文章编号:1000-2367(2023)04-0093-07 DOI:10.16366/j.cnki.1000-2367.2023.04.012

一种基于区块链的云镜像完整性监测方法

马威[1],曹刘洋[2],管朋朋[1]

(1.华北水利水电大学 信息工程学院,郑州 450046;2.重庆理工大学 两江人工智能学院,重庆 401135)

摘 要:针对云环境中镜像文件遭到篡改攻击这一问题,结合区块链技术提出了一种云镜像完整性监测方法.首先,每隔一段时间计算所有云镜像的哈希信息,并将哈希信息记录在 Merkel 树数据结构中;然后,将这一 Merkel 树打包成为区块并构成区块链;最后,周期性地对当前区块中 Merkel 树根节点的验证以实现对全部云镜像的完整性验证,并在发现篡改后定位被篡改的镜像文件.经过实验验证,所提出的方法相比之前的方法有 3% 的效率提升,同时经过理论分析,方法具有安全性、可追溯性以及可扩展性等特点.

关键词:云计算;完整性监测;区块链

中图分类号:TP309 **文献标志码**:A

近年来,云计算由于低成本、高可靠、高可用以及弹性可伸缩的优势,在各行各业得到广泛应用[1-2].快速部署是云计算中的关键技术之一,这一技术能够按需动态组织云环境中的软件、硬件资源,同时能及时响应用户动态的、多样化的计算需求.在当前的快速部署技术中,往往需要一个预定义的模板,其中包含了组织好的软件资源,这一模板被称为云镜像.当用户发起部署请求时,快速部署机制能够基于云镜像快速实例化用户所需的云计算资源[3].例如在 IaaS(Infrastructure as a Service)中,用户获得的是一个基于模板实例化的虚拟主机;在 PaaS(Platform as a Service)中,用户获取的是一个基于模板实例化的容器.快速部署随着这些年来云计算技术自身的发展而发展,如虚拟机克隆技术,能够从运行态的父虚拟机中动态克隆得到一个新的虚拟机[4],或是运用 fork 函数的思想,利用父虚拟机并行克隆出大量子虚拟机,提高部署效率[5-6];Copy-On-Write技术也被运用在虚拟机的快速部署中,它仅传输模板中需要修改的数据块,因此能够极大提高部署的速度[7-8];有方法进一步将云镜像的存储也分布式化和虚拟化,使用存储池和并行传输来进一步提高部署的速度[3,9-10].

关于快速部署的研究进展主要集中在效率方面[11],即如何使部署速度更快,以便更快捷地响应用户的服务请求,而云镜像则在快速部署中始终扮演不可或缺的核心角色.但是,云镜像往往会成为攻击者的攻击目标.例如篡改云镜像,在其中注入恶意代码,进而构建僵尸网络[12],最近几年中,又频繁出现攻击者篡改云镜像,植入挖矿软件从而获得经济利益的事件[13].因此,对云镜像的保护应当是云计算安全的重要一环.

对云镜像的保护,主要体现在防止云镜像被篡改,即确保云镜像的数据完整性不会遭到破坏.而云计算环境中数据完整性的保护一直是研究重点之一,最普遍采用的方法是使用镜像文件或数据文件的散列值对数据完整性进行监控,但这种传统方法的效率低下,面对云计算环境中海量的镜像数据处理速度无法满足云环境的需求.在使用散列值的基础上,引入了哈希链来确保数据完整性.文献[14-15]使用了哈希链,并结合了其他密码学技术提出了在分布式系统上的证书撤销机制,能够检测到对数据块修改和删除的动机.但是基于哈希链的方法与传统的基于散列值的方法类似,需要大量的验证工作,难以适应数据量巨大的云环境.另一类广泛运用的检测数据完整性的方法是使用 Merkel 树,Merkel 树是一种树形结构,其中每个叶结点是数

收稿日期:2022-04-04;**修回日期**:2022-05-11.

基金项目:国家自然科学基金青年基金(62107014);河南省科技攻关项目(212102210100);河南省高等学校重点科研项目(20A413008;21B520022;20B520040);河南省高等教育教学改革研究与实践项目(2021SJGLX581).

作者简介(通信作者):马威(1985-),男,河南郑州人,华北水利水电大学副教授,博士,研究方向为信息安全,E-mail:mawei@ncwu.edu.cn.

据块的哈希,每个非叶子结点是其子结点的哈希.文献[16－18]使用 Merkel 树对云存储中的数据进行了动态验证,文献[19－20]使用 Merkel 树实现了云数据的保护和验证.相比较于哈希链和传统散列值的完整性验证方法,Merkel 树最大的优点在于它可以通过对根节点的单次哈希运算实现对所有叶子节点(即对应的数据)的完整性验证,能够有效提高验证效率,因此 Merkel 树在完整性验证中有着十分广泛的应用.但是,由于 Merkel 树是静态构建的,与云计算环境中动态性的要求难以匹配.

近年来,区块链技术的发展为完整性检测提供了新的方法和思路.区块链是随着加密数字货币出现的一种数据结构,具有去中心化、时序性、安全可信等特点[21].在每一个区块中,将不同的事务/交易进行哈希运算后,使用 Merkel 树的形式进行存储,确保了事务/交易的完整性;使用链式结构将区块进行组织,让信息可以动态追溯.从本质上而言,区块链是一种不可篡改的账本,保障数据完整性的核心内容相匹配,因此区块链在数据完整性保护方面具有很强的研究意义[22].

在本文中,结合区块链技术提出了一种云镜像完整性监测方法.首先,对云环境中可用的云镜像进行哈希运算,形成一棵 Merkel 树并记录在一个区块中,该 Merkel 树记录了所有镜像的哈希信息,通过对它根节点的验证即可实现对所有镜像的完整性验证;然后,以一个固定的时间间隔重复上一步,并形成区块链;最后,实时监控最新区块中 Merkel 树的根节点,发现变化则意味着有镜像的完整性遭到了破坏,进而迅速定位被篡改的镜像文件.方法的创新点主要体现在:(1)引入 Merkel 树作为存储云镜像完整性信息的静态数据结构;(2)使用区块链机制实现云镜像完整性的动态存证和可追溯;(3)通过实验证明,本文提出的方法能够带来 3%以上的性能提升.

1 云镜像完整性问题分析及模型

1.1 问题分析

在云计算环境中,云镜像主要用来快速实例化所需的虚拟机实例,单个云镜像的生命周期如图 1 所示.

云镜像是云计算快速部署机制的核心.一般情况下,云镜像会集中存储在云镜像服务器中,当收到用户发起的部署请求后,云中的调度机制会寻找有足够空闲资源的主机,然后将相应云镜像复制(全部复制或使用 Copy-On-Write 机制部分复制)到目标主机,并使用该镜像实例化虚拟机.云镜像也支持云用户之间和云服务提供商之间的共享.此外,云镜像还会根据需求(软件升级、安装补丁等)进行更新.基于某个云镜像实例化的虚拟机,其中必定包含和运行云镜像中预置的软件程序,因此对云镜像进行篡改,在其中注入恶意软件、僵尸程序或挖矿软件,是针对云计算环境的一种有效攻击手段,保护云镜像的完整性至关重要.

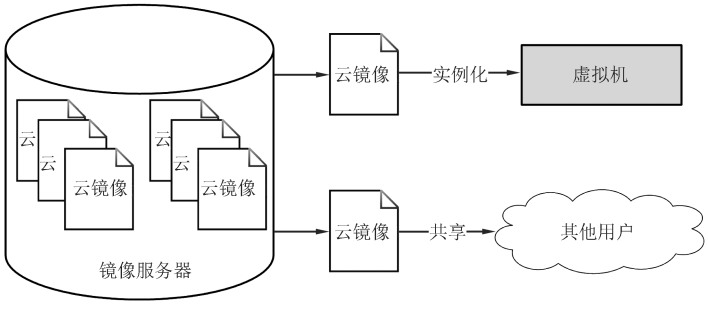

图1 云镜像的生命周期

Fig.1 Life cycle of a cloud image

1.2 攻击者模型

首先分析攻击者的目的.对于攻击者而言,其目的主要在于:(1)利用云计算平台可观的算力来进行发起于网络内或者网络之间的恶意攻击;(2)浪费云计算资源,直接损害云计算供应商的利益;(3)恶意占用云计算平台算力资源,对云计算用户造成间接的危害;(4)利用云计算海量的网络资源和数据资源非法获取云计算平台中用户的隐私数据,给云计算平台企业、用户和与之互联的网络主机带来直接和间接的危害.攻击者能力如表 1 所示.

表 1　攻击者能力

Tab. 1　Capabilities of the attacker

攻击者所具有的能力	攻击者没有的能力
攻击者对云计算中的镜像文件敏感,即攻击者能够发现目标云环境中的镜像文件; 攻击者在入侵阶段的最终攻击目标就是云环境中的镜像文件,对镜像文件进行篡改.	攻击者不会改变云镜像的数量,即不会增加或减少云镜像; 攻击者对除镜像文件之外的其他类型的文件不敏感,即不会篡改除镜像文件外其他类型的文件; 攻击者不清楚相关保护机制的原理.

这一攻击者模型的定义旨在明确攻击者的目标、能力范围,进而在这些基础上设计方法.

2　基于区块链的云镜像监测方法

本文提出的方法可用算法 1 中的伪代码描述.

算法 1　云镜像监测

输入:镜像文件 $M_t^{(1)},M_t^{(2)},\cdots,M_t^{(n)}$

输出:被篡改的文件编号 i

```
if no New Block then
    MT_t ← MerkelTree(M_t^(1), M_t^(2), ..., M_t^(n))
    Build New Block with MTt
end if
MT_{t+δ} ← MerkelTree(M_{t+δ}^(1), M_{t+δ}^(2), ..., M_{t+δ}^(n))
Node_t ← MTt. RootNode
Node_{t+δ} ← MT_{t+δ}.RootNode
while Diff (Node_t, Node_{t+δ}) = TRUE do
    if Node_t is Leaf Node then
        return id of Node_t
    else
        Node_t ← Node_t.ChildNode
        Node_{t+δ} ← Node_{t+δ}.ChildNode
    end if
end while
```

在这一方法中,主要包括以下几个步骤:镜像完整性度量步骤、区块链构造步骤、镜像完整性监测步骤和镜像篡改检测步骤.

2.1　镜像完整性度量

方法的第一步,需要对镜像进行完整性度量并构成区块链所需的 Merkel 树.在这一步所产生的 Merkel 树中,其叶子节点与云环境中的镜像一一对应,即每个叶子节点存放的是镜像文件的哈希值.用 $M_1,M_2,\cdots,M_n$ 表示云环境中的 n 个镜像,$H(M)$表示其对应的哈希值,则 Merkel 树的叶子节点的内容分别为$H(M_1)$,$H(M_2)$,…,$H(M_n)$.Merkel 树中非叶子节点中存放的哈希值,则由对其左右孩子节点的哈希值进行二次哈希得到,如 M_1 和 M_2 所对应的父节点中存放的内容为 $H(H(M_1)\|H(M_2))$.最终产生的 Merkel 树的结构如图 2 所示.

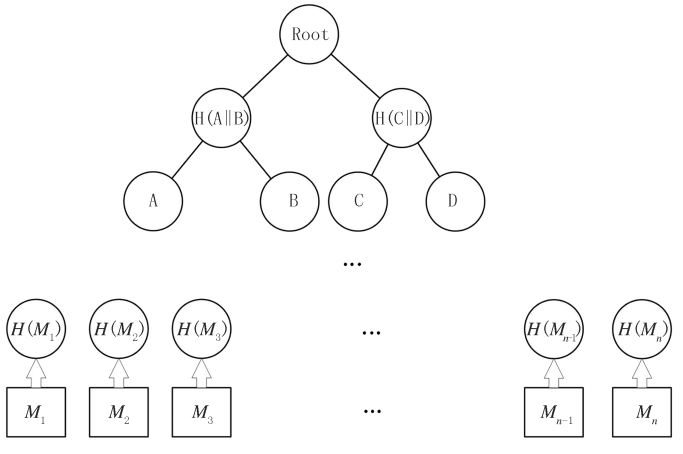

图2　完整性度量得到的Merkel树

Fig.2　Merkel Tree generated by integrity measurement

由于 Merkel 树是一棵完全二叉树,而镜像文件的数量未必满足这一要求,因此在生成 Merkel 树时,需要对树的高度加以考虑.用 h 表示完整性度量得到的 Merkel 树的高度,那么在 n 个云镜像的情况下,必然有 $n\leq 2^{(h-1)}$.为了生成高度最小的 Merkel 树,需要遵循

以下步骤:

(1)计算镜像 M_1 至 M_n 的哈希值,在 Merkel 树的 $2^{(h-1)}-1$ 至 $2^{h-1}+n-2$ 叶子节点中存储;

(2)使用空白值填充 Merkel 树 $2^{h-1}+n-1$ 至 $2^{h-1}-2$ 的叶子节点;

(3)除叶子结点外的所有非叶子结点按照从最大非叶子结点递减的顺序递归地从其左右孩子的哈希值中计算出新的等长哈希值,逐层向上递归计算哈希值,最终计算到 Root 结点的哈希值.

经过这一过程构造的 Merkel 树,其根节点中存储的哈希值对所有镜像的完整性敏感,即任何镜像的修改都会导致根节点哈希的变化.

2.2 区块链的构成

为镜像文件生成相应的完整性度量 Merkel 树后,可以进一步构造区块链.在 t_0 时刻,系统生成创世区块;每隔一段时间,系统会重新为每个镜像进行完整性度量,重构 Merkel 树,并将新的 Merkel 树打包到区块链中.区块及区块链的结构如图 3 所示.

每个区块包含区块头和区块体,其中区块头中包含当前的 Merkel 树的根节点,以及指向前一个和后一个区块的指针、时间戳和一个随机数;区块体则包含了当前 Merkel 树的根节点以外的其他节点.系统在 t_0 时刻生成创世区块后,会在 $t_1,t_2,\cdots,t_m$ 时刻生成区块 1,区块 2,区块 m 等,其时间戳 $t_k,k\in[1,m]$被记录在区块头中,区块体则是在这个时刻重构的 Merkel 树.为了防止 TOCTOU(Time-Of-Check-Time-Of-Use)攻击,系统通过调整随机数的计算难度来控制区块产生的速度,即区块产生的时间间隔 $\delta=t_1-t_0=t_2-t_1=\cdots=t_m-t_{m-1}$是一个恒定值,且需要根据云镜像的具体情况确定其具体取值.

在本文提出的方法中,Merkel 树保存了静态的云镜像完整性信息,区块链则实现了信息的动态存证.

2.3 镜像完整性监测

在构造了相应 Merkel 树和区块链的基础上,可以对当前时刻镜像库中的云镜像完整性进行监测.监测的目的在于确认云镜像没有被篡改,由于区块链中 Merkel 树的存在,因此对所有镜像完整性的监测主要在于对 Merkel 树根节点的监测.首先,方法依据时间戳检查当前区块的生成时间,即检查最新区块是否生成:如果最新区块尚未生成,则生成最新区块;如果最新区块已经生成,则从最新区块中读取当前 Merkel 树中根节点中存储的哈希值,监测最新区块中的一根哈希与前一个区块相比是否变化;当监测到根哈希变化时,由于 Merkel 树的特性,即可认为是某个镜像被篡改,进而启动检测机制对篡改进行检测.

2.4 镜像篡改检测

当监测机制发现了镜像篡改后,会启动检测机制定位被篡改的镜像.首先,依据现有镜像构建新的 Merkle 树,将其根结点与当前区块中 Merkle 树的根节点进行对应比较,如果这两结点的哈希值相同的话,则认为没有发生篡改;如果这两结点的哈希值不同的话,则需要进行进一步的判断:先判断此结点是否为 Merkle 树的叶子结点,如是,则该节点对应的镜像文件已经遭到篡改;如果此结点不是叶子结点,则分别对此结点的左孩子结点和右孩子结点进行以上判断.通过这种递归式的判断,进而定位到被篡改的镜像文件所对应的叶子节点.

3 性能分析及讨论

3.1 时间复杂度分析

本文从遍历、监测和监测 3 个角度分析时间复杂度.在有 n 个云镜像的云环境中,之前的方法中使用传统哈希值或文献[10-11]中使用的哈希链方法对云镜像的完整性验证时,由于两者均为线性结构,因此遍历需要逐一处理,其时间复杂度为 $O(n)$;监测镜像完整性时,需要逐一比对,因此监测的时间复杂度也是 $O(n)$;检测被篡改的镜像文件时,同样需要逐一处理,因此时间复杂度是 $O(n)$.而本文提出的方法,在遍历时,首先需要为每个镜像进行完整性计算并建立 Merkel 树,因此时间复杂度是 $O(n)$;在监测镜像文件是否被篡改时,由于 Merkle 树的良好性质,仅需比对根结点处的哈希值,因此在镜像文件的监测阶段时间复杂度仅为$O(1)$.当发现根节点的变化,即需要检测和定位具体的被篡改镜像时,会依据上述流程执行递归操作:

(1)读取根结点的内容;

(2)判断根结点内容是否发生了变化,若没有变化则结束,若有变化则执行(3);

(3)若此节点为叶子结点,则此结点对应镜像文件遭到篡改,否则将此结点的左右孩子结点分别带入(2)中进行判断.

图 4 描述了一个定位两个被篡改镜像的示例.在图 4 所示的例子中,镜像和被篡改了.重构 Merkel 树后,首先,方法会发现根结点的哈希值变化;然后,方法会寻找根结点的孩子结点即 A 结点和 B 结点,比对其值与当前区块中的对应节点的值,发现 A 结点的值发生了变化而 B 结点的值并未改变,这意味着篡改发生在 A 结点的子孙结点;然后,方法再进一步比对 A 结点的孩子结点,即 C 结点和 D 结点.这样逐步递归,可以最终定位到被篡改的镜像文件.这一检测过程实质上是完全二叉树的查找过程,其时间复杂度为 $O(\log_2 n)$.表 2 总结了之前文献中使用哈希值或哈希链的方法与本文提出方法的时间复杂度的对比.

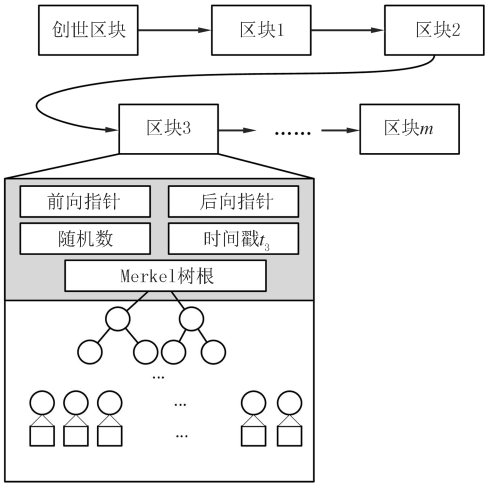

图3　区块链及区块的结构

Fig.3　Structure of blockchain and block

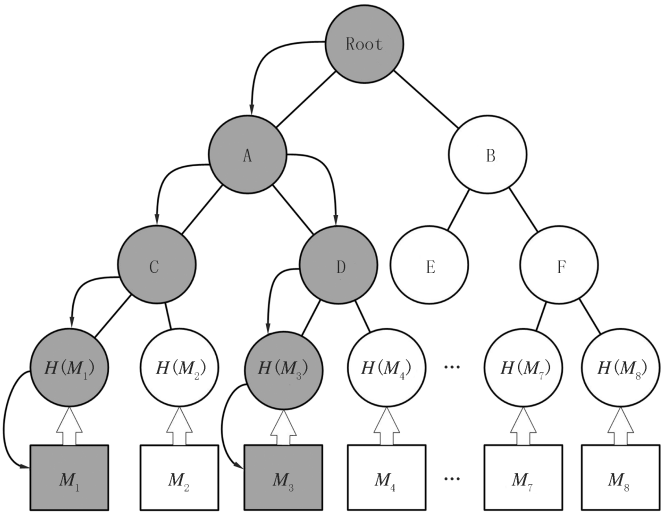

图4　定位篡改镜像

Fig.4　Locating the tampered images

3.2　实验测试

本文实现了一套原型系统以进一步验证所提出方法的性能.原型系统使用以太坊构建私有区块链,部署 5 个结点完成共识,选用 PoA(Proof of Authority)共识机制.其中创世区块文件中的 period 参数被设置为 300,即区块产生的时间间隔被设置为 5 min (300 s).因此,当前的云镜像每 5 min 会进行一次完整性检查,构造 Merkel 树并生成一个新的区块.原型系统中准备了 3 种不同的云镜像文件:(1)Windows 镜像文件,可用于快速部署 Windows 虚拟机;(2)RedHat Linux 镜像文件,可用于快速部署 Linux 虚拟机;(3)CentOS 镜像,其中内置了 LAMP 程序,可用于快速部署 LAMP 环境.这些云镜像文件的大小各不相同,通过组合这 3 种云镜像,实验在原型系统中模拟了 3 种应用场景,以检验方法的性能,其中场景 1 包含两个 CentOS 镜像以及 RedHat 镜像和 Winows 镜像各 1;场景 2 包含两个 CentOS 镜像和 5 个 RedHat 镜像;场景 3 包含 4 个 CentOS 镜像、10 个 RedHat 镜像以及 4 个 Windows 镜像.实验分别计算了在这 3 种场景下构建 Merkel 树和定位被篡改的镜像文件时的时间开销,以验证方法的性能,实验结果如图 5 所示.

表 2　时间复杂度对比

Tab. 2　Comparison of time complexity

类别	之前的方法[10-11]	本文的方法
遍历	$O(n)$	$O(n)$
监测	$O(n)$	$O(1)$
检测	$O(n)$	$O(\log_2 n)$

图 5 中的实验结果显示,当云镜像规模的增长时,构建 Merkel 树的时间还是定位篡改文件的时间开销都会随之增加.而在所有的场景下,本文提出的方法的性能表现都要优于之前基于哈希值或哈希链的方法,

并且云镜像的规模越大,本文方法所表现出来的优越性越强,例如在构建 Merkel 树的时间开销上,在场景 1 即云镜像总规模为 10.2 GB 时,本文方法相对于之前的方法有约 0.5%的性能优势,而在场景 3 即云镜像总规模为 50 GB 时,本文方法则表现出了约 3.25%的性能优势.这表明了本文提出的方法更适用于大规模存储的环境,例如云镜像的集中存储环境.此外,在原型系统的区块链环境中,5 个参与结点仅用于完成 PoA 共识,将镜像的哈希信息记录在 Merkel 树中并在新生成的区块中存储,并不涉及代币的交易或智能合约的运行,同时也没有 PoW(Proof of Work)共识机制下的"挖矿"活动,因此不会产生区块链系统内部资源(如 gas)的消耗.

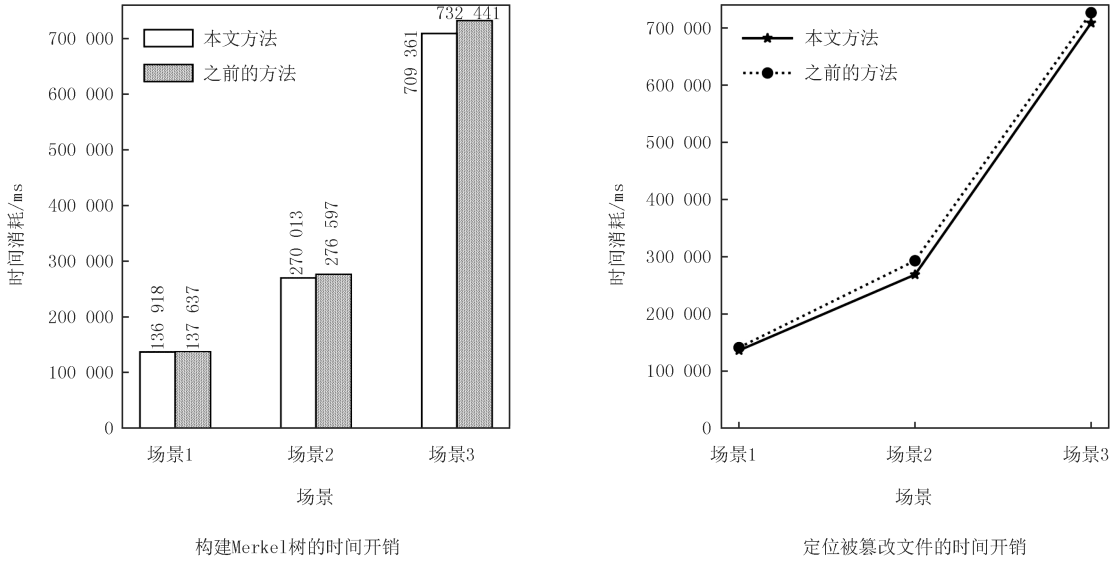

图5 实验中的时间开销

Fig.5 Time overhead in the experiments

3.3 讨论

下面从安全性、可追溯性和可扩展性三方面对本文提出的方法进行讨论.

安全性:本文提出方法的监测与检测过程都依赖于预先构建的 Merkel 树,该 Merkel 树存放在区块链数据结构中,而区块链防篡改的特性决定了这一 Merkel 树是无法被非法修改的.同时,方法可以通过调整 δ 的取值来规避 TOCTOU 攻击,确保无法在区块生成的间隙实施攻击.此外,根据 1.2 中定义的攻击者模型,攻击者不具有攻击区块链机制的知识和能力.因此,本文所提出方法的安全性是可以保证的.

可追溯性:本文提出的方法中,区块是依据同样的时间间隔逐一生成的,这种时序特性赋予了方法可追溯性.当检测到当前区块的完整性变化时,表明攻击行为发生在当前区块建立之前,前一个区块建立之后,因此能够通过完整性检测确定系统遭受攻击的时间.此外,镜像的完整性信息按照时间顺序被保存在区块链中,作为存证便于系统在遭受攻击后恢复被篡改的镜像.

可扩展性:区块链去中心化的特性使得本文提出的方法具有可扩展性.在真实的云环境中,云镜像往往采用分布式储存的方法进行部署.在这种情况下,本文的方法可以调整为:各个存储节点共同维护一个区块链,单个存储节点为自己维护的镜像文件构建 Merkel 树,获取区块链的记账权,生成新的区块并最终组成区块链;其中,对单个存储节点而言,其所维护的云镜像所对应的区块按照固定时间间隔生成,记作 δ_1;对于整个区块链而言,两个相邻区块的生成时间间隔也是固定值,记作 δ_2.通过调整 δ_1 和 δ_2 的取值,可以确保区块链有序地对所有存储节点中存储的镜像文件进行完整性存证,同时也支持新存储节点的动态加入和动态离开,即实现了扩展性.

4 结 论

本文研究了一种云镜像完整性监测方法,基于 Merkel 树技术和区块链技术对云镜像文件的完整性实施

动态监测和篡改检测.其中,借助于 Merkel 树的特性,本文提出的方法实现了高效率的完整性监测;区块链在本文提出的方法中,体现出了时序性、分布式和防篡改的特性,借助于这些特性,使得本文提出的方法是一种安全、可追溯及可扩展的方法.通过实验和分析,验证了本文提出的方法在效率上优于传统方法.在未来的工作中,将着力于解决方法在动态性方面尚存的不足.

参 考 文 献

[1] 罗军舟,金嘉晖,宋爱波,等.云计算:体系架构与关键技术[J].通信学报,2011,32(7):3-21.
LUO J Z,JIN J H,SONG A B,et al.Cloud computing:architecture and key technologies[J].Journal on Communications,2011,32(7):3-21.

[2] KUMAR R,GOYAL R.On cloud security requirements,threats,vulnerabilities and countermeasures:A survey[J].Computer Science Review,2019,33:1-48.

[3] 郭涛,刘菲军,杜垚,等.云计算环境下虚拟机部署策略的优化[J].计算机应用研究,2012,29(9):3425-3427.
GUO T,LIU F J,DU Y,et al.Virtual machine deployment optimization in cloud[J].Application Research of Computers,2012,29(9):3425-3427.

[4] MA W,LI X,SHI Y,et al.A virtual machine cloning approach based on trusted computing[J].Telkomnika Indonesian Journal of Electrical Engineering,2013,11(11):6935-6942.

[5] VRABLE M,MA J,CHEN J,et al.Scalability,fidelity,and containment in the potemkin virtual honeyfarm[C]//Proceedings of the twentieth ACM symposium on Operating systems principles.[s.l.:s.n.],2005:148-162.

[6] LAGAR-CAVILLA H A,WHITNEY J A,BRYANT R,et al.Snowflock:Virtual machine cloning as a first-class cloud primitive[J].ACM Transactions on Computer Systems(TOCS),2011,29(1):1-45.

[7] 李俊涛,吴小开.基于布朗指数法的虚拟机动态整合方法[J].计算机工程与应用,2016,52(7):56-61.
LI J T,WU X K.Dynamic consolidation of virtual machines based on Brown's exponential smoothing[J].Computer Engineering and Applications,2016,52(7):56-61.

[8] 郑婷婷,武延军,贺也平.云计算环境下的虚拟机快速克隆技术[J].计算机工程与应用,2011,47(13):63-67.
ZHENG T T,WU Y J,HE Y P.Rapid virtual machine cloning in cloud computing environment[J].Computer Engineering and Applications,2011,47(13):63-67.

[9] DU D,YU T Y,XIA Y B,et al.Catalyzer:sub-millisecond startup for serverless computing with initialization-less booting[C]//Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems.New York:ACM,2020:467-481.

[10] NICOLAE B,BRESNAHAN J,KEAHEY K,et al.Going back and forth:efficient multideployment and multisnapshotting on clouds[C]//Proceedings of the 20th international symposium on High performance distributed computing.New York:ACM,2011:147-158.

[11] USTIUGOV D,PETROV P,KOGIAS M,et al.Benchmarking,analysis,and optimization of serverless function snapshots[C]//Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems.New York:ACM,2021:559-572.

[12] 成淑萍,袁小艳,廖小平,等.基于云计算安全环境的僵尸网络传播模型[J].计算机与数字工程,2018,46(9):1830-1833.
CHENG S P,YUAN X Y,LIAO X P,et al.Botnet propagation modeling in cloud computing security environment[J].Computer & Digital Engineering,2018,46(9):1830-1833.

[13] Palo Alto Unit 42.20 Million Miners:Finding Malicious Crypto jacking Images in Docker Hub[EB/OL].[2022-03-26].https://unit42.paloaltonetworks.com/malicious-cryptojacking-images/.

[14] YAVUZ A A,NING P.BAF:an efficient publicly verifiable secure audit logging scheme for distributed systems[C]//2009 Annual Computer Security Applications Conference.[s.l.:s.n.],2010:219-228.

[15] YAVUZ A A,NING P,REITER M K.BAF and FI-BAF:efficient and publicly verifiable cryptographic schemes for secure logging in resource-constrained systems[J].ACM Transactions on Information and System Security,2012,15(2):1-28.

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

[17] XU Y,ZHANG C,WANG G J,et al.A blockchain-enabled deduplicatable data auditing mechanism for network storage services[J].IEEE Transactions on Emerging Topics in Computing,2021,9(3):1421-1432.

[18] 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 and Distributed Systems,2011,22(5):847-859.

(下转第 151 页)

Journal of Henan Normal University(Natural Science Edition),2022,50(3):150-156.

[17] SHIRLEY H F,SYAHIIRAH A A,MAWARNI H,et al.2D:4D ratio and autism spectrum disorder in brunei darussalam[J].Journal of Autism and Developmental Disorders,2021,51:4577-4586.

Study on finger length ratio of tennis players based on binary Logistic regression

Yan Limin

(Department of Physical Education, Henan Normal University, Xinxiang 453007, China)

Abstract: The second to fourth finger length ratio(D_2 : D_4)is thought to be related to diverse traits including athletic ability. To examine the relationship between D_2 : D_4 and sports ability in tennis players and college students. The length of the right hand fingers were measured in vivo to estimate D_2 : D_4. Tennis players and ordinary college students were used as the independent variable, and right-hand D_2 : D_4 ratios and gender were used as the dependent variable. All the statistics were computed with binary logistic regression model and tree model. Results showed that the D_2 : D_4 of tennis players were significantly lower than those of ordinary college students.

Keywords: tennis player; finger length ratio; Logistic regression

[责任编校　杨浦　刘洋]

(上接第 99 页)

[19] DILLIBABU M,KUMARI S,SARANYA T,et al.Assured protection & veracity for cloud data using merkle hash tree algorithm[J].Indian Journal of Applied Research,2011,3(3):124-126.

[20] YU M C,SAHRAEI S,LI S Z,et al.Coded merkle tree:solving data availability attacks in blockchains[C]//BONNEAU J,HENINGER N.International Conference on Financial Cryptography and Data Security.Cham:Springer,2020:114-134.

[21] 袁勇,王飞跃.区块链技术发展现状与展望[J].自动化学报,2016,42(4):481-494.
YUAN Y,WANG F Y.Blockchain:the state of the art and future trends[J].Acta Automatica Sinica,2016,42(4):481-494.

[22] 刘明达,陈左宁,拾以娟,等.区块链在数据安全领域的研究进展[J].计算机学报,2021,44(1):1-27.
LIU M D,CHEN Z N,SHI Y J,et al.Research progress of blockchain in data security[J].Chinese Journal of Computers,2021,44(1):1-27.

A blockchain based integrity monitoring method for cloud images

Ma Wei[1], Cao Liuyang[2], Jian Pengpeng[1]

(1. School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China;
2. School of Artificial Intelligence, Chongqing University of Technology, Chongqing 401135, China)

Abstract: Aiming at the problem of the image files being tampered with in cloud environment, an integrity monitoring method based on blockchain technology for cloud images is proposed. Firstly, hash values of all cloud images are calculated periodically and recorded in a Merkel Tree data structure. Next, the Merkel Tree is packed in a block and then a blockchain would be constructed. And finally, the root node of Merkel Tree in the current block is verified periodically to implement the overall verification for the integrity of all cloud images, and the tampered image file would be located when tampering attack is detected. According to experiments, the method proposed in this paper outperforms former methods for about 3%, and the theoretical analysis indicate that the method in this paper is secure, traceable, and expandable.

Keywords: cloud computing; integrity monitoring; blockchain

[责任编校　陈留院　赵晓华]