基于 TPA 云联盟的数据完整性验证模型

田俊峰[1,2]，李天乐[1,2]

（1. 河北大学网络空间安全与计算机学院，河北 保定 071002；
2. 河北省高可信信息系统重点实验室，河北 保定 071002）

摘 要：针对公有性验证模型中第三方审计机构（TPA, third-party auditor）不可信问题，提出基于 TPA 云联盟的数据完整性验证模型。首先，设计 TPA 云联盟的体系结构并定义系统平台的主要功能组件及作用，联盟可以对 TPA 云成员进行管理和控制。其次，利用可信计算技术和区块链技术对 TPA 进行详细的设计，确保 TPA 执行环境和工作流程的可信性。最后，利用 TPA 云联盟构建数据完整性验证模型，并对模型的正确性、安全性和有效性进行理论和实验分析。

关键词：数据完整性验证；区块链；可信计算；第三方审计机构
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2018144

Data integrity verification based on model cloud federation of TPA

TIAN Junfeng[1,2], LI Tianle[1,2]

1. Institute of Network Technology, Hebei University, Baoding 071002, China
2. Key Lab on High Trusted Information System in Hebei Province, Baoding 071002, China

Abstract: Aiming at the untrustworthiness of third-party auditor (TPA) in the publicity verification model, a data integrity verification model based on the cloud federation of TPA was proposed. Firstly, the cloud federation of TPA's architecture was designed and the main functional components and function of the system platform was defined. The federation could manage and control the TPA cloud members. Secondly, TPA was designed in detail by using trusted computing technology and blockchain technology to ensure the credibility of the TPA execution environment and workflow. Finally, the data integrity verification model was built by using cloud federation of TPA. The correctness, security and effectiveness of the model were analyzed theoretically and experimentally.

Key words: data integrity verification, blockchain, trusted computing, third-party auditor

1 引言

随着移动互联网、物联网、大数据时代的到来，数据量呈指数级增长趋势，个人的存储能力已无法满足现有的存储需求。云存储（cloud storage）[1]是基于云计算衍生出来的概念，即为满足云计算系统海量的数据存储空间而产生的。其通过集群应用、网络技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集成起来协同工作，共同对外提供数据存储和业务访问功能。通过云存储系统，用户可以采用按需付费的方式，利用较小的花费即可在云端得到更强大的存储能力。然而用户将数据存储在云端，并将本地数据删除，失去了对数据的实质性控制。云端是否完整地持有用户的数据、怎样证明云端持有用户数据的正确性等问题引起了学术界的广泛关注。

收稿日期：2017−06−11；**修回日期**：2018−06−14
通信作者：李天乐，csscholar2017@163.com
基金项目：国家自然科学基金资助项目（No.61170254，No.60873203）；河北省自然科学基金资助项目（No.F2016201244）
Foundation Items: The National Natural Science Foundation of China (No.61170254，No.60873203), The Natural Science Foundation of Hebei Province (No.F2016201244)

不可信云存储服务商（CSP, cloud storage provider）是否完整地持有用户的数据，是近年来在云存储安全领域内备受关注的问题。目前，对于验证云存储服务商或入侵者有没有对用户存储的数据进行恶意的删除、修改等行为，研究者们提出 2 类校验模型：数据持有性证明（PDP, provable data possession）和数据可恢复性证明（POR, proof of retrievability）。前者可以验证数据是否被云端正确地持有；后者可以对丢失、损坏的数据进行一定的恢复。此外，参照验证者身份，验证方案可以分为 2 种：私有验证方案和公开验证方案。相较于私有验证，设有第三方审计机构 TPA 的公开验证更好地支持了公开审计、动态更新、验证高效等审计特点。

Ateniese 等[2]首先提出了 PDP 的概念，利用基于 RSA 的验证方法和同态验证标签（HVT, homomorphic verifiable tag）来进行文件块的完整性验证工作，并利用概率性验证方案，有效地减少了计算代价和通信开销。Wang 等[3]采用在同等安全条件下比 RSA 和 DSA 更短的 BLS 签名技术构造数据标签进行验证。由于 BLS 签名机制的同态特性，可将多个签名聚集为一个签名进行统一验证，使基于 BLS 签名的验证方案存储代价和通信开销大大减少。此外，基于 BLS 签名的 PDP 机制是一种公开验证机制，用户可以将烦琐的审计任务交给 TPA 来完成，减轻了用户的计算开销，提高了整个方案的效率。在公开审计过程中，第三方可能利用多次审计中的线性组合方程式进行高斯消元，从而窃取用户的数据。针对审计过程中可能泄露用户数据隐私的问题，Wang 等[4]采用随机掩码技术，利用随机置换函数保证用户的数据不会泄露给 TPA 或云存储提供商。针对动态更新验证问题，Erway 等[5]首先考虑引入动态数据结构来支持全动态操作，提出基于跳表的 PDP 机制来验证动态操作。利用认证跳跃表的数据结构，根据云存储服务器返回的认证路径、标签信息和本地存储的认证元数据判断数据块在位置上是否正确。Wang 等[6]提出基于 Merkle 树的数据结构来验证动态更新操作。与 Erway 等[5]不同的是，数据标签并没有参与动态结构中根节点散列值的计算，而是利用构造的 Merkle 认证散列树中的根散列值和某个叶子节点的辅助认证信息来保证云存储提供商的动态操作更新的正确性。此外，Wang 等[4]提出数据验证文件批审计（batch auditing）的概念。TPA 如果同时处理多个审计任务请求时，单个处理起来会非常烦琐和浪费资源。如果将不同的审计请求的线性签名聚合成一个签名，使 TPA 进行一次审计，则大大提高了审计效率。

以上的公开审计验证模型中，大多数都是在假设 TPA 是可信的前提下来完成整个数据完整性验证工作的。然而在实际应用中，TPA 内部的工作流程及其可信性都需要进一步的研究和证明。虽然可以利用加密数据块以及随机掩码等技术防止原始数据在 TPA 中泄露。然而，对于用户来说，TPA 是一个黑盒子，其内部构造和运作流程都不得而知。在现实中，TPA 有可能与 CSP 串谋对用户进行攻击，即无论 CSP 中用户数据是否完整，返回给用户的完整性验证结果都是验证成功的。此外，TPA 还有可能与用户进行串谋（或被其他 CSP 竞争者贿赂），故意验证失败 CSP 生成的证明，这时返回用户的完整性验证结果都是验证失败。

Xu 等[7]利用加密的审计方案和复审方案来保证 TPA 在规定时间内完成数据完整性验证。在此基础上，Huang 等[8]采用多个 TPA 进行审计的授权，如果一个 TPA 不能被信任，它将会被撤销并被其他 TPA 替换。然而文献[7-8]的审计方案都定义了“time sever”和“receive sever”，而这 2 个实体的可信性和安全性有待进一步证明。Wu 等[9]提出把 TPA 的验证工作分为复杂的计算过程和简单的验证过程，前者由第三方审计机构处理，后者由用户自己验证。肖达等[10]利用部署在云存储服务器端的可信硬件来进行审计日志的生成与存储，由于可信硬件的性能有限以及部署在不可信云存储提供商等风险，其安全性和可信性也有待进一步验证。以上方案并没有真正解决使 TPA 诚实可信地完成云用户的审计任务。

针对如何构建可信 TPA 以及利用可信 TPA 诚实可信地完成用户的审计任务，本文做了如下工作。

1) 设计 TPA 云联盟的体系架构，利用 TPA 云联盟管理平台对 TPA 的运行进行管理和控制。

2) 设计 TPA 的逻辑框架，使其进行审计任务的工作流程透明化、可视化。

3) 构造关于 TPA 可信度量值以及审计任务参数的区块链，使 TPA 的状态和行为可追溯。

4) 完成基于 TPA 云联盟的数据完整性验证模型，在保证可信 TPA 行为可信的同时，验证存储在不可信云存储商的用户数据的完整性。

2　TPA 云联盟的设计方案

本节定义了 TPA 云联盟的体系架构和功能模块，利用联盟对 TPA 进行管理和控制，利用可信计算技术和区块链技术对 TPA 进行设计，包括 TPA 操作层的关键功能模块以及 TPA 区块链层可信区块链的形成。

2.1　TPA 云联盟

TPA 云联盟（CFTPA，cloud federation of TPA），是参考欧洲 SUNFISH 项目中云联盟（FaaS, federation as a service）[11]与基于 TPM 联盟的可信云平台管理模型[12]，为可信 TPA 而设计的一种云联盟模型。TPA 云联盟保证了 TPA 可信的运行环境和诚实的验证流程。此外，TPA 云联盟可以根据业务与隐私保护等级的不同，将政务、金融、医疗等机构的验证任务分配给符合要求的 TPA 云成员来进行处理。云联盟中，分布式云成员之间通过 P2P 方式进行通信和交互，并通过 TPA 云联盟管理平台进行统一的管理与调度。

TPA 云联盟结构如图 1 所示。虚线内的 TPA 表示多个 TPA 构成的 TPA 云联盟，每个 TPA 由 TPM 和 vTPM 构成，可信计算基，为可信度量提供存储、验证、计算等功能。TPM 与 vTPM 的组合体系结构采用 Berger 等[13]提出的框架模型，vTPM 的采用大大扩展了 TPM 的应用场景，突破了 TPM 的性能瓶颈。TPA 在 vTPM 创建的虚拟机中执行相应的审计任务。TPA 逻辑结构分为操作层和区块链层。操作层进行数据完整性验证工作流程，区块链层收到操作层的日志记录，形成关于可信度量值与审计日志记录的不可更改区块链。

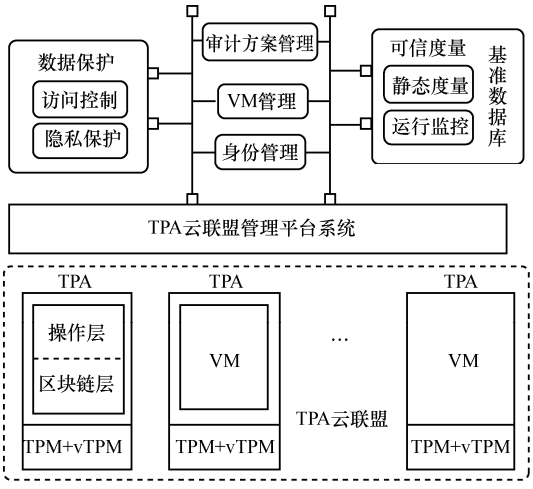


图 1　TPA 云联盟结构

TPA 云联盟管理平台系统对 TPA 进行综合的管理与调度，包括可信执行环境度量、组件间的认证、审计方案的分配、TPA 云资源的使用与注销等。其主要功能组件有如下定义。

1) 身份管理组件：包括一系列认证功能的集合，认证管理 TPA 云成员、云用户（CU）和联盟管理员的身份注册、注销等功能，以及认证和管理平台系统中各个组件的身份和正常功能；认证管理 TPA 云联盟与管理平台的系统交互；认证云用户任务请求并通过 VM 管理开启相应云资源；认证云用户与 TPA 通过数据保护组件进行交互；认证平台组件通过加密令牌与区块链的交互。总之，身份管理组件是平台系统中必不可少的组件，是所有任务流和数据流的认证校验基础。

2) VM 管理组件：即虚拟机云资源的管理组件。在身份认证通过的情况下，可以根据业务需求和节点状态对 TPA 云联盟中的云资源进行开启、关闭和挂起等命令。联盟管理员一旦发现 TPA 行为异常，可对该 TPA 进行撤销命令，并扣除相应积分。特别地，一旦有云用户向 TPA 云联盟发起审计任务，VM 管理组件会根据参数信息（如花费或 SLA 协议[14]）在 TPA 云联盟中匹配到合适的 TPA 资源并提供给云用户，还能提供其他公有云、私有云的交互 API。

3) 审计方案管理：可以存储高效优秀的审计方案，并支持用户自定义审计方案（需要通过安全审核）；提供对审计方案的加密功能[7-8]。此外，如果对安全隐私要求较强的用户，可采用强加密方案[15]。

4) 数据保护组件：数据保护组件旨在保证云用户数据的安全。包括用户要存储在 TPA 的元数据以及对 CSP 返回的证据进行强制随机掩码处理[4]。并采用基于属性的访问控制（AAC, attribute-based access control）[16]协议，保证访问控制协议与所使用的云资源服务相匹配，进而保证组件间的隐私保护。

5) 可信度量组件：负责管理监控各个 TPA 中 TPM 可信证据收集与对比校验工作。利用收集到的可信证据与基准数据库进行对比校验，实现对 TPA 启动时的静态完整性度量（包括 BIOS、bootloader、OS、配置文件）和系统运行时的动态完整性度量（包括虚拟机、进程的可执行文件、环境变量）。

2.2　TPA 操作层的设计

如图 1 的 TPA 逻辑框架所示，TPA 的逻辑框架分为操作层和区块链层。TPA 的操作层逻辑框架如图 2 所示。其中，空心粗箭头表示数据的传输，实

心细箭头表示指令的发送。接下来，将详细介绍各个模块的功能及其在业务中的作用。

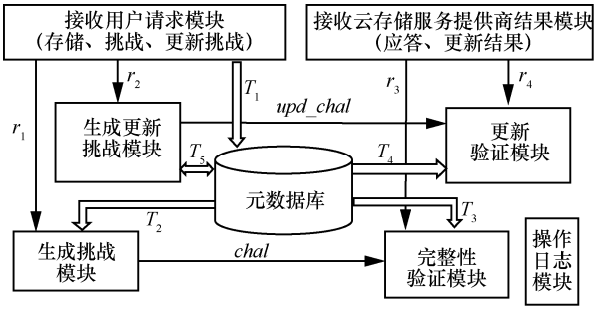


图 2 TPA 的操作层逻辑框架

1) 接收用户请求模块（AURM, accept CU request module）：分析用户发送给 TPA 的请求。分为 3 种请求类型：第一种为存储元数据请求，T_1 即用户发送经过预处理过的原始数据到 TPA 的元数据库；第二种为挑战请求，r_1 即用户要求 TPA 生成正确格式的挑战发送给 CSP；第三种为更新请求，r_2 即用户传达给 TPA 的更新要求。

2) 接收云存储服务提供商结果模块（APRM, accept CSP result module）：接收并处理云存储服务提供商的结果。第一种为 CSP 返回给 TPA 的“应答”结果，由 r_3 发送给完整性验证模块；第二种为 CSP 返回给 TPA 的“更新”结果，由 r_4 发送给更新验证模块。

3) 生成挑战模块（GCM, generate challenge module）：生成用户进行完整性验证请求的“挑战”命令，并发送给 CSP。根据收到的挑战请求 r_1，向元数据库发送相关元数据 T_2 生成“挑战”，最终发送给 CSP。

4) 生成更新挑战模块（GUCM, generate update challeng module）：对用户的更新请求生成更新挑战操作。收到更新挑战请求 r_2 后，该模块输入 T_5 和相应更新数据的位置信息等，生成 *upd_chal* 更新挑战，并发送给 CSP。同时，通过 T_5 保证存储在元数据库与存储在 CSP 中的数据的一致性。

5) 更新验证模块（UVM, update verification module）：对 CSP 发送来的更新结果进行更新验证。输入更新结果 r_4、更新请求 *upd*、更新后的元数据 T_4，进行三方的更新校验对比。确认 CSP 的“更新操作”是否正确，将输出对比结果发送给用户。

6) 完整性验证模块（IVM, integrity verification module）：是 TPA 的关键模块，其功能是完成对数据的完整性验证。首先，收到处理后的“应答”结果并向元数据库调取标签信息 T_3；然后，根据挑战 *chal*、应答 r_3、元数据 T_3，进行数据的完整性验证，并将验证结果发送给用户。

7) 元数据库（MDB, metadata database）：是存储用户经过特殊处理之后的元数据的一种特殊数据库，以供生成挑战模块、生成更新模块、更新校验模块、完整性验证模块调取所需数据。

8) 操作日志模块（OLM，operation log module）：在规定时间间隔内对 TPA 操作层的模块间和模块内的操作进行记录，并将数据记录进行封装，然后发送到区块链网络中。

2.3 TPA 区块链层的设计

TPA 内的区块链层设置在 TPA 操作层的逻辑底层。在区块链层内，有多个基于可信 TPM 的区块链节点。网络节点在区块链层内进行关于审计消息的广播和计算，并记录和存储审计状态和结果。所有 TPA 云联盟成员的区块链层节点构成区块链网络。可信硬件平台保障区块层的节点完整性，区块链网络中的共识算法保障网络节点的一致性。

图 3 所示为区块链层以及区块链形成的示意。每个 TPA 的区块链层内的节点如图 3 虚线椭圆部分所示，图 3 中有 2 种类型的节点：领导节点和普通节点。领导节点在全网络中负责把 TPA 的度量值和某时间段的操作记录打包进区块链中；普通节点负责收集 TPA 的可信度量值以及审计任务相关参数。2 种节点都会在区块链网络中进行监听、接收、计算验证、广播信息等工作。2 种节点的设置可以减轻单类型节点工作任务。

图 3 虚线矩形部分即为区块链示意。区块由区块头和区块体构成。区块头中“前区块的散列值”是指利用 SHA256 对前区块进行计算，然后保存到当前区块中，其可以保证区块链的链接和数据的不可更改；领导信息指形成此区块的领导信息，即将此轮选举出的节点信息保存在区块中，可供回溯查询相关信息；时间戳确定了区块的写入时间，同时使区块链具有时序的性质，且时间戳可以作为区块数据的存在性证明，有助于形成不可篡改、不可伪造的分布式账本；Merkle 根表示 TPA 中可信证据收集到的度量值进行两两散列运算的最终所得结果。区块体中除了包括在 TPA 中收集到的各组件、系统、软件运行状态的度量值外，还包括 TPA 审计任务的相关信息，例如，审计任务 ID、云用户的 ID、

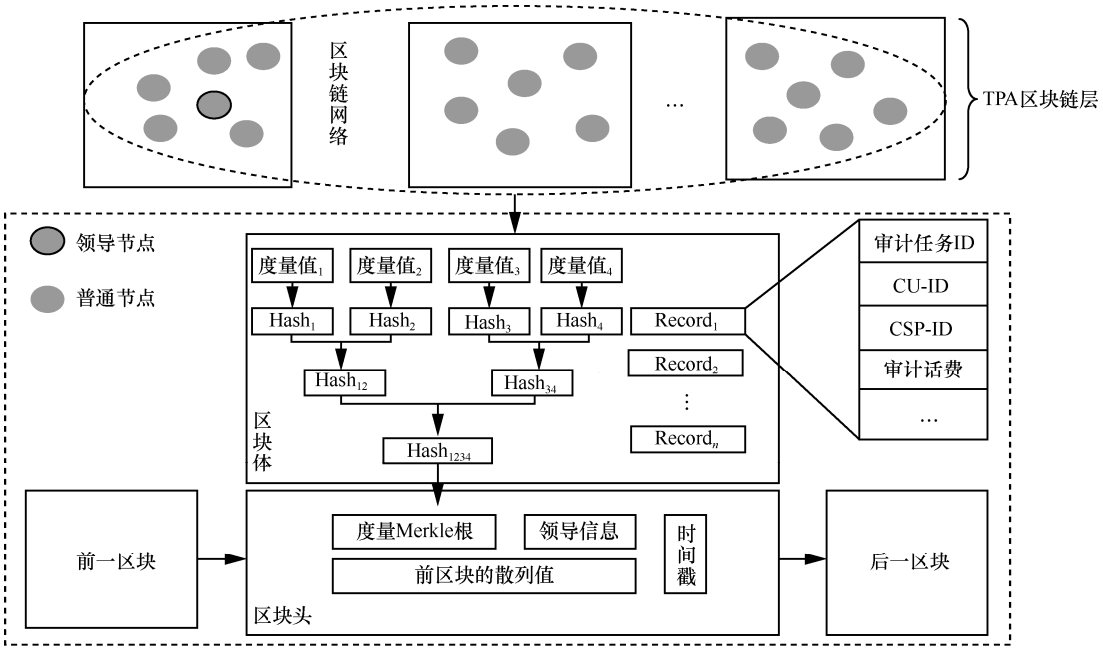


图 3　TPA 区块链层和区块链形成示意

云存储服务商的 ID 以及此次审计任务的具体方案和资源花费、时间花费等。

基于 PBFT 算法[17-18]以及 POS 权益证明方法[19]，并结合本文设计的 TPA 以及数据完整性验证的特点，提出 TPA 领导选举共识算法（TPA-LECA，TPA-leader election consensus algorithm），具体步骤如下。

1) 操作记录模块收集工作层的可信证据，在区块链网络中持续广播数据验证信息（K_{CU}, K_{CSP}, K_{TPA}, T_{int}, OPR）。其中，K_{CU} 表示云存储用户的一个度量值，由用户所属行业数据的机密性和重要性以及在 TPA 云联盟平台的注册时长和良好的用户行为决定；K_{CSP} 表示云存储提供商的一个度量值，由 CSP 返回“应答”效率以及数据完整性验证成功情况确定；K_{TPA} 表示 TPA 云联盟中的 TPA 成员的度量值，由 TPA 的工作效率、完整性度量结果和审计任务参数决定；T_{int} 表示记录此操作的时间间隔；OPR 表示在此时间间隔内 TPA 操作层进行的操作。

2) 所有节点均独立监听区块链网络中的广播信息并记录。

3) 经过时间间隔 t 后，各节点均把自己监听到的广播信息以及自己的签名（K_{CU}, K_{CSP}, K_{TPA}, T_{int}, OPR, W）发送到区块链网络，其中，W 表示权值，为各特征项与权重的乘积之和。

4) 各节点根据表 1 自动计算各节点的权值 W，并将权值最大的节点信息向区块链网络中广播。

表 1　　特征项权值参考

特征项	权重	参考值
K_{CU}	W_1	0.2
K_{CSP}	W_2	0.3
K_{TPA}	W_3	0.4
T_{int}	W_4	0.1

5) 任意节点收到超过 n 个（n 与节点总数有关，根据系统的容错能力而定）相同回应信息后，则达成共识。此相同信息中的签名者即本轮共识的领导，它负责将可信度量值和操作记录等信息打包进新的区块；作为领导节点的奖励，包含领导节点的 TPA 节点池会提高相应单个 TPA 的度量值 K_{TPA}。

6) 新区块完成后，各节点将之前的信息删除，并开始下一轮的共识。

区块链网络中的各个节点监听全网操作记录以及在接收到操作记录的信息后，需要对操作记录进行合法性的验证。如果发现非法的操作记录，则不写入其记账本；如果非法操作记录包含在领导节点打包的区块中，则放弃本轮共识结果，重新进行下一轮的共识过程。操作记录的验证流程如下所示。

1) 操作记录的数据格式是否符合系统规则，如果不符合则判定为非法。

2) 操作记录的行为是否符合系统规则，如果不符合则判定为非法。

3) 操作记录在区块链中是否已经存在，如果存

在则判定为非法。

4) 如果以上判定都不符合，则操作记录被认为是合法的。

3 基于 TPA 云联盟的数据完整性验证过程

基于 TPA 云联盟的数据完整性验证（CFTPA-DIV, CFTPA based data integrity verification）模型的参与实体有如下 3 种。

1) 云用户，也就是拥有原始数据的云存储服务使用者，将原始数据存储在云端服务器。

2) TPA 云联盟，代替用户向云存储提供商进行挑战，并验证挑战证据。

3) 云存储提供商，其存储着云用户的数据，且有可能无法保证用户数据的完整性，所以 TPA 云联盟要对其进行完整性“挑战”。

基于 TPA 云联盟的数据完整性验证模型的过程示意如图 4 所示。有别于一般公开审计的数据完整性验证，图 4 的 TPA 云联盟执行 TPA 的工作流程，其与用户和 CSP 之间构成的完整性验证过程如下所示。

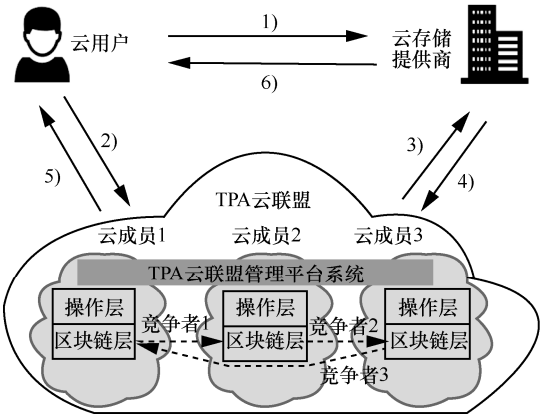


图 4 基于可信区块链的数据完整性模型

1) 用户将加密处理后的文件 F 和数据标签存储到 CSP，或用户在云端对数据的操作。

2) 将处理后的认证元数据发送到 TPA 云联盟，或是用户需要进行挑战命令（包括挑战完整性和挑战更新）。

3) 将 TPA 云联盟生成的完整性挑战或更新挑战发送给 CSP。

4) CSP 将完整性应答或更新的应答发送给 TPA 云联盟。

5) TPA 云联盟将“挑战—应答”的对比结果发送给用户。

6) CSP 提供给用户数据操作的反馈。

3.1 完整性验证方案

令 $G_1 = G_2 = G$，并且 $e: G \times G \to G_T$ 为一个双线性映射，其中，G 和 G_T 为质数 p 的乘法循环群，g 为 G 的一个生成元。令 $H: \{0,1\}^* \to G$ 表示一个映射点散列函数，它将字符串映射到 G，并且令 $h(\cdot): G \to Z_p$ 为另一个散列函数，表示将 G 的元素组一致地映射到 Z_p。

为了生成随机挑战索引 s_j 和相应的系数 v_{s_j}，定义 $\pi_{key}: \{0,1\}^{\mathrm{lb}(n)} \times K \to \{0,1\}^{\mathrm{lb}(n)}$ 的伪随机置换和伪随机函数 $f_{key}\{0,1\}^* \times K \to Z_p$，其中，$key$ 属于密钥空间 K。

CFTPA-DIV 完整性验证过程如下所示。

1) 初始化阶段（setup phase）

步骤 1 对 TPA 云联盟中的每个 TPA 底层 TPM 模块进行初始化设置，即进行静态的完整性度量。如果出现度量值与默认值不匹配的情况，则发出警报给 VM 管理，后者撤销此 TPA 参与此轮审计任务的分配。

步骤 2 身份管理组件对 TPA 云联盟中其他组件进行完整性度量以及正常功能检测。

步骤 3 云用户对 TPA 云联盟提出审计任务，即“生成挑战请求”（前提是已经在 TPA 云联盟管理平台进行注册）。

步骤 4 云用户在客户端运行 $KeyGen(1^k)$，生成一对随机签名密钥对 (ssk, spk)。然后选择 $x \leftarrow Z_p$ 和 $u \leftarrow G$，并计算 $v \leftarrow g^x \in G$。即 $sk = (x, ssk)$ 为用户的私钥，而 $pk = (u, v, g, spk)$ 为公钥。

步骤 5 为了给文件命名，用户为文件 $F = \{m_i\}_{1 \leqslant i \leqslant n}$ 在 Z_p 上随机选择一个元素 id，并计算其文件标签 $T = id \,\|\, Sig_{ssk}(id)$，然后对于每个文件块 $m_i \in Z_p$，用户生成一个签名为

$$\sigma_i = (H(i)u^{m_i})^x \in G, 1 \leqslant i \leqslant n \tag{1}$$

最后，用户将 $\{F, \phi = \{\sigma_i\}_{1 \leqslant i \leqslant n}, T\}$ 发送给 CSP，再发送 $\{\phi = \{\sigma_i\}_{1 \leqslant i \leqslant n}, T\}$ 给 TPA 云联盟作为认证元数据。

2) 挑战阶段（challenge phase）

步骤 1 TPA 云联盟收到云用户的审计请求后，根据请求数据保护组件执行相应的政策，审计方案管理组件选择合适的审计方案。

步骤 2　VM 管理组件，综合云用户的审计请求、数据保护政策以及所选择的审计方案和其他参数（包括云用户身份等级、TPA 在系统中的积分值）作为参考，将审计任务分配给合适的 TPA。

步骤 3　TPA 云联盟将任务分配到特定的 TPA 之后，首先在 TPA 操作层中的 AURM 模块执行 $(x,input)\rightarrow(sto,chal,upd)$ 请求分析，判断请求类型。其中，*sto* 表示存储元数据到 TPA，*chal* 表示生成挑战信息，*upd* 表示生成更新挑战信息。

步骤 4　随后，TPA 内的 GCM 模块生成挑战信息 $chal\leftarrow(c,k_1,k_2)$，其中，$c$ 为数据块个数且 $1\leqslant c\leqslant n$，k_1 和 k_2 为 TPM 每次审计而生成的随机置换密钥且 $k_1\in Z_p$，$k_2\in Z_p$，最后，将 *chal* 发送给 CSP。

3) 应答阶段（response phase）

步骤 1　CSP 接收到 TPA 云联盟发送的挑战 *chal*，执行 $prepro(chal)\rightarrow(\alpha,chal')$，$\alpha$ 表示要挑战的内容状态，即持有性证明或数据更新操作。

步骤 2　CSP 确认是挑战信息后，首先要确定在 $[1,n]$ 中要进行挑战的子集 $I=\{s_j\},1\leqslant j\leqslant c$，利用伪随机置换 $\pi_{key}(\cdot)$ 计算 $s_j=\pi_{k_1}(j)$，并利用伪随机函数 $f_{key}(\cdot)$ 计算 $v_{s_j}=f_{k_2}(j)(1\leqslant j\leqslant c)$。对于 $i\in I$，CSP 的计算式为

$$\mu^*=\sum_{i=s_1}^{s_c}v_i m_i \tag{2}$$

$$\sigma=\prod_{i=s_1}^{s_c}\sigma_i^{v_i} \tag{3}$$

然后将证据 $P=\{\mu^*,\sigma\}$ 发送给 TPA 云联盟。

4) 验证阶段（verification phase）

步骤 1　TPA 云联盟收到 CSP 发送来的证据 P，首先在数据保护组件中对其进行隐私保护，即选择一个随机元素 $r\leftarrow Z_p$，并利用相同的随机函数 $r=f_{k_3}(chal)$，其中，k_3 为 TPM 为每次审计而生成的随机函数密钥。然后计算

$$R=u^r\in G \tag{4}$$

$$\mu=\mu^*+rh(R)\in Z_p \tag{5}$$

最后将 (μ,σ,R) 作为证据发送给 TPA。

步骤 2　TPA 中操作层中 APRM 模块接收到经过数据保护组件处理后的应答信息后，首先对应答信息执行 $prepro(P)\rightarrow(ch\_request,up\_request)$，判断其是“挑战完整性应答”还是“更新完整性应答”。

步骤 3　如果是挑战完整性应答，则将证据 (μ,σ,R) 传递到 IVM 模块，如果是后者则根据生成的挑战 *chal* 以及认证元数据库，做出完整性验证，即利用证据 (μ,σ,R) 计算 $s_j=\pi_{k_1}(j)$ 和 $v_{s_j}=f_{k_2}(j)$，其中，$1\leqslant j\leqslant c$，最后该模块验证为

$$e(\sigma,g)\stackrel{?}{=}e(\prod_{i=s_1}^{s_c}(H(i)^{v_i}u^{\mu-rh(R)}),v) \tag{6}$$

如果式(6)相等则发送 success 给用户，否则发送 fail 给用户。

正确性证明如下所示。

$$\begin{aligned}e(\sigma,g)&=e(\prod_{i=s_1}^{s_c}\sigma_i^{v_i},g)\\&=e(\prod_{i=s_1}^{s_c}(H(i)u^{m_i})^{x\cdot v_i},g)\\&=e(\prod_{i=s_1}^{s_c}(H(i)^{v_i}u^{v_i m_i}),g)^x\\&=e(\prod_{i=s_1}^{s_c}(H(i)^{v_i}u^{\sum_{i=s_1}^{s_c}v_i m_i}),g^x)\\&=e(\prod_{i=s_1}^{s_c}(H(i)^{v_i}u^{\mu^*}),v)\\&=e(\prod_{i=s_1}^{s_c}(H(i)^{v_i}u^{\mu-rh(R)}),v)\\&=e(\prod_{i=s_1}^{s_c}(H(i)^{v_i}u^{\mu}\cdot u^{-rh(R)}),v)\end{aligned}$$

3.2　更新验证方案

本文模型支持动态更新操作[19]，采用基于 MHT（merkle Hash tree）的数据结构对用户存储在云端的数据进行修改、插入、删除等操作的验证。利用叶子节点的辅助认证信息（AAI, auxiliary authentication information）和用户签名的 MHT 的根 $root_{ssk}$ 来确定要进行动态操作数据块的位置和保证 CSP 进行用户要求的操作。

1) 初始化阶段（setup phase）

初始化步骤与完整性验证过程相同。不同的是在步骤 5 时，用户对文件块生成用户签名 $\sigma_i=(H(i)u^{m_i})^x\in G,1\leqslant i\leqslant n$，并将每个标签值进行散列运算，将结果存储在 MHT 的叶子节点中。

然后对叶子节点两两向上再进行散列运算（若叶子节点为奇数，则直接对自己进行散列运算）生成 MHT 的根 root，并记录每个叶子节点的辅助认

证信息[19]，最后将其根 root 和 AAI 进行签名。

$$root_{ssk} \leftarrow sig_{ssk}(root) \tag{7}$$

$$AAI_{ssk} \leftarrow sig_{ssk}(AAI) \tag{8}$$

发送 $\{F, \phi = \{\sigma_i\}_{1 \leqslant i \leqslant n}, T, root_{ssk}, AAI_{ssk}\}$ 到 CSP，发送 $\{\phi = \{\sigma_i\}_{1 \leqslant i \leqslant n}, T, root_{ssk}, AAI_{ssk}\}$ 到 TPA 作为元数据。

2) 更新挑战阶段（update challenge phase）

TPA 云联盟收到用户的更新挑战请求后，要求 TPA 验证 CSP 是否进行相应的更新操作，包括对文件的修改、插入、删除。

TPA 云联盟将任务分配到特定的 TPA。TPA 操作层内的 AURM 模块分析结果为 r_2 生成更新挑战请求。GUCM 模块收到更新挑战请求 r_2，执行 $(T_5, root'_{ssk}, AAI'_{ssk}) \rightarrow upd\_chal$，其中，$T_5$ 表示认证元数据信息，$root'_{ssk}$ 表示更新后的 MHT 的根，AAI'_{ssk} 表示更新后的叶子节点的辅助认证信息。最后将生成的更新挑战请求 upd_chal 发送给 CSP。

3) 更新应答阶段（update challenge phase）

CSP 收到更新挑战请求 upd_chal 后，执行生成证据算法 $Gen\_proof_u(spk, \phi, root_{ssk}, AAI_{ssk}, upd\_chal) \rightarrow P_u$，输出更新操作证明 P_u 并发送至 TPA 云联盟。

4) 更新验证阶段（update verification phase）

TPA 操作层内的 UVM 模块接收到 CSP 发送来的更新挑战证据 P_u。随后运行 $Verify\_upd(upd\_chal, P_u) \rightarrow$ (true, false) 算法，即核对更新操作证明 P_u 中更新后的 MHT 的根节点和叶子节点位置信息 AAI 以及更新后的文件块标签与签名是否一致，最后将结果发送给用户。

4 安全性分析

本节从可信度量的安全性、区块链的安全性以及可证明安全方面提出 4 个问题，并分别进行了安全性分析证明。

问题 1 TPM 模块能否正确地对 TPA 进行完整性度量。

利用 TPM 模块设置在 TPA 的底层硬件，可以收集 TPA 运行环境中静态度量值，并与基准数据库进行对比验证，如果验证不通过则通过 VM 管理模块挂起此 TPA 的云资源。具体可信度量方法可采用树形可信度量模型 TSTM 来提高度量模型的可拓展性[20]。在 TPA 工作层进行审计任务时，需要对其进行动态度量，包括 TPA 进行审计任务时的可信证据收集以及运行环境的可信性动态验证机制。其具体实现可采用“流嵌入法”[21]对 TPA 工作流程中的命令控制流和数据流嵌入计算度量值的程序。所以根据以上方法和模型的使用，可保证 TPM 模块正确地对 TPA 进行完整性度量。

问题 2 TPA 度量值和 TPA 操作记录以及审计参数能否正确地保存在区块链中。

在区块链的形成过程中，采用 TPA-LECA 领导选举共识算法选举出领导节点，对 TPA 的度量值、操作记录、审计参数的生成和保存进行监督，并将其以正确的格式打包进区块。可信区块链中的领导节点由 TPA 的区块链层全节点选举而出，其完整性和打包信息的公正性由区块链中其他节点的记账本和计算验证保证。此外，所有网络节点都有自己的归属 TPA，即不会有外网的任意节点加入此网络中进行恶意的广播虚假信息。并且，TPA 云联盟高级管理员也可以对领导节点的工作进行监督，以此保证了 TPA 的度量值以及 TPA 的操作记录、审计参数能够正确完整地保存在区块中，形成区块链。

问题 3 区块链在一定容忍度下，是否是不可更改和不可伪造的。

采用密码学技术来保证消息传递的完整性和真实性，消息的发送者要对消息的散列值进行签名。本文定义 $\langle m \rangle_{\sigma_i}$ 为节点 i 对消息 m 的电子签名，$D(m)$ 为消息 m 的散列值。本文算法对由 n 个共识节点组成的共识系统提供 $f = \dfrac{n-1}{3}$ 的容错能力，其中，$n = |R|$ 为参与共识的节点数，R 为共识节点集合。

由于节点广播的内容包含发送者的签名，恶意记账节点无法伪造节点，它只能试图将系统的状态回退到过去，从而使系统发生“分叉”，即伪造一个区块链。本文假设系统所在的网络环境恰好将所有的共识节点分割成 3 个部分，即 $R = R_1 \cup R_2 \cup F$，且 $R_1 \cap R_2 = \varnothing$，$R_2 \cap F = \varnothing$，$R_1 \cap F = \varnothing$。假设 R_1 和 R_2 都由诚实的记账节点组成，且已形成网络孤岛，各自只能与自己所在的集合内的节点通信；F 全部都是恶意记账节点且已经合谋，可以统一行动；此外，F 的网络条件允许它们和任意节点进行通信，包括 R_1 和 R_2。

如果 F 想要系统发生“分叉”，只需与 R_1 达成共识并发布区块，并在不通知 R_2 的情况下与之达成

第二次共识，“撤销”与 R_1 的共识。

若想满足以上条件，需满足 $|R_1|+|F|\geqslant n-f$，且 $|R_2|+|F|\geqslant n-f$。最坏情况下有 $|F|=f$，即恶意节点的数量达到系统所能容忍的最大值，则上述关系变为 $|R_1|\geqslant n-2f$，且 $|R_2|\geqslant n-2f$。两式相加得 $|R_1|+|R_2|\geqslant 2n-4f$，化简后得 $n\leqslant 3f$。已知 $f=\dfrac{n-1}{3}$，这与 $n\leqslant 3f$ 矛盾，所以可证明系统在容错范围内无法被分叉。

问题 4　基于双线性对和 Diffie-Hellman 问题，CFTPA-DIV 方案模型能否在随机预言模型中保证数据的完整性。

即要证明在基于双线性对和 Diffie-Hellman 困难问题的随机预言模型中，除了回复正确验证信息外，敌手只有可忽略不计的概率赢得敌手游戏。

证明　在这里，CSP 被认为是敌手，而 TPA 被认为是仿真者，其控制着随机预言机 。

给定 $(g,g^{\omega},h)\in G$，仿真者即 TPA 需要输出 h^{ω}。设 $v=g^{\omega}$，$u=g^{a}h^{b}$，$a,b\leftarrow Z_p$ 为仿真者选取的随机值。在每次挑战中，仿真者选取 $r_i\leftarrow Z_p$，并执行随机预言模型 $H(i)=\dfrac{g^{r_i}}{g^{am_i}h^{bm_i}}$。

当 $u=g^{a}h^{b}$ 时，仿真者可以计算的签名集合为

$$H(i)u^{m_i}=\left(\frac{g^{r_i}}{g^{am_i}h^{bm_i}}\right)u^{m_i}=\left(\frac{g^{r_i}}{g^{am_i}h^{bm_i}}\right)g^{am_i}h^{bm_i}=g^{r_i} \tag{9}$$

$$\sigma_i=(H(i)u^{m_i})^{\omega}=(g^{\omega})^{r_i} \tag{10}$$

诚实的 CSP 会将 $P=(\mu,\sigma,R)$ 返回给 TPA，并满足

$$e(\sigma,g)=e\left(\prod_{i=s_1}^{s_c}H(i)^{v_i}u^{\mu}R^{-h(R)},v\right) \tag{11}$$

如果给定 r 相同，敌手将 $P'=(\mu',\sigma',R)$ 作为回应，也同样满足

$$e(\sigma',g)=e\left(\prod_{i=s_1}^{s_c}H(i)^{v_i}u^{\mu'}R^{-h(R)},v\right) \tag{12}$$

很明显 $\mu'\neq\mu$，否则 $\sigma'=\sigma$，$P'=P$。定义 $\Delta\mu=\mu'-\mu$。令式(12)除以式(11)得

$$\begin{aligned}e\left(\frac{\sigma'}{\sigma},g\right)&=e\left(\prod_{i=s_1}^{s_c}\left(\frac{\sigma_i'}{\sigma_i}\right)^{v_i},g\right)\\&=e\left(\prod_{i=s_1}^{s_c}\left(\frac{u^{m_i'v_i}}{u^{m_iv_i}}\right)^{\omega},g\right)\\&=e(u^{\sum m_i'v_i-m_iv_i},v)\\&=e(u^{\Delta\mu^*},v)\end{aligned}$$

既然对于所有验证等式给定的 r 相同，则有

$$e\left(\frac{\sigma'}{\sigma},g\right)=e(u^{\Delta\mu},v) \tag{13}$$

将 $u=g^{a}h^{b}$ 代入式(13)得

$$e(\sigma'\sigma^{-1},g)=e((g^{a}h^{b})^{\Delta\mu\omega},v)=e(v^{a\Delta\mu}(h^{b\Delta\mu})^{\omega},g)$$

整理得

$$e(\sigma'\sigma^{-1}v^{-a\Delta\mu},g)=e(h^{b\Delta\mu},v)=e(h,v)^{b\Delta\mu}=e(h^{\omega},g)^{b\Delta\mu}$$

进一步整理和化简得

$$e(\sigma'\sigma^{-1}v^{-a\Delta\mu},g)^{\frac{1}{b\Delta\mu}}=e(h^{\omega},g) \tag{14}$$

根据双线性特性，从式(14)中可以整理得出 $h^{\omega}=(\sigma'\sigma^{-1}v^{a\Delta\mu},g)^{\frac{1}{b\Delta\mu}}$。为了分析本文能否根据求解式(14)得到 h^{ω}，本文只需要计算 $b\Delta\mu=0\bmod p$ 即可。因为 b 为挑战者所选并且对敌手隐藏，$b\Delta\mu=0\bmod p$ 的概率仅为 $\dfrac{1}{p}$，可以忽略不计。如果敌手概率性地在此实例中成功，则仿真者能够解决离散对数问题。

证毕。

综合以上 4 个问题和安全性分析，可以将可信计算安全性、区块链安全性与完整性验证的安全性进行有效结合，整体上达到了整个验证模型的安全可信。

5　性能分析

5.1　模型的理论分析

在模型分析之前，本文设定在不增加云用户的开销前提下（或减少云用户开销），在 TPA 和 CSP 上增加一定开销是可以接受的。CFTPA-DIV 模型中的主要审计过程是在 TPA 云联盟环境下单个 TPA 的操作层完成的。相较于传统公开型 PDP 验证模型，本文模型增加了 TPA 云联盟系统平台的管理功能和 TPA 区块链层的工作流程来保证 TPA 为用户提供诚实可靠的数据完整性验证工作。此外，TPA

云联盟系统平台中的组件初始化时需要认证和交互。所以在整体的验证过程中计算开销、通信开销、存储开销等方面会有一定的增加。

这里，本文选择与同样针对恶意 TPA 提出模型方案的 Huang 等[8]的方案和 Wu 等[9]的方案进行对比。其中，Huang 等[8]的方案是对 Xu 等[7]的方案进行了改进，利用复审方案来核查 TPA 是否正确地执行审计任务。Wu 等[9]的方案采用一种安全轻量级的审计模型，令用户对恶意 TPA 进行审计核查。基于 TPA 云联盟的数据完整性验证方法最大的优势是通过可信计算技术和区块链技术保证了 TPA 的可信审计过程，规避了 TPA 的恶意行为。3 种方案的特点如表 2 所示。

表 2　　相似模型方案对比

方案	隐私保护	动态操作	多文件处理	可信第三方
Huang 等[8]	支持	不支持	不支持	无证明
Wu 等[9]	不支持	不支持	不支持	无证明
本文方案	支持	支持	支持	可证明

Huang 等[8]的方案中利用 receive server 和 time server 来保证 TPA 准确准时地完成审计任务，然而引入的 server 实体的可信性无法被证实。Wu 等[9]的方案中将 CSP 返回的验证结果分为复杂计算和简单计算 2 个部分，前者由 TPA 完成专业计算，后者由用户完成简单验证。然而，如果用户最后验证计算失败，将会直接向 CSP 进行验证。此时用户计算负担不但大大增加，而且 TPA 与 CSP 可能已经串谋，用户无法得到真实的验证结果。

从计算开销方面分析 CFTPA-DIV 模型，即对所有实体参与到数据完整性验证方面的所有计算进行理论分析。假设只进行一次挑战设定数据 F 被分为 n 个数据块，TPA 进行挑战时随机选取 c 个数据块。在 CFTPA-DIV 模型中，主要计算开销分别是在初始化阶段的步骤 5 计算式(1)，云用户要对计算文件标签并生成用户签名。在应答阶段的步骤 2 中计算式(2)和式(3)，CSP 接收到 TPA 的挑战，做出应答生成证据。在验证阶段的步骤 1 中计算式(4)和式(5)，即 TPA 云联盟中的数据保护组件对 CPS 发送来的证据进行隐私保护即随机掩码技术；步骤 3 中验证式(6)是否成立和进行随机置换和随机函数运算。定义在数据完整性验证过程中的相关操作如表 3 所示，然后分别对 3 种方案进行计算开销的理论分析，如表 4 所示。

表 3　　相关操作及定义

操作	定义
A	在群 G 上的一次加法运算
M	在群 G 上的一次乘法运算
F	在群 G 上的一次除法运算
E	在群 G 上的一次指数运算
H	一次散列运算
P	一次配对运算
f	一次伪随机函数运算
π	一次伪随机置换运算
TEnc	加密运算[7]
TDec	解密运算[7]

从表 4 的 3 种模型方案的计算开销对比来看，CFTPA-DIV 模型的用户开销较其他 2 种模型少。分析原因，是因为在 Huang 等[8]的方案中用户要进行审计方案的设计以及审计方案的加密等操作，从而增加了用户的计算开销。在 Wu 等[9]的方案中，用户需要对 TPA 的计算结果再进行一次配对运算，进而验证 TPA 是否验证成功。在不考虑 Huang 的方案中引入不可信 server 实体和 Wu 等[9]的方案中如果验证失败用户还需要大量计算的基础上，此 2 种方案虽然对恶意 TPA 有一定效果，但是增加了用户的计算开销。而 CFTPA-DIV 方案则尽量减小用户的开销，使对 TPA 的审计工作放在 TPA 的云联盟中进行，并且用户可在 TPA 云联盟平台对 TPA 审计过程和状态进行监督。

在 Wu 等[9]的方案中，需要 CSP 重新生成签名标签，所以理论计算开销较大。在 TPA 计算开销方面，Wu 等[9]的理论计算开销较其他 2 种方案少，但是其增加了用户计算开销。Huang 等[8]的方案与 CFTPA-DIV

表 4　　模型方案计算开销

方案	用户计算开销	CSP 计算开销	TPA 计算开销
Huang 等[8]	nH+2nM+nE+1TEnc	1P+1H+2cM+cA+cE	(2+c)M+(3+c)E+2P+1D+1TDec
Wu 等[9]	1P+nH+nM+nE	nH+(n+2c−1)M+(n+c+1)E+(c−1)A	cM+(1+c)E+cH+1P
本文方案	nH+nM+nE	(2c−1)M+(c−1)A+cE+c(π+f)	(2c−1)M+(c+2)H+(c+3)E+A+f+2P

方案的计算开销需要利用仿真实验来实现进一步的对比分析。

5.2　CFTPA-DIV 模型的仿真实验与结果分析

CFTPA-DIV 模型是建立在基于可信区块链的 TPA 基础之上的，所以要首先进行基于可信区块链的 TPA 的环境搭建。实验物理主机采用 Windows10 64 位系统，硬件配置如下：英特尔酷睿 i5 四核 3.20 GHz 处理器，DDR3 1 600 MHz，8 GB 内存，120 GB 固态硬盘。在物理主机的 VMwareWorkstation12 虚拟机创建 TPA 实例，TPA 实例采用 Ubuntu 16.04 LTS 64 位系统，随后在 TPA 的系统中，采用基于 TPM2.0 的软件 TPM，即 ibmtpm1119 代替真实的硬件 TPM，并结合 ibmtss1119 软件协议栈创建 vTPM 实例。

在 vTPM 实例中利用 C 语言实现了原型系统，包括利用 PCR 寄存器进行完整性度量、对 VM 的开启及注销以及对该 TPA 的审计过程形成日志打包进区块链中等功能。然后在搭建的系统中使用 Pairing-Based Cryptography(PBC)库版本 0.5.14 和 OpenSSL 版本 1.0.2g 对验证过程提供加密、解密以及配对操作。

首先，对 3 种方案做定量的实验分析。假设用户的文件块总数与 TPA 所选取验证文件块数量相同，即 n=c=300。根据表 4 的计算开销对比，在 5.2 节中搭建好的可信区块链环境中，对 CFTPA-DIV 中的 TPA 运算操作进行模拟仿真实验，CU 和 CSP 运算和其他 2 个方案实验采用 Ubuntu 16.04 LTS 64 位系统和同样的 PBC 库、OpenSSL 库。分别对 3 种方案 CU、TPA、CSP 中的主要计算开销进行 20 次模拟仿真实验，并对实验结果取平均值。其中，实验中选择椭圆曲线形式为 MNT 曲线，其基础字段大小为 159 bit，嵌入度为 6，实验中安全等级选择 80 bit，即$|v_i|$=80、$|p|$=160。选择每个块的大小为 40 B。实验结果即其所各自所花费的时间如表 5 所示。

表 5　　模型实际计算开销对比

方案	数据量	CU/ms	CSP/ms	TPA/ms
Huang 等[8]	n=c=300	672.7	588.8	706.3
Wu 等[9]	n=c=300	647.9	1 143.4	590.5
本文方案	n=c=300	578.8	579.3	618.5

从表 5 可以看出，在实验选定的数据块数量下，CFTPA-DIV 模型在用户端较其他 2 种方案时间花费较低，TPA 的时间花费比其他 2 种方案时间花费较高，CSP 的时间花费相差不大。分析原因可能为，CFTPA-DIV 模型在 TPA 工作的处理上较其他 2 种方案有组件间交互的影响，而 CU 时间消耗的较少也达到了预期的目标。

其次，对 3 种方案分别增加选取块数，同样假设 n=c，并进行多次模拟仿真，实验结果分别如图 5 和图 6 所示。

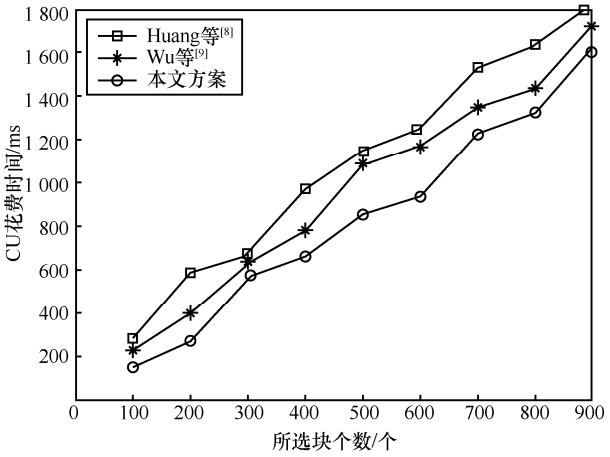


图 5　3 种方案选取相同块个数下 CU 花费时间对比

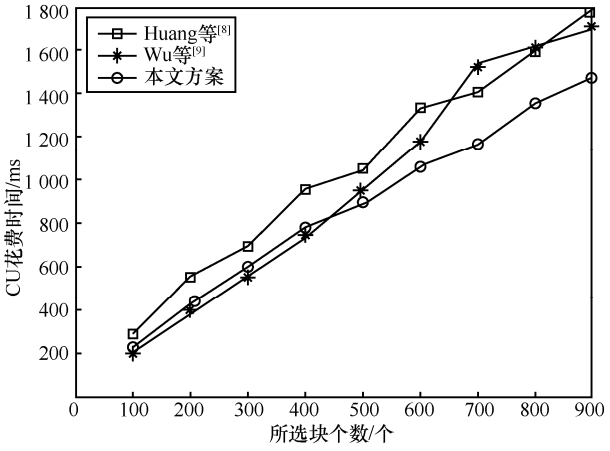


图 6　3 种方案选取相同块个数下 TPA 花费时间对比

图 5 和图 6 表明，本文方案在减少了用户开销的基础上，实现了利用 TPA 云联盟中的可信计算技术与区块链技术保证 TPA 进行审计任务的诚实可信。并且 TPA 云联盟系统具有可扩展性，可综合其他审计方案做出具体优化部署在审计方案管理组件中，使原本不安全的审计方案在 TPA 云联盟中变得安全可信。

6　结束语

本文针对数据完整性验证的公开审计第三方

审计机构的可信性问题，提出了一种基于可信区块链的数据完整性验证模型。该模型通过设计 TPA 云联盟系统以及在 TPA 云成员中构造关于操作记录的区块链，证明了可信 TPA 的可信性。通过对模型的分析以及仿真实验可以得出，本文模型在正确性的前提下不仅保障了公开审计的安全性，还有效地提高了数据完整性验证的工作效率。该模型可根据安全等级的不同适配相应的审计方案，适用于对安全等级需求较高的行业以及大多数需要公有验证云端数据的用户。并且在互联网与物联网的信息安全、云存储的数据安全等领域的应用研究中具有重要的意义和价值。

参考文献：

[1] WU J, PING L, GE X, et al. Cloud storage as the infrastructure of cloud computing[C]//IEEE International Conference on Intelligent Computing and Cognitive Informatics (ICICCI). 2010:380-383.

[2] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//The 14th ACM Conference on Computer and Communications Security. 2007: 598-609.

[3] WANG Q, WANG C, LI J, et al. Enabling public verifiability and data dynamics for storage security in cloud computing[C]//European Conference on Research in Computer Security. 2009: 355-370.

[4] WANG C, WANG Q, REN K, et al. Privacy-preserving public auditing for data storage security in cloud computing[J]. 2010, 62(2):525-533.

[5] ERWAY C, PAPAMANTHOU C, TAMASSIA R. Dynamic provable data possession[C]//ACM Conference on Computer and Communications Security. 2009:213-222.

[6] WANG Q, WANG C, REN K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel & Distributed Systems, 2011, 22(5):847-859.

[7] XU J. Auditing the auditor: secure delegation of auditing operation over cloud storage[C]//IACR Cryptology ePrint Archive. 2011: 304.

[8] HUANG K, XIAN M, FU S, et al. Securing the cloud storage audit service: defending against frame and collude attacks of third party auditor[J]. IEEE Transactions on Communication, 2014.

[9] WU Y L, LIN X, LU X C, et al. A secure light-weight public auditing scheme in cloud computing with potentially malicious third party auditor[J]. IEICE Transactions on Information & Systems, 2016(10): 2638-2642.

[10] 肖达, 杨绿茵, 孙斌, 等. 面向真实云存储环境的数据持有性证明系统[J]. 软件学报, 2016, 27(9):2400-2413.
XIAO D, YANG L Y, SUN B, et al. Provable data possession system for realistic cloud storage environments[J]. Journal of Software, 2016, 27(9):2400-2413.

[11] FRANCESCO P S , VLADIMIRO S , LUCA N I, et al. FaaS: Federation-as-a-Service[J]. 2016.

[12] 田俊峰, 常方舒. 基于 TPM 联盟的可信云平台管理模型[J]. 通信学报, 2016, 37(2):1-10.
TIAN J F, CHANG F S. Trusted cloud platform management model based on TPM alliance[J]. Journal on Communications, 2016, 37(2): 1-10.

[13] BERGER S, GOLDMAN K A, PEREZ R, et al. vTPM: virtualizing the trusted platform module[C]//Conference on Usenix Security Symposium. 2006:21.

[14] 张健. 云计算服务等级协议(SLA)研究[J].电信网技术, 2012(2): 7-10.
ZHANG J. Study on cloud computing SLA[J]. Telecommunication network technology, 2012(2):7-10.

[15] SYED S R, KATIE C, ABDUL R. Cloud data integrity using a designated public verifier[C]//HPCC-CSS-ICESS.2015.

[16] HU V C, KUHN D R, FERRAIOLO D F. Attribute-based access control[J]. Computer, 2015, 48(2):85-88.

[17] CASTRO M, LISKOV B. Practical byzantine fault tolerance[C]// OSDI. 1999: 173-186.

[18] 范捷, 易乐天, 舒继武. 拜占庭系统技术研究综述[J]. 软件学报, 2013(6): 1346-1360.
FAN J, YI L T, SHU J W. Research on the technologies of Byzantine system[J]. Journal of Software, 2013, 24(6): 1346-1360.

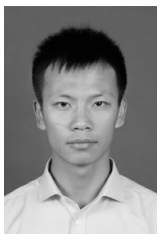
[19] WANG Q, WANG C, REN K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel & Distributed Systems, 2011, 22(5):847-859.

[20] 周振吉, 吴礼发, 洪征, 等. 云计算环境下的虚拟机可信度量模型[J]. 东南大学学报(自然科学版), 2014, 44(1): 45-50.
ZHOU Z J, WU L F, HONG Z, et al. Trustworthiness measurement model of virtual machine for cloud computing[J]. Journal of Southeast University(Natural Science Edition) , 2014, 44(1): 45-50.

[21] 吴昊, 毋国庆. 程序的动态完整性:模型和方法[J]. 计算机研究与发展, 2012, 49(9):1874-1882.
WU H, WU G Q. Dynamical integrity of codes:model and method[J]. Journal of Computer Research and Development, 2012, 49(9): 1874-1882.

[作者简介]

田俊峰（1965−），男，河北保定人，河北大学教授、博士生导师，主要研究方向为信息安全与分布式计算。



李天乐（1990−），男，河北沧州人，河北大学硕士生，主要研究方向为信息安全与分布式计算。