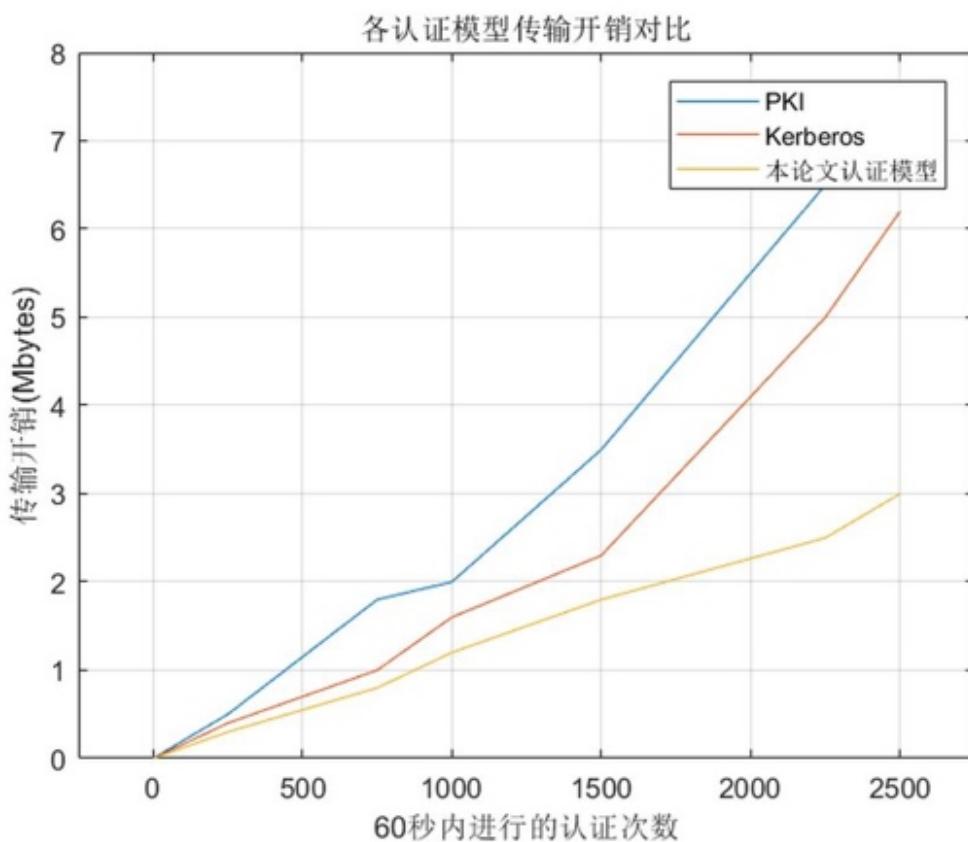


图 3.25 认证效率对比

可以看到，在我们的认证模型中，相同认证次数所产生的网络开销远远小于两种广泛使用的传统认证模型，降低了约 70%，对应的加解密签名验证计算开销也相应减少，性能表现较以往更加优越。

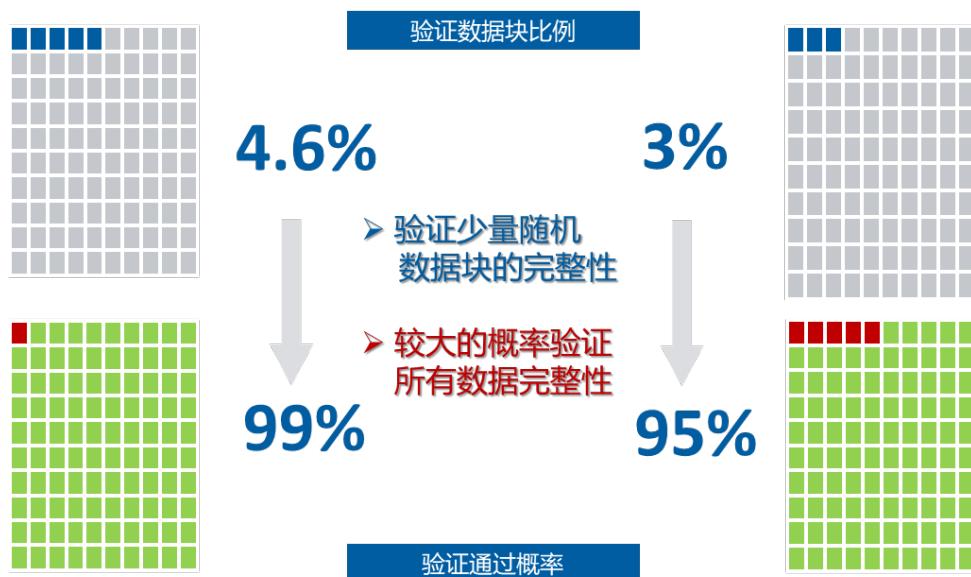


图 3.26 完整性审计验证性能

在公平完整性审计模块中，我们方案可以通过仅审计目标数据中不到 5% 的数据块，就能以 99% 的概率保证数据的完整性（删除 1% 的数据块），减少云服务商审计数据完整性过程中的运算开销，提高验证效率，更高频率地更新数据状态，方便用户及时了解。

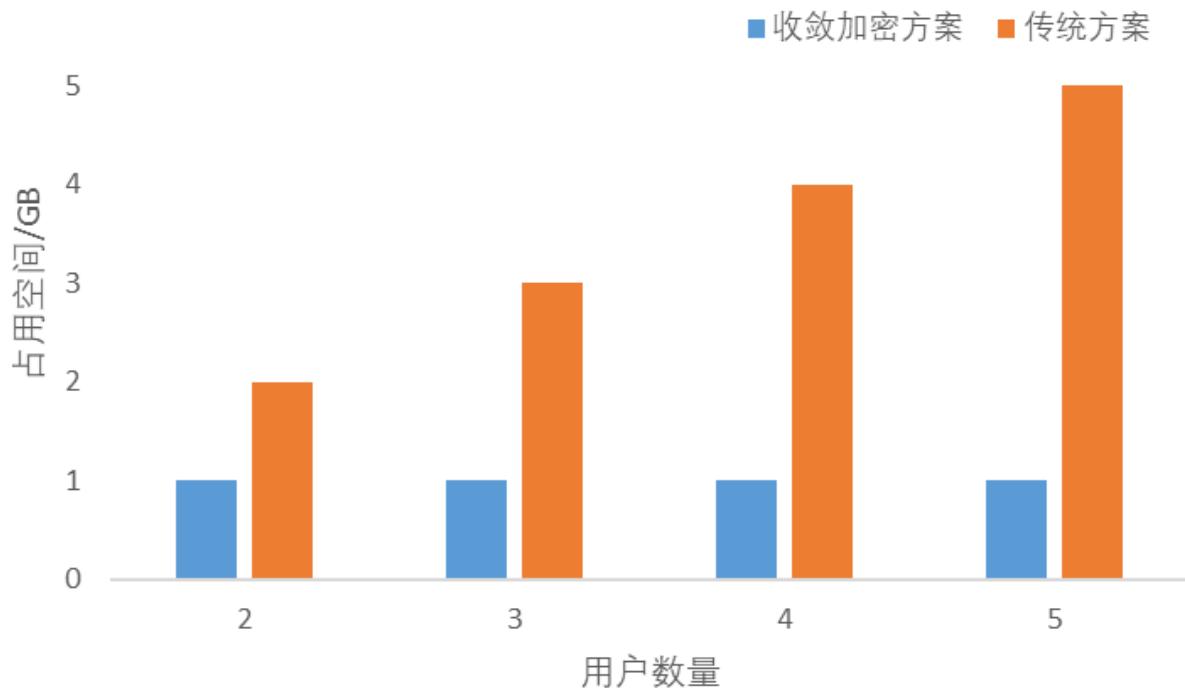


图 3.27 不同加密方案存储开销对比

在密文去重模块中， N 个用户对同一文件加密后的结果可以被云服务商识别，减少不必要的存储冗余，开销变为传统方案的 $1/N$ ，同时保护用户隐私。

第四章 应用前景与结论

在大数据时代，个人企业乃至国家机关提供应用服务主要基于云开发，云的安全性直接关系到建立于其之上的网络应用安全性。近些年来随着互联网的发展，云安全事件层出不穷，黑客攻击手段也在不断进化，尽管 Web 应用层面上的安全产品不断涌现，国内各大厂家如腾讯，阿里，360 结合人工智能技术，推出繁多的网络安全软硬解决方案，但各种云计算依托的基础设施（如认证，存储等）本身的脆弱性，受到关注的程度仍然十分有限。

云存储安全事件的频繁发生，暴露出了我国信息安全的基础设施仍不完善，中商产业研究院预计以云存储安全为核心的信息安全应用市场会成为行业的增长点，其成长速度将高于整体行业。当前我国的云存储安全市场规模在 20 亿元左右，中商产业研究院估计未来几年，我国云存储安全市场规模年增速在 30%以上，预计 2020 年市场规模将超过 70 亿元，前景广阔。

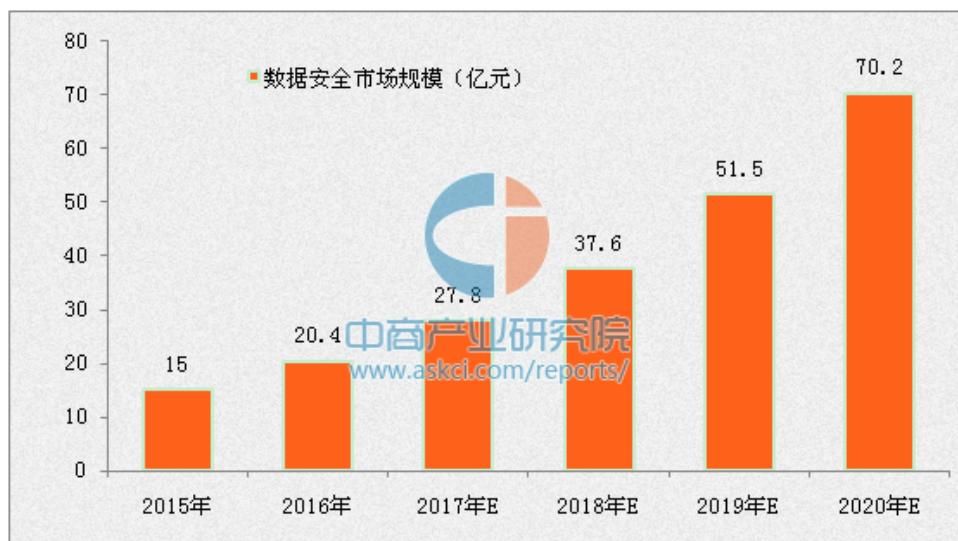


图 4.1 云存储市场增长图

我们的作品创新性地结合了区块链与多种密码学技术，从底层出发，为跨域认证，完整性审计，加密存储等云存储关键设施提供了性能更加优良，也更具安全性的全方位解决方案。可以为企业，政府机关云存储安全建设提供从上倒下整体技术支持。

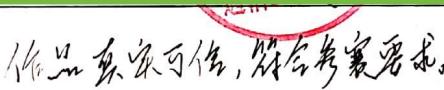
本方案安全可靠，技术壁垒较高，能从根本上保护云计算关键安全设施。我们相信，随着个人对数据隐私安全意识不断提高，国家对于网络安全战略地位的不断重视，我们的方案能最大程度上满足使用需求，并保持较大的发展潜力，拥有广阔的市场前景.

第五章 附录

5.1 陕西省区块链与安全计算重点实验室主任 ISN 成员 裴庆祺教授推荐信

D. 推荐者情况及对作品的说明

- 说明：1. 由推荐者本人填写；
 2. 推荐者必须具有高级专业技术职称，并是与申报作品相同或相关领域的专家学者或专业技术人员（教研组集体推荐亦可）；
 3. 推荐者填写此部分，即视为同意推荐；
 4. 推荐者所在单位签章仅被视为对推荐者身份的确认。

推荐者情况	姓名	性别	男	年龄	44	职称	教授
	工作单位						
	通讯地址						
	单位电话						
推荐者所在单位签章							
请对申报者申报情况的真实性作出阐述	作品真实可信，符合参赛要求。 						
请对作品的意义、技术水平、适用范围及推广前景作出您的评价	该作品思路新颖，将认证、数据完整性审计与区块链热门技术结合在一起，提出了相较于传统更加高效的解决方案。研究处于领先水平，是非常有益的尝试。安全是云计算发展的重中之重，但信息安全产品市场仍有很大空间，相信此作品在未来有极大的发展前景。						
其它说明							

5.2 中国航天二院 706 所研究员 密码学前沿国防项目带头人 徐志华推荐信

D. 推荐者情况及对作品的说明

- 说明：1. 由推荐者本人填写；
 2. 推荐者必须具有高级专业技术职称，并是与申报作品相同或相关领域的专家学者或专业技术人员（教研组集体推荐亦可）；
 3. 推荐者填写此部分，即视为同意推荐；
 4. 推荐者所在单位签章仅被视为对推荐者身份的确认。

推荐者情况	姓名	性别	男	年龄	39	职称	研究员
	工作单位						
	通讯地址						
	单位电话						
推荐者所在单位签章							
请对申报者申报情况的真实性作出阐述	作品由团队独立完成，功能完善，达到了参赛要求，情况属实。 						
请对作品的意义、技术水平、适用范围及推广前景作出您的评价	该作品将密码学、区块链技术有效结合，针对云计算场景中存在的身份认证、数据安全、数据隐私等关键问题，提出了具有创造性地解决方案，研究水平领先。相比传统集中式的防护手段，该作品首次利用了区块链去中心化的特点，为云计算提供了新的研究思路。习总书记强调，提高网络安全就建设国家总体安全观，网络安全和信息化是国家安全的主要领域和前沿领域，该作品必然会在日后带来较高的使用与研究价值。						
其它说明							

5.3 区块链头部公司（专利数国内 TOP3），杭州复杂美公司副总裁 曹竞推荐信

D. 推荐者情况及对作品的说明

说明：1. 由推荐者本人填写；

2. 推荐者必须具有高级专业技术职称，并是与申报作品相同或相关领域的专家学者或专业技术人员（教研组集体推荐亦可）；
3. 推荐者填写此部分，即视为同意推荐；
4. 推荐者所在单位签章仅被视为对推荐者身份的确认。

推荐者情况	姓名	性别	男	年龄	42	职称	讲师
	工作单位						
	通讯地址						
	单位电话						
推荐者所在单位签章							
请对申报者申报情况的真实性作出阐述	<p>作品由团队独立完成，创新新颖，功能完善，达到了参赛要求，情况属实。</p>						
请对作品的意义、技术水平、适用范围及推广前景作出您的评价	<p>基于区块链技术的分布式记账系统一直以来都以其去中心化去信任的优势在数据存证领域被讨论和实践，鼎安云的跨域认证和公平审计云存储系统也是一次很有意义的尝试。2018年杭州市和北京市互联网法院的区块链电子证据公证模式被誉为“中国司法区块链第一案”，就是很好的示范。鼎安云系统将区块链技术与CA技术结合，案例就是很好的示范。鼎安云系统将区块链技术与CA技术结合，将跨域电子证据的存储和使用同公钥信用结合，安全性得到了一个新的高度，而在公平审计的云存储架构设计上提出了独到的见解，有重要的意义。</p>						
其它说明							

5.4 项目团队成员曾在知名企安全部门工作



字节跳动

实习生录用通知函

李艺扬同学，你好！

我们非常高兴地通知你，你已经通过笔试/面试考核，公司拟录用你并拟与你签订实习合同。



聘用函

致：谢意先生/女士

非常感谢您参加360公司实习生招聘，欢迎您加入360！在您加入之前，以下内容需要您与公司达成共识，并共同确认：

您实习的职位情况：部门：安全研究院·无线电安全研究部·安全研究组.，职位：安全研究实习生，工作地点：北京

其中，华为云安全部门对我们的方案高度评价，并达成初步合作意向。

5.5 项目成员的国家机构云服务安全漏洞证明

漏洞编号

vulbox-2017-0103135

<http://www.shaanxirk.com/Scores.asp?pt=0>存在注入漏洞 已通报国家机构

漏洞编号

vulbox-2018-0138683



CNNVD 确认了您的漏洞 4天前

已确认此漏洞，开始漏洞修复工作，漏洞名称：西北电力建设第四工程公司存在xss (vulbox-2019-0169504)

[查看漏洞](#)

漏洞编号

vulbox-2019-0169504

[shaavipolyclinic](#)—国外医务网站存在存储型XSS 已通报国家机构

5.6 测试数据与结果

根据认证功能测试方案，测试得到认证次数与网络开销的原始数据表 5.1 认证开销数据表。

表 5.1 认证开销数据表

认证次数/开销	本文模型/Mb	PKI/Mb	Kerberos/Mb
0	0	0	0
250	0.39	0.51	0.41
750	0.83	1.82	1.07
1000	1.24	2.06	1.63
1500	1.84	3.52	2.36
2250	2.55	6.57	5.16
2500	3.01	7.52	6.29

根据测试四个系统性能的测试方案，测试得到数据加密时间、生成证据时间、审计时间、数据解密时间的原始数据见表 5.2。

表 5.2 运行时间测试数据表

文件大小/MB	数据加密时间/s	生成证据时间/s	审计时间/s	数据解密时间/s
1	0.0377	4.6378	0.00106	0.0399
2	0.0571	9.1624	0.00107	0.0585
4	0.0956	19.145	0.00106	0.1012
6	0.1289	28.386	0.00107	0.1293
8	0.1612	37.843	0.00107	0.1675
10	0.1981	47.293	0.00107	0.2012

计算得到速度指标数据见表 5.3。

表 5.3 速度计算数据表

文件大小 /MB	数据加密速度 /Mbps	生成证据速度 /Mbps	审计速度 /Mbps	数据解密速度 /Mbps
1	26.525	0.216	943.396	25.063
2	35.026	0.218	1869.159	34.188
4	41.841	0.209	3773.585	39.526
6	46.548	0.211	5607.477	46.404
8	49.628	0.211	7476.636	47.761
10	50.480	0.211	9345.794	49.702

根据以上结果，得到四个项目的平均速度指标见表 5.4。

表 5.4 各项平均速度表

文件大小 /MB	数据加密速度 /Mbps	生成证据速度 /Mbps	审计速度/s	数据解密速度 /Mbps
平均值	41.675	0.212	0.00107	40.441

由于用户将大量的数据外包存储在云服务器上，服务器出于自身利益的考虑，可能删除用户很少访问的数据块。用户为了保证所有数据块的完整性，如果在审计过程中对所有的数据块进行完整性审计将消耗大量的计算资源。因此，我们使用概率性完整性审计方案，用户在每次发起挑战时随机选择需要挑战的数据块，通过验证少量的随机数据块的完整性以较大的概率验证服务器是否完整的存储了所有的用户数据。我们需要验证随机块的个数和验证概率的关系见表 5.5。

表 5.5 验证概率数据

验证概率	数据块总个数	验证数据块的个数
99%	10000	460
95%	10000	300
99%	8000	480
95%	8000	320
99%	6000	456
95%	6000	360

通过使用可证明数据拥有技术，可以在验证少量随机数据块的同时以较高的概率保证所有数据块的完整性。当数据总量 1%、5%的数据被恶意删除时，需要验证数据块的个数和验证通过的概率的关系分别如图 5.1、图 5.2 所示。

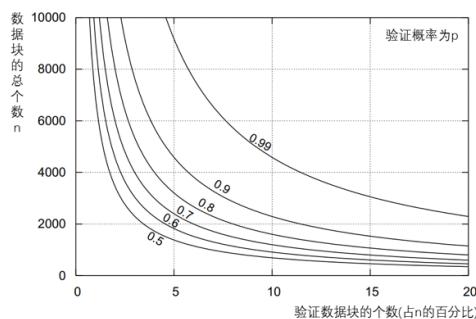


图 5.1 删数据总量 1% 时验证数据块个数与验证通过概率曲线图

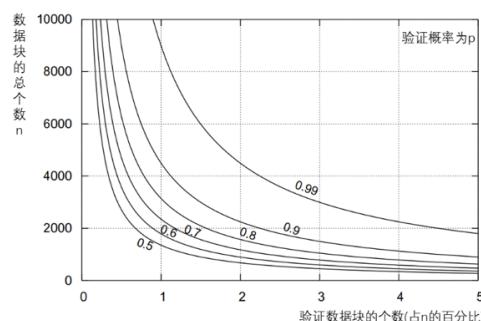


图 5.2 删数据总量 5% 时验证数据块个数与验证通过概率曲线图