基于联盟链的指挥信息系统数据保护研究

陈传坤[1],谷立祥[2],颜廷贵[1]
(1. 北京宇航系统工程研究所,北京 100076;
2. 中国运载火箭技术研究院,北京 100076)

摘 要:随着社会信息化的快速发展,指挥信息系统逐渐成为作战的核心要素,所面临的数据安全问题也随之增加。传统的中心化数据保护方式易出现单点失效和数据篡改等问题,难以有效保障系统的数据安全。针对这些问题,结合区块链去中心化、不可篡改、可追溯审计以及可编程等优势,提出了一种基于联盟链的指挥信息系统数据保护方案,分析了指挥信息系统的节点功能,利用多通道结构对节点权限进行了划分,设计并实现了基于混合加密的数据保护智能合约,保证链上数据分布式存储、不可篡改、完整性可验证,支持数据的授权共享,并对非授权节点保密。测试结果显示,系统能够细粒度地对用户数据进行有效保护,系统响应延迟较低,节点容器内存开销较小,保证该系统能够在资源受限的平台上运行。

关键词:联盟链;指挥信息系统;智能合约;多通道;混合加密
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2021)10-0105-06
doi:10.3969/j.issn.1673-629X.2021.10.018

Research on Data Protection of Command Information System Based on Consortium Blockchain

CHEN Chuan-kun[1], GU Li-xiang[2], YAN Ting-gui[1]
(1. Beijing Institute of Astronautical Systems Engineering, Beijing 100076, China;
2. China Academy of Launch Vehicle Technology, Beijing 100076, China)

Abstract: With the rapid development of social informatization, the command information system has gradually become the core element of military operations. Meanwhile, the data security problems have also increased. The traditional centralized data protection method is prone to problems such as single point failure and data tampering, which cannot guarantee the data security of current system. In order to solve these problems, a command information system data protection method based on consortium blockchain is proposed, which utilizes the decentralized, immutable, auditable and programmable characteristics of blockchain. The node function of the command information system is analyzed, while system nodes are classified into different channels according to their data access permissions. Data protection smart contracts based on hybrid encryption are designed and implemented, which can be used to realize the characteristics of distributed storage, tamper-proof and verifiable integrity. The raw data can be shared within authorized nodes and disclosed to unauthorized nodes. The test shows that the proposed method can achieve fine-grained data protection with low system response delay and relatively small memory overhead of node containers, which illustrates that this system is able to run on the platform with limited hardware performance.

Key words: consortium blockchain; command information system; smart contract; multichannel; hybrid encryption

0 引 言

在信息化时代背景下,指挥信息系统正向网络中心化方向快速发展,作为指控指令、情报、作战平台状态等军事信息获取、传递、处理、分析的统一平台,指挥信息系统连接地理上分布的作战单元,将作战信息集中于网络,能够保障各作战单元间有效的数据交互与共享。但随着信息化程度的提高,其应用服务、数据、终端不断增加,面对当前日益复杂的网络攻击,指挥信息系统的数据安全问题日渐严峻。目前,指挥信息系统主要依靠中心化架构实现,一旦中心化服务器被攻击,系统将面临数据篡改和泄漏的风险,同时,中心化架构难以有效满足分布式、网络化作战的需求[1]。以防火墙、访问控制为代表的传统安全手段可控性较差,安全管理效率较低,难以有效保障系统的数据安全[2]。

收稿日期:2020-11-13 **修回日期**:2021-03-15
基金项目:中国航天科技集团钱学森青年创新基金(20170102)
作者简介:陈传坤(1996-),男,硕士研究生,研究方向为指挥信息系统数据安全;谷立祥,博士,研究员,研究方向为飞行器总体设计;颜廷贵,博士,研究员,研究方向为软件体系架构设计。

作为近期引起广泛关注的新兴技术,区块链技术具有去中心化、去信任化、集体维护、数据不可篡改和可追溯等优势[3],契合指挥信息系统对数据保护提出的数据可信、可审计追踪、分布式存储与验证等要求。然而,在区块链系统中,各参与方以对等节点的形式参与区块链维护,且都拥有全局账本,这使得数据保护无法利用传统的中心化方式解决,因此,如何将区块链的原生优势与数据保护进行有效结合是当前指挥信息系统面临的一大难题[4]。文中对当前基于区块链的数据保护方案进行了研究,分析了适用于指挥信息系统的区块链方案,设计并实现了基于混合加密的数据保护智能合约,搭建了基于联盟链的指挥信息系统进行方案验证,结果证明该方案能够实现细粒度的数据保护,并可以高效地进行机密数据的授权共享。

1 相关工作

数据保护的核心在于对数据进行完整性和机密性保护,得益于区块链在数据安全可信方面的原生优势,相关技术现已被广泛应用于多种数据保护场景中。目前,基于区块链的数据保护方案可分为以下四类。

(1)明文上链:交易数据明文记录在区块链上,利用区块只增不减、历史区块不可篡改的特性实现数据的完整性保护,该方案主要应用在以比特币(Bitcoin)[5]、以太坊(Ethereum)[6]为代表的区块链平台上。赵赫等人将该方案应用于采样机器人系统,保证数据记录的真实性、完整性[7];Enis 等人提出了基于区块链的空域指挥控制系统,以保障情报数据真实可信[8]。这种方案的缺点是链上明文数据对其他节点可见,无法对数据共享进行有效控制,进而无法保证数据的机密性,存在极大的数据泄露风险[9]。

(2)对称加密上链:数据在链下经对称加密后上传至区块链存储,加密密钥由节点自身或密钥管理中心存储。文献[10-12]中的区块链系统均采用本方案对隐私数据进行有效保护。该方案中,由于区块链仅存储密文,数据的完整性和机密性都能够得到有效保障,但数据共享需要在链下进行密钥传输,共享过程不透明,无法对该过程进行有效追溯和审计,因此,该方案依然存在数据泄露的风险。

(3)非对称加密上链:数据上传节点利用接收节点公钥对数据进行非对称加密后上传至区块链存储,文献[13-14]应用该方案对医疗数据进行隐私保护,以避免患者隐私被非法利用。该方案能够有效实现数据的完整性、机密性保护,数据共享结果可追溯,但其缺点是非对称加密算法加密效率低,与多个节点共享数据时需进行多次加密计算,系统性能难以保证。

(4)数据哈希上链:仅将数据哈希值存储在区块链上,原始数据以哈希值为索引加密存储在链下数据库中。目前,大多数针对文档的数据保护系统[15-16]采用此方案。该方案的优点是数据存储与管理分离,系统性能通常较高,但由于区块链不存储原始数据,无法充分发挥区块链的去中心化优势,一旦链下数据发生丢失或篡改,原始数据无法恢复。

2 系统设计

2.1 区块链选型

区块链技术是一种将数据以区块形式存储并链式连接的分布式数据库技术,以 P2P(peer-to-peer)网络作为底层通信载体,利用密码学保障数据隐私,并依靠分布式共识机制保障数据一致性[17]。目前,按照节点记账权的归属,区块链的应用模式可分为公有链、私有链和联盟链:

(1)公有链:完全去中心化,任何节点都拥有记账权,并能够自由进出区块链网络,利用 PoW(proof of work)等共识机制维护区块链的数据一致性,交易吞吐率最低(比特币约为 7 TPS)且交易确认时间长[18];

(2)私有链:仅运行于组织机构内部,新区块由特定节点生成,数据读写权限仅向内部节点开放,利用中心化程度较高的共识机制大幅缩短交易确认时间,提升交易吞吐率;

(3)联盟链:作为公有链和私有链的中间态,联盟链运行于多个组织之上,节点进入网络前需经过认证许可,通常采用拜占庭容错的共识机制,由多个成员共同参与共识过程,交易吞吐率可达 2 000 TPS 且交易确认时间可控制在毫秒级[19]。

作为军事作战的核心要素,指挥信息系统应保证系统内部存储和传输的数据安全可信,具体地,有以下的数据保护需求:(1)数据不可被恶意篡改;(2)数据来源可追溯;(3)支持数据的授权共享,对非授权节点隔离;(4)数据完整性可验证;(5)数据分布式存储;(6)数据处理效率高。综上,联盟链呈现出多中心化的特征,能够较好地平衡区块链的安全性和系统性能。而 Hyperledger Fabric[20]作为当前应用最为广泛的企业级联盟链,具有节点授权准入、实名身份认证、高性能共识机制等特点,结合可编程的数据保护智能合约,Fabric 能够有效满足指挥信息系统对数据安全保护的需求,因此,文中选择采用联盟链 Hyperledger Fabric 作为系统的区块链实现平台。

2.2 系统架构

基于联盟链的指挥信息系统整体架构如图 1 所示,由用户层、智能合约层、区块链层协同实现:用户层的各个组织节点参与联盟链的管理,调用智能合约接口与对应的合约交互;智能合约层利用其可编程特性,

对数据及其访问控制实施灵活的配置和管理;区块链层利用历史区块的不可篡改性,为加密数据的存储与共享提供支撑。

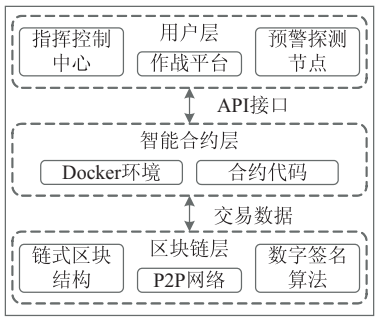

图 1　基于联盟链的指挥信息系统整体架构

为了便于分析,文中抽象出系统用户层模型。如图 2 所示,用户层由一个指挥控制中心、两个预警探测节点、两个次级指控中心和附属于次级指控中心的四个作战平台节点构成。其中,预警探测节点主要由预警雷达、预警机和侦查卫星构成,负责为指挥控制中心生成情报信息;指挥控制中心一般为固定指挥所,负责战场信息的接收、处理和指挥决策下达;次级指控中心通常由移动指挥车组成,主要任务是根据指挥控制中心下达的任务要求求解火控数据;作战平台节点由火力单元构成,负责依据火控数据实施具体的作战任务。两类指控中心和预警探测节点通常具有较强的计算、存储性能以及较高的带宽,能够作为联盟链的 Peer(对等)节点,存储全部区块链数据并响应客户端请求,其他节点仅作为客户端节点向 Peer 节点发送请求。

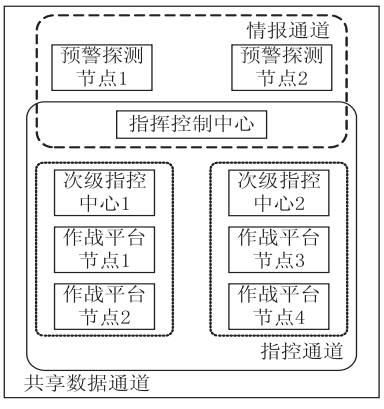

图 2　系统用户层模型

由于指挥信息系统的数据机密性较高,针对不同类型节点应提供不同的数据访问策略,文中建立了记录不同类型数据的三条独立通道:(1)情报通道,维护情报类数据,数据仅在指挥控制中心和预警探测节点内部共享;(2)指控通道,维护指控指令类、作战平台状态类数据,仅对指挥控制中心与次级指挥控制中心可见;(3)共享数据通道,记录所有节点共享的全局数据。每条通道维护一条独立的区块链,从而保证数据对通道内节点共享、对外隔离。

2.3　合约设计

作为数据保护方案的核心,可编程智能合约利用混合加密的方式支持数据的高效加密与授权共享,从而在通道相互独立的基础上,以单条数据的粒度保障机密性。文中的智能合约由加密密钥控制合约、数据共享与保护合约两部分构成。

2.3.1　加密密钥控制合约

加密密钥控制合约主要负责用户非对称加密密钥的注册与控制,包括密钥申请模块、密钥更改模块和密钥撤回模块,该合约仅由指挥控制中心节点审核执行,以保证指挥控制中心作为最高权限机构的权威性。非对称加密采用 NTRU 算法[21],相较于传统的 ECC 和 RSA 算法,NTRU 算法具有计算量小、硬件资源要求低、安全强度高的特点,加解密速度可达 ECC 和 RSA 算法的 20 倍以上[22]。

密钥申请模块为用户生成全局唯一的身份标识以及对应的 NTRU 密钥对<pubKey,priKey>,节点调用该合约模块时,指挥控制中心对节点信息进行审核,验证通过后将公钥 pubKey 存储于区块链上,并利用安全信道将私钥 priKey 传输给用户秘密存储。一旦私钥遭到黑客或内部人员非法窃取,区块链存储的加密数据可能被泄漏、篡改,为了避免私钥泄漏导致的安全性风险,需要进行密钥重置。密钥更换模块负责为原用户重新生成密钥对,将新的公钥数据写入区块链,并将此前与该用户相关的数据重新加密,从而保证历史数据的可用性。密钥撤回模块仅负责删除即将退出通道的用户密钥,从而防止黑客利用过期密钥获取相关数据。

2.3.2　数据共享与保护合约

数据共享与保护合约包括数据共享存储模块和数据获取模块。其中,数据共享存储模块流程如算法 1 所示,用户将原始数据 data 和接收节点列表 List<recId>打包为请求后签名,并通过安全信道发送至智能合约处理,合约模块验证客户端节点证书并获取节点标识 ownId,计算原始数据的哈希值 hash,为数据生成唯一标识 dataId,并生成随机对称密钥 key,根据接收节点标识 recId 从区块链中获取各节点加密公钥 pubKey,用各节点公钥对 key 进行非对称加密,并以 dataId 和 recId 为组合索引上传至区块链存储。最后,智能合约利用 key 对数据进行对称加密得到密文 cipher,并将 ownId、cipher、hash 以及 List<recId>等数据共同组成 dataJson 对象,以 dataId 为索引上传至区块链,从而完成数据共享存储模块的全部流程。

算法 1:dataUpload(data,List<recId>)

1:ownId=getCreator()

2:hash=sha256(data)

```
3:dataId = getDataId( )
4:key = AES. generateKey( )
5:for recId:List<recId>
6:pubKey = getState( composUser( recId) )
7:if pubKey = = nil
8:return userNotExistError( )
9:end if
10:encryptKey = NTRUencrypt( key,pubKey)
11:putState( composKey( dataId,recId) ,encryptKey)
12:end for
13:cipher = AESencrypt( data,key)
14:dataJson = jsonify( ownId,cipher,hash,List<recId>)
15:putState( composData( dataId) ,dataJSON)
16:return dataId
```

数据获取流程如算法 2 所示,客户端节点将包含目标数据标识的请求签名后发送至智能合约处理,在验证节点证书通过后,该模块获取节点标识 targetId,并在区块链中查询得到数据标识 dataId 对应的数据对象 dataJSON,验证请求节点标识是否在数据对象的接收节点列表中,若为否,则返回拒绝访问错误。最后,根据 dataId 和 targetId 获取加密后的对称密钥,并将密文 cipher、被公钥加密的密钥 encryKey 以及原始数据哈希 hash 返回请求节点。

```
算法 2:dataAccess( dataId)
1: targetId = getCreator( )
2:dataJSON = getState( composData( dataId) )
3:if dataJSON = = nil
4:return dataNotFoundError( )
5:end if
6: parse( dataJSON,ownId,cipher,hash,List<recId>)
7:if validateAuthority( List<recId>,targetId) = = false
8:return PermissionDeniedError( )
9:end if
10:encryKey = getState( composKey( dataId, targetId) )
11:return cipher,encryKey,hash
```

客户端节点收到数据获取模块返回的数据后,执行算法 3 对数据解密并进行完整性验证,客户端节点利用本地存储的私钥 priKey 对 encryKey 解密后得到对称密钥 key,利用对称解密算法解密 cipher 得到 plainText 并计算其哈希值 dataHash,若与区块链中记录的哈希值 hash 不一致,则说明数据曾被篡改过,报告哈希验证错误,否则,返回正常的解密数据。

```
算法 3::decryptAndVerify( cipher,encryKey,hash)
1: priKey = NTRU. getPrivateKey( )
2:key = NTRUdeceypt( encryKey,priKey)
3:plainText = AESdecrypt( cipher,key)
4:dataHash = sha256( plainText)
5:if dataHash! = hash
6: return hashVerificationError( )
7:end if
8:return plainText
```

与现有的数据保护方案相比,文中基于混合加密的数据保护方案有以下优势:(1)全部数据加密上链,充分发挥区块链分布式、不可篡改的优势;(2)充分利用对称加密速度快与非对称加密算法安全强度大的特点,满足指挥信息系统在数据处理速度方面的需求;(3)每条数据对应一个对称密钥,可有效防止数据大规模泄露;(4)数据共享信息存储于区块链上,从而保证信息透明、可审计追溯;(5)指挥控制中心负责管理所有节点的加密密钥,从而保证其作为指挥控制中枢的权威。

3 系统实现

系统方案实现由三部分构成:Fabric 网络环境搭建、多通道配置和智能合约实现。下面将介绍各部分的具体实现。

3.1 网络环境搭建

Fabric 网络提供了联盟区块链系统的核心服务:(1)区块链服务,维护各通道对应的区块链,并根据区块链的历史交易数据更新最新数据状态;(2)智能合约服务,为智能合约运行提供安全、隔离且轻量化的 Docker 运行环境;(3)成员管理服务,利用公钥基础设施实现节点的身份证书创建、管理以及交易的签名验证;(4)共识服务,接收并验证各通道的交易请求,并将其打包为区块返回至各个通道。在以上基础服务之上,文中根据图 2 用户层模型搭建 Fabric 网络环境,编写 YAML 配置文件定义各个组织的属性,具体属性如表 1 所示。

表 1 指挥信息系统组织信息

组织名称	组织标识	组织 ID	客户端节点数
指挥控制中心	OrgCC	OrgCCMSP	1
次级指控中心 1	OrgSC1	OrgSC1MSP	3
次级指控中心 2	OrgSC2	OrgSC2MSP	3
预警探测节点 1	OrgInfo1	OrgInfo1MSP	1
预警探测节点 2	OrgInfo2	OrgInfo2MSP	1

各组织生成一个 Peer 节点参与区块链管理并存储全部区块数据,同时,每个组织包含一个代表自身的客户端节点用于调用智能合约,作战平台节点受限于硬件资源,仅作为次级指控中心的客户端节点参与到网络中。最后,利用 Fabric 编译工具 cryptogen 和 configtxgen 生成各组织节点的证书文件、系统的创世区块以及通道配置区块等信息。

3.2 多通道配置

为了对机密数据实行访问控制,利用配置文件将

系统组织划分为三个相互独立的通道,各通道配置如表 2 所示。其中,情报通道包含指挥控制中心和两个预警探测节点,情报数据仅在这三个组织内流通,次级指挥控制中心没有权限也无需存储情报通道数据,从而降低存储和带宽的需求;同理,指控通道包含指挥控制中心和两个次级指挥控制中心;共享数据通道包含所有组织以实现全局通用数据的共享。每个通道内均有多个组织对数据进行分布式存储,从而保障系统有较强的抗毁伤能力,有效避免单点失效问题。

表 2　通道配置信息

通道名称	通道标识	通道内组织
情报通道	InfoChannel	OrgCC, OrgInfo1, OrgInfo2
指控通道	CommandChannel	OrgCC, OrgSC1, OrgSC2
共享数据通道	SharedChannel	全部

3.3　智能合约开发

文中使用 Go 语言实现上述智能合约,在完成通道配置后,任一客户端节点可调用智能合约的 instantiate 方法在所属通道内对合约进行实例化,并调用 Init 方法初始化合约。在完成上述过程后,区块链网络搭建完成,客户端节点可调用 Invoke 方法执行对应的智能合约。

4　结果与分析

本系统的环境搭建工作在 VMware 虚拟机内完成,虚拟机的配置如下:操作系统为 Ubuntu16.04,处理器数量为 1,内存为 2 GB,硬盘容量为 20 GB。

4.1　系统交互

下面以情报通道为例,对智能合约的主要功能进行测试。预警探测节点 OrgInfo1 调用密钥申请模块注册加密密钥,结果如图 3 所示。

```
cck@ubuntu: ~/go/src/github.com/hyperledger/nodeTest
cck@ubuntu:~/go/src/github.com/hyperledger/nodeTest$ sudo node invoke.js
Successfully sent Proposal and received ProposalResponse:
Status - "200"
result: User Registration Success
```

图 3　密钥申请模块调用结果

预警探测节点 OrgInfo1 发现敌方目标后,将目标类型、坐标、速度等信息封装为合约参数,调用数据共享存储模块,并将数据设置为与 OrgCC 共享,调用结果如图 4 所示。通道内的 OrgCC 节点调用数据获取模块并对结果进行解密,可获取相关情报数据,最终调用结果如图 5 所示。

```
cck@ubuntu: ~/go/src/github.com/hyperledger/nodeTest
cck@ubuntu:~/go/src/github.com/hyperledger/nodeTest$ sudo node invoke.js
Successfully sent Proposal and received ProposalResponse:
Status - "200"
result: Data Upload Success
data ID: { 0105 }
```

图 4　数据共享存储模块调用结果

```
cck@ubuntu: ~/go/src/github.com/hyperledger/nodeTest
cck@ubuntu:~/go/src/github.com/hyperledger/nodeTest$ sudo node query.js
returned from query
Response:  "target: {Trident} current time: {2020.06.09 12:09:11} longitude: {179.17} latitude: {46.47} velocity: {3152.07}"
```

图 5　数据获取存储模块调用成功

通道内的 OrgInfo2 节点不在该数据的共享节点列表中,因此,没有该数据的访问权限,调用数据获取模块会导致失败,其结果如图 6 所示。

```
cck@ubuntu: ~/go/src/github.com/hyperledger/nodeTest
cck@ubuntu:~/go/src/github.com/hyperledger/nodeTest$ sudo node query.js
returned from query
Response:  Error: data can not be shared to you
```

图 6　数据获取存储模块调用失败

4.2　性能分析

由于加密密钥控制合约调用次数较少,因此,以下性能分析主要集中于数据共享与保护合约上。图 7 展示了系统执行多次数据共享存储和数据获取请求的响应延迟。整体上,共享存储请求的响应延迟高于数据获取请求的延迟,这主要是因为执行共享存储请求时会生成用于存储数据的新区块,区块的生成以及在各节点间的同步需消耗一定的时间,而数据获取请求仅需在 Peer 节点中执行查询操作,不存在生成新区块的时间开销,响应更快。对于共享存储请求,系统的响应延迟稳定在 0.15 s 至 0.25 s,平均响应时间为 0.198 s;数据获取请求的响应延迟在 0.1 s 至 0.15 s 间波动,平均响应延迟为 0.126 s,数据写入和读取请求都能够得到较快的系统响应。因此,系统中各节点能够快速地将战场信息上传至区块链,通道内的其他节点可以及时跟踪所需信息,保障战场环境下的数据安全可信。

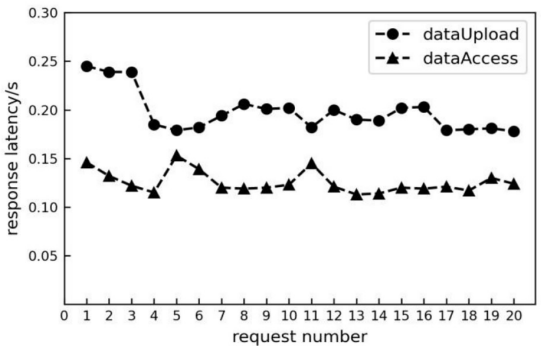

图 7　区块链请求响应延迟

指挥信息系统的部分节点硬件资源性能受限,因此,文中测试了各节点以及链码的 Docker 容器内存开销,在各通道执行 1 000 次数据存储合约调用后,具体的内存占用情况如表 3 所示。其中,所有 Peer 节点容器所占用的内存均小于 50 MB,客户端容器、链码容器的内存占用分别小于 2 MB 和 8 MB,因此,本系统可运行在无人机等小型便携计算设备上,从而进一步拓展信息广度和维度,保障战场的多域信息协同。

表 3　节点容器内存占用情况

节点名称	Docker 容器名称	占用空间/MB
预警探测节点 1	peer0.orgInfo1	40.980
预警探测节点 2	peer0.orgInfo2	39.880
指挥控制中心	peer0.orgCC	41.180

续表 3

节点名称	Docker 容器名称	占用空间/MB
次级指控中心 1	peer0. orgSC1	9. 195
次级指控中心 2	peer0. orgSC2	41. 240
预警探测节点 1 客户端	cli. orgInfo1	1. 484
预警探测节点 2 客户端	cli. orgInfo2	1. 508
指挥控制中心客户端	cli. orgCC	1. 461
次级指控中心 1 客户端	cli. orgSC1	1. 660
次级指控中心 2 客户端	cli. orgSC2	1. 477
作战平台节点 1 客户端	cliBattle1. orgSC1	1. 473
作战平台节点 2 客户端	cliBattle2. orgSC1	1. 457
作战平台节点 3 客户端	cliBattle3. orgSC2	1. 469
作战平台节点 4 客户端	cliBattle4. orgSC2	1. 453
智能合约容器	comctrlchaincode	7. 738

5 结束语

针对目前指挥信息系统存在的数据安全难以有效保障的问题,提出了基于联盟链的指挥信息系统数据保护方案。方案在 Fabric 的基础上,搭建了小型指挥信息系统网络,依靠 Fabric 节点授权准入、通道相互隔离等特性,极大地避免了非授权节点的入侵。同时,为了对数据安全进行细粒度控制,开发了基于混合加密的数据保护智能合约,数据与加密密钥一一对应,防止数据大规模泄漏,且数据授权共享结果可追溯审计,有效地提高了数据的安全性。测试结果显示,系统对请求的响应延迟在 0. 25 s 内;客户端节点和 Peer 节点所占用内存空间分别小于 2 MB 和 50 MB,说明本系统可运行在性能受限的小型计算设备上,有利于信息的多域协同。综上,区块链技术与指挥信息系统的结合,能够保障数据的机密性、完整性,满足指挥信息系统数据安全保护的需求。文中是区块链应用于指挥信息系统上的初步探索,下一步将继续完善区块链在该领域的工程应用,提升系统的扩展性,为指挥信息系统提供更高效、安全的数据保护平台。

参考文献:

[1] 蒋晓原,邓克波. 面向未来信息化作战的指挥信息系统需求[J]. 指挥信息系统与技术,2016,7(4):1-5.

[2] 金 朝,杨 文,李英华,等. 指挥信息系统信息网络安全控制研究[J]. 火力与指挥控制,2014,39(11):97-100.

[3] 邵奇峰,金澈清,张 召,等. 区块链技术:架构及进展[J]. 计算机学报,2018,41(5):969-988.

[4] 刘敖迪,杜学绘,王 娜,等. 区块链技术及其在信息安全领域的研究进展[J]. 软件学报,2018,29(7):2092-2115.

[5] NAKAMOTO S. Bitcoin:a peer-to-peer electronic cash system[EB/OL]. (2008-08-21)[2020-10-15]. https://bitcoin. org/bitcoin. pdf.

[6] BUTERIN V. A next-generation smart contract and decentralized application platform[EB/OL]. (2014-08-10)[2020-10-15]. https://github. com/ethereum/wiki/wiki/White-Paper.

[7] 赵 赫,李晓风,占礼葵,等. 基于区块链技术的采样机器人数据保护方法[J]. 华中科技大学学报:自然科学版,2015,43(S1):216-219.

[8] KONACAKLI E,KARAARSLAN E. Blockchain-based secure recognized air picture system proposal for NATO air C2 capabilities[C]//International conference on artificial intelligence and applied mathematics in engineering. [s. l.]:Springer,2019:758-765.

[9] 许重建,李险峰. 区块链交易数据隐私保护方法[J]. 计算机科学,2020,47(3):281-286.

[10] 巫岱玥,余 祥,王 超,等. 基于区块链的信息系统数据保护技术研究[J]. 指挥与控制学报,2018,4(3):183-188.

[11] 董 蓉,苑明海,周 灼. 基于区块链的云制造信息数据记录技术[J]. 计算机技术与发展,2019,29(5):97-101.

[12] 邢少敏,冯 维,王泉景. 基于区块链技术的涉密电子文档保护方案研究[J]. 信息安全研究,2017,3(10):884-892.

[13] 薛腾飞,傅群超,王 枞,等. 基于区块链的医疗数据共享模型研究[J]. 自动化学报,2017,43(9):1555-1562.

[14] XIA Q I,SIFAH E B,ASAMOAH K O,et al. MeDShare:trust-less medical data sharing among cloud service providers via blockchain[J]. IEEE Access,2017,5:14757-14767.

[15] 谭海波,周 桐,赵 赫,等. 基于区块链的档案数据保护与共享方法[J]. 软件学报,2019,30(9):2620-2635.

[16] ZHANG J,XUE N,HUANG X. A secure system for pervasive social network-based healthcare[J]. IEEE Access,2016,4:9239-9250.

[17] 邵奇峰,张 召,朱燕超,等. 企业级区块链技术综述[J]. 软件学报,2019,30(9):2569-2592.

[18] ZHENG Z,XIE S,DAI H,et al. An overview of blockchain technology:architecture,consensus,and future trends[C]//2017 IEEE international congress on big data (BigData congress). New York:IEEE,2017:557-564.

[19] 刘懿中,刘建伟,张宗洋,等. 区块链共识机制研究综述[J]. 密码学报,2019,6(4):395-432.

[20] ANDROULAKI E,BARGER A,BORTNIKOV V,et al. Hyperledger fabric:a distributed operating system for permissioned blockchains[C]//Proceedings of the thirteenth eurosys conference. Porto,Portugal:ACM,2018:1-15.

[21] HOFFSTEIN J,PIPHER J,SILVERMAN J H. NTRU:a ring-based public key cryptosystem[C]//International algorithmic number theory symposium. Portland,Oregon,USA:Springer,1998:267-288.

[22] GAITHURU J N,BAKHTIARI M. Insight into the operation of NTRU and a comparative study of NTRU,RSA and ECC public key cryptosystems[C]//2014 8th. Malaysian software engineering conference (MySEC). New York:IEEE,2014:273-278.