

分布式网络环境下基于区块链的密钥管理方案

戴千一¹, 徐开勇¹, 郭松¹, 蔡国明¹, 周致成^{1,2}

(1. 战略支援部队信息工程大学密码工程学院, 河南 郑州 450001;
2. 61660 部队, 北京 100089)

摘要: 为解决在分布式网络实施密钥管理困难和通信开销过大等问题, 提出一种基于区块链技术的密钥管理方案(KMSBoB, key management schemes based on Blockchain)。KMSBoB 中设计了分布式群组网络下基于区块链的密钥管理和传输过程, 并将区块链全体成员挖矿过程和 MTI/CO 协议过程相结合形成动态生成会话密钥生成协议, 简化了跨异构自治域下的密钥管理策略。通过仿真测试和结果分析表明 KMSBoB 的安全性和有效性, 并且相比传统密钥管理方案通信开销更少和扩展性更高。

关键词: 区块链; 密钥管理; 分布式网络

中图分类号: TP399

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2018077

Blockchain-based key management scheme for distributed networks

DAI Qianyi¹, XU Kaiyong¹, GUO Song¹, CAI Guoming¹, ZHOU Zhicheng^{1,2}

1. School of Cryptographic Engineering, Strategic Support Force Information Engineering University, Zhengzhou 450001, China
2. 61660 Troops, Beijing 100089, China

Abstract: Regarding the difficult key management and excessive communication overhead in the distributed group networks, KMSBoB-Key management schemes based on blockchain was proposed, featuring key management and transmitting procedure based on blockchain in distributed group networks. Meanwhile, the all-member mining process of blockchain and the MTI/CO protocol process were combined to establish a protocol of dynamically generated session key, so as to streamline key management strategies in cross-heterogeneous autonomous domains. According to simulation testing and result analysis, KMSBoB is safe and effective, with less communication overhead and higher expansibility than traditional key management schemes.

Key words: Blockchain, key management, distributed networks

1 引言

随着移动互联网技术的快速发展, 移动终端设备数量激增。移动互联网设备所采用的分布式自组织网络模式具有拓扑动态性、无中心分布性、

节点成员身份状态高度的动态性等特点, 使分布式网络环境下密钥管理较为复杂, 成为相应领域主要研究热点。

密钥管理主要为群组全体合法成员提供密钥生成、密钥分发以及密钥更新等服务。目前, 分

收稿日期: 2018-08-10; 修回日期: 2018-09-05

通信作者: 戴千一, qianyi.dai@outlook.com

基金项目: 国家重点研发计划基金资助项目 (No.2017YFB080210)

Foundation Item: The National Key Research and Development Program of China (No.2017YFB080210)

布式网络中进行密钥管理方案有集中式和分布式两类。集中式密钥管理方案^[1-2]由于其结构高度依赖中心化(KMC, key management center), 导致其存在易发生单点故障、跨域密钥传输时延较大、使用效率不高和密钥管理较为复杂等问题, 难以适应动态变化的分布式网络环境。针对分布式的密钥管理协议, 近年来已有许多相关研究, 其中, 常见的分布式密钥协商协议有 BD(burmeser-desmedt) 协议^[2]、STR(steer, et al) 协议、TGDH(tree-based group diffie-Hellman) 协议^[3]等。BD 协议通信开销较大, 但计算开销较少。TGDH 协议的扩展计算量和通信量都比较小, 但密钥存储开销大。文献[4]提出了成员匿名的无线网络组播建立协议, 在不泄露接收者属性信息的情况下完成密钥传输, 但文献[4]协议缺乏身份验证和密钥确认机制, 无法提供前向保密性和后向保密性。文献[5-6]设计了移动环境下高效的密钥交换握手协议, 但其管理机制较为复杂, 不适合动态群组网络。上述协议方案在群组成员动态变化的动态群组通信中的应用效果不佳, 因此需要设计一种在动态分布式网络环境下的无中心的密钥管理方案, 适应动态群组通信系统。

区块链最早是由中本聪提出比特币的基础支撑技术^[7], 实现在开放的对等网络环境下, 通过透明可信的智能合约, 构建不可篡改和可追溯的块链式数据结构, 在多个站点或机构组成的网络中实现全体成员对数据共享、审计、管理和集体维护。待传输数据被封装入区块并广播到网络中以供全体成员进行集体挖矿过程, 实现区块数据全网同步。如果消息通过认证, 则在本地账本中附加新区块。通过多方备份维护区块数据可实现将待传递数据直接发送到目的地成员, 而不是通过中央管理器传递数据, 可以加快跨域信息传播, 并且可以保证传输数据的完整性和可靠性。基于区块链网络的分布式结构在应对单点故障方面具有更好的顽健性, 因此, 本文采取区块链作为解决密钥管理的技术方案。

为了实现分布式网络的可靠密钥传输, 提高群组密钥生成和更新的效率, 简化密钥管理策略, 降低计算和通信开销, 本文提出了 KMSBoB, 实现密钥动态安全管理, 满足动态群组网络下密钥

管理需求, 主要贡献在以下两方面。

1) 设计了分布式网络环境下基于区块链的密钥管理和策略。

2) 将 MTI/CO 协议与区块链多节点挖矿相结合, 形成一种分布式环境下多节点会话密钥生成协议。

2 相关知识和说明

2.1 MTI/CO 密钥协商过程

本文设计的算法中, 采用了 MTI/CO 协议, 协议过程如下。

1) 成员 A 和成员 B 生成各自公开密钥 $pk_A = g^a, pk_B = g^b$, 并全网公开。

2) 成员 A 生成随机数 r , 计算 $k_A = g^{br}$, 发送给成员 B。

3) 成员 B 生成随机数 s , 计算 $k_B = g^{as}$, 发送给成员 A。

4) 成员 A 计算会话密钥 $k_1 = k_B^{a^{-1}r} = g^{asa^{-1}r} = g^{rs}$, 成员 B 计算会话密钥 $k_2 = k_A^{b^{-1}s} = g^{brb^{-1}s} = g^{rs}$ 。

5) 通过计算 $k = k_1 = k_2 = g^{rs}$, 实现会话密钥安全共享。

2.2 双线性映射

G_1 、 G_2 和 G_T 为 3 阶且是素数 q 的乘法循环群, g_1 、 g_2 、 g_3 为 3 个循环群的生成元。令 $e: G_1 \times G_2 \rightarrow G_T$, 映射具有以下性质。

1) 双线性。对于所有 $g_1 \in G_1, g_2 \in G_2, a, b \in \mathbb{Z}_q$, 有 $e(g_1^a, g_2^b) = e(g_1, g_2)ab$ 。

2) 非退化性。存在 $g_1 \in G_1, g_2 \in G_2$, 满足 $e(g_1, g_2) \neq 1$ 。即对 G_1, G_2 中的任何元素, 通过这个映射都不会得到 G_T 中的单位元。

3) 可计算性。对于任意 $g_1 \in G_1, g_2 \in G_2$, 存在有效的算法计算 $e(g_1, g_2)$ 。

2.3 密钥管理协议的安全属性

因为密钥管理协议是在公开的网络中由所有合法成员基于共识协商的密钥管理协议共同生成的, 所以一个安全的密钥管理方案需要满足如下基本安全属性。

1) 已知密钥安全性。由于每次生成的会话密钥中都含有每个组员根据相应算法产生的分量, 共同合成为会话密钥。所以每次生成的会话密钥都是唯一的, 即使以前使用过的会话密钥发生了泄露, 敌手也不可能知道当前全体成员使用的会

话密钥,并且新加入成员无法使用旧的会话密钥。

2) 前向保密性。即使部分成员的私钥泄露,也不会影响旧会话密钥的安全。

3) 密钥不可控制性。由于会话密钥是全体组员共同协商产生的,其中包含随机生成的值,因此群组中任何成员都不能预先生成和控制会话密钥生成其选定的值。

4) 抗泄露攻击。假设某个成员 Node_A 的私钥泄露,敌手能够假扮该节点欺骗其他成员,但是敌手不能扮演其他成员参与者欺骗 Node_A 。

5) 密钥共享可知性。协议必须在所有被网络中确认合法的成员参与的情况下进行密钥协商协议,任意合法成员未参与协议过程的情况下都无法进行密钥协商协议,并且全体合法成员可以验证会话密钥的正确性和合法性。

2.4 相关符号说明

为了更加清晰地说明算法过程,特做以下算法说明,如表1所示。

表1 相关符号说明	
符号	说明
SMN-A	安全管理节点 A
Src	源节点
Dest	目的节点
Address	安全管理节点 A 网络地址编号
ID	网络节点身份标识,本文模型为 URL 标识
Cert	数字证书,本文默认证书格式为 X.509 格式 ^[8]
$pk_{\text{SMN-A}}$	安全管理节点 A 的公钥,公钥对全体 SMN 成员公开
$sk_{\text{SMN-A}}$	安全管理节点 A 的私钥,本文默认私钥是安全的
$\text{enc}(M)_{pk} \rightarrow C$	加密算法,使用公钥 pk 加密明文 M 可得到密文 C
$\text{dec}(C)_{sk} \rightarrow M$	解密算法,使用私钥 sk 解密密文 C 可得到明文 M
$\text{sig}(M)_{sk} \rightarrow C$	签名算法,使用私钥 sk 对明文 M 签名可得到签名 C
$\text{ver}(C)_{pk} \rightarrow M$	验签算法,使用公钥 pk 对签名 C 验签可得到明文 M
hash()	杂凑算法,本文指 SHA1 算法
doublehash()	$\text{doublehash}() = \text{SHA256}(\text{SHA256}())$
Rand()	随机数生成算法
randPerm()	随机排序算法,对特定序列进行排序
g	双线性对生成元
G	乘法循环群
BlockHead	区块头
BlockBody	区块体
Block	区块,包含 BlockHead 和 BlockBody

特别地, Enc 、 $\text{Dec}()$ 、 $\text{Sig}()$ 和 $\text{Ver}()$ 这 4 个算法选取 ECDSA 的 secp256k1 椭圆曲线。

3 方案设计

3.1 系统模型

为了实现方案设计目标,本文分析并改进传统密钥管理模型缺陷,提出基于区块链的密钥管理模型,使网络群组成员之间可以高效简单地进行密钥管理和传输。

传统密钥管理模型严格遵循层次结构^[1-2],为典型的分级管理网络拓扑结构,其中服务层共有 3 层,用户节点 (user node) 则占一层,共同构成网络自治域 (AD, autonomous domain)。服务提供方各层包括中央管理节点 (CMN, central manager node)、安全管理节点 (SMN) 和网络接入点 (AP, access point)。CMN 的逻辑覆盖区域为 AD, CMN 负责 AD 内的节点身份注册、生成证书和验证身份,并负责多个 AD 内的 SMN 密钥管理和授权,其通常是 PKI 的一级 KMC^[8]; SMN 在 AD 的密码资料的第二层, SMN 的逻辑覆盖区域为安全域 (SD, secure domain),其通常是 PKI 的二级 KMC; AP 提供了一个网络接口将 UN 的消息路由到 SMN。全体 UN 构成了一个由纯对等非结构化模型组织的逻辑层,彼此间地位对等。

这种网络结构在顶层使用 CMN 管理 AD 内相关密钥数据,这样的网络结构高度依赖 CMN 的安全性,容易产生单点故障,并且如果密钥信息从一个 AD 转移到另一个 AD,则需要额外的握手协议进行密钥交换,不仅增加了通信开销,同时增加了密钥泄露的风险。

因此,为应对传统密钥管理模型带来的单点故障、跨域密钥传输时延较大、使用效率不高和密钥管理较为复杂等问题,本文考虑基于区块链的运行模式下,将验证密钥分量合法性,密钥分量传输和密钥分量共享的 3 个过程统一于全体 SMN 节点挖矿过程,通过全体 SMN 成员验证区块数据合法性的过程,实现密钥数据的动态安全管理,降低了传统自治域之间因密钥传输的通信开销和密钥泄露的安全风险。如图 1 所示,与传统结构最大的不同在于,顶层 CMN 的职能以基于区块链的结构合并到全体 SMN 成员组成的网络群组中。为了应对隐私安全,CMN 可以在以下

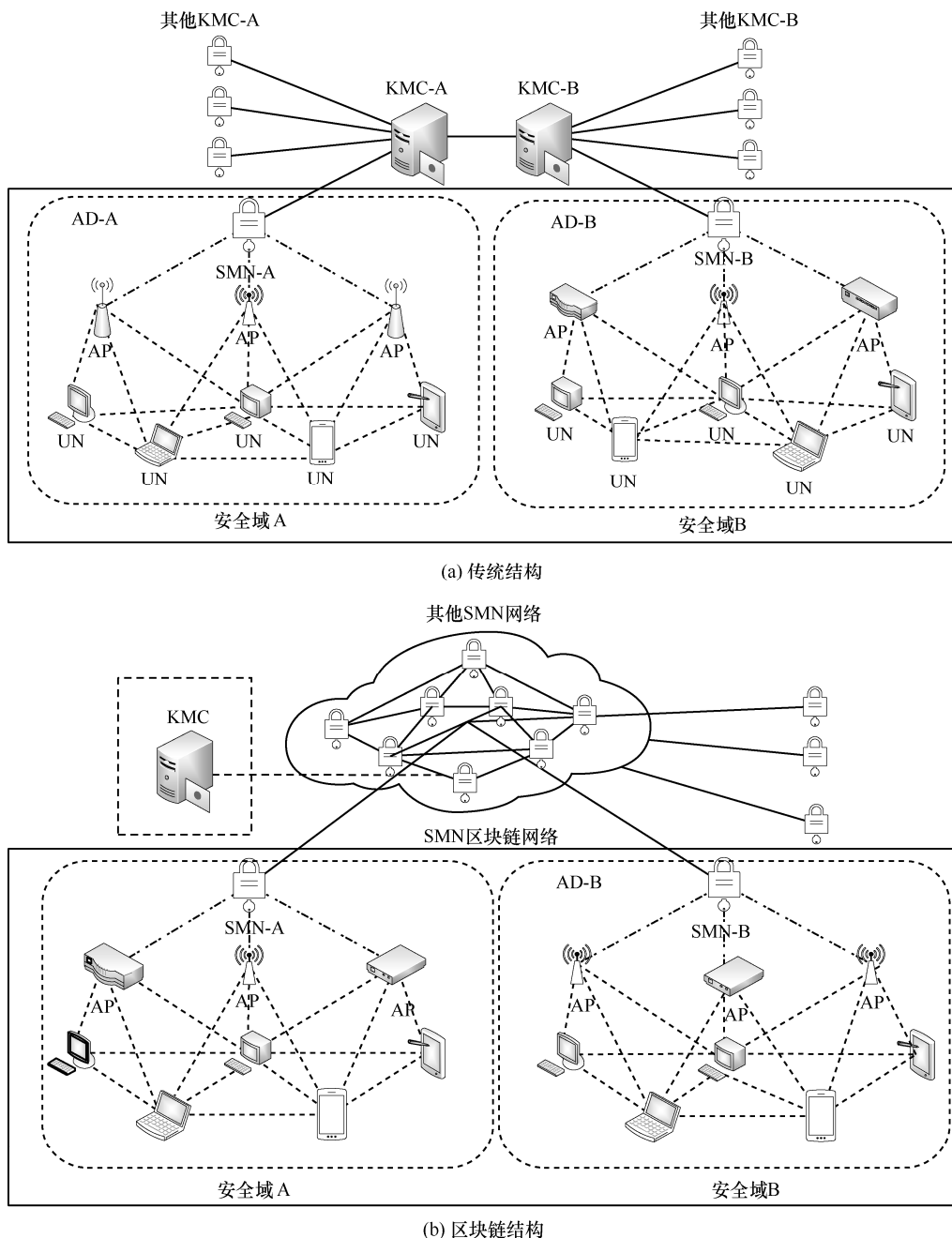


图 1 传统密钥管理结构和 KMSBoB 结构

3 种情况下为区块链网络提供安全服务。① 初始注册。新成员首次加入新的 AD 时需要 KMC 为其注册身份信息和密钥。② 更改身份相关信息。SMN 必须定期更改身份信息以及与身份相关的所有密钥资料, 因此需要 CMN 为其生成新的身份和密钥。③ 恶意节点身份撤销。在基于区块链的密钥管理结构中, KMC 可以作为独立审计方, 通过检索区块链识别成员恶意行为, 一旦恶意行为得到确认, 恶意节点的身份将被公布并撤销身

份及其所属密钥。

区块链可以在无中心节点参与维护的情况下, 通过全体成员的共识机制完成共享信息。SMN 群组之间可以动态互联, 也可以与其他有实体认证的自治域相链接, 通过屏蔽结构差异形成类似动态的 SMN 云结构。如果有多域之间的密钥生成或密钥更新的事务请求, SMN 则会将事务请求及相关信息封装到区块中, 并在网络信标中标明是否为跨域密钥传输。SMN 既是事务代理处

理方,同时也是矿工。本文的设计思路是通过全体节点挖矿来存储区块并验证区块的合法性,实现多方维护区块数据和密钥分量安全完整地传输。最终实现在多个 SMN 域内,通过全体 SMN 节点维护区块链数据实现在异构网络环境中达成密钥管理目的。基于以上过程,无论新成员加入网络还是密钥动态更新的过程,全体 SMN 都可以通过从区块链中检索相应信息并解密恢复区块内容来实现密钥传输和生成。特别说明的是,本文所指交易是相关密钥管理或传输事务信息。

3.2 分布式网络环境下密钥管理策略

本节介绍分布式网络环境下基于区块链密钥管理策略,旨在简化大型异构动态群组网络中各安全域的分布式密钥管理步骤,实现轻量级和可扩展的密钥传输方案。

3.2.1 密钥传输握手过程

传统网络^[5-6]中的跨域握手过程如图2所示。SMN 之间的握手信息由 KMC 强制进行交易检查,SMN-A 在交易收集期(t_{CP} , transaction collection period)内从网络信标中获取所有跨域传输密钥节点交易,这些跨域行为形成了特定格式的交易信息。SMN-A 将这些交易逐一发送给 KMC-A,交易信息使用 pk_{KMC-A} 加密,使用 sk_{SMN-A} 签名。为确保包含传输密钥分量的交易信息安全,KMC-A 在交易收集期(t_{PP} , transaction processing period)内验证交易的签名信息完整性,使用 sk_{KMC-A} 解密交易信息内密文,并将所有交易密文封装成交易数据分组,使用 pk_{KMC-B} 进行二次加密。KMC-B 在收到事务数据分组时用 sk_{SMN-B} 解密,重复交易验证步骤并将它们转换为 SMN-B

可读交易版本,解封交易数据分组成交易。上述步骤如图2所示,握手消息信息流过程如下。

步骤 1 SMN-A sends transactions to KMC-A:

$$Enc \{info\}_{PK_{KMC-A}} + Dest_{SMN} + Sig \{Cipher + Dest_{SMN}\}_{SK_{SMN-A}}$$

步骤 2 KMC-A forwards the transactions packet to KMC-B:

$$Enc \{info\}_{PK_{KMC-B}} + Dest_{SMN} + Sig \{Cipher + Dest_{SMN}\}_{SK_{KMC-A}}$$

步骤 3 KMC-B forwards the transaction packet to SMN-B:

$$Enc \{info\}_{PK_{SMN-B}} + Dest_{SMN} + Sig \{Cipher + Dest_{SMN}\}_{SK_{KMC-B}}$$

如果 SMN 位于相同的自治域内,2 个 SMN 由同一 KMC 负责密钥管理,握手步骤会减少。SMN-A-1 将交易转发给 KMC-A 以进行完整性证明,无需交易内容版本切换,直接将所有交易封装成交易数据分组并使用 $pk_{SMN-A-2}$ 加密发送至 SMN-A-2,SMN-A-2 解密交易数据分组并拆封,获得所有密钥传输交易。上述步骤如图3所示,传输密钥信息流如下。

步骤 1 SMN-A-1 sends transactions to KMC-A:

$$Enc \{info\}_{PK_{KMC-A}} + Dest_{SMN} + Sig \{Cipher + Dest_{SMN}\}_{SK_{SMN-A}}$$

步骤 2 KMC-A forwards the transaction packet to SMN-A-2:

$$Enc \{info\}_{PK_{SMN-A-2}} + Dest_{SMN} + Sig \{Cipher + Dest_{SMN}\}_{SK_{KMC-A}}$$

结合区块链的多节点挖矿过程,可以简化密钥传输的握手过程,无论是跨自治域密钥传

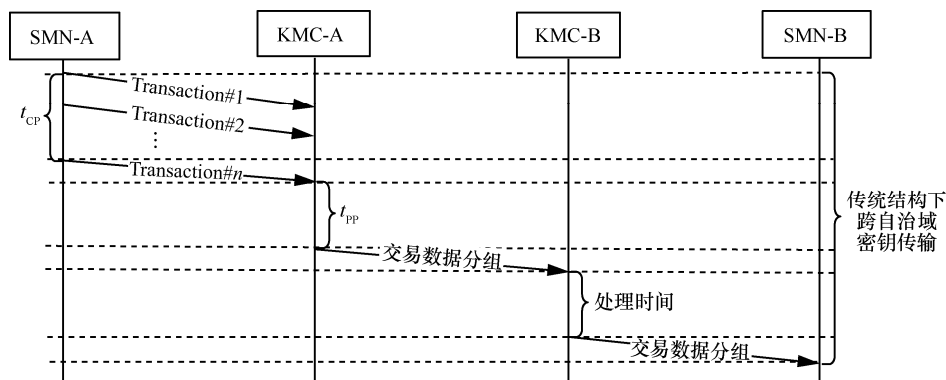


图2 传统结构下跨自治域密钥传输过程

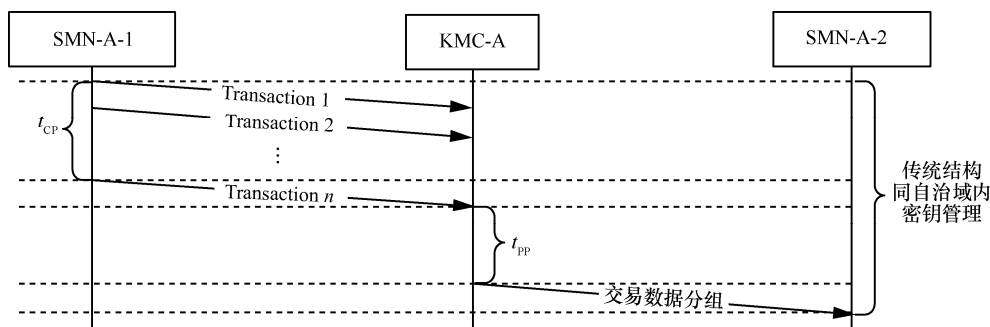


图3 传统结构下同自治域内密钥传输管理过程

输密钥过程还是同自治域内密钥传输过程，都将其视为在类似由全体 SMN 成员构成的分布式异构的云结构环境下全体节点挖掘封装着密钥分量的区块并广播挖掘结果的过程。将封装着待传输密钥数据的区块交由全体 SMN 成员组成的区块链网络对区块及其内容的合法性和完整性进行多方验证，基于此过程从功能角度考虑可以移除中心化的 KMC，并且避免了 KMC 伪造密钥数据。其简化的握手如图 4 所示，在 t_{CP} 内网络允许多个交易广播到 SMN 上，并由该 SMN 负责代理接收交易并将交易组数据封装成区块。为了确保交易信息安全可靠，代理 SMN 使用 $sk_{SMN-Scr}$ 对区块中交易数据进行签名，待区块发布到网络后，交由全体 SMN 节点进行验证。由于区块中待传输数据密文使用 SMN_{Dest} 的公钥加密，可以保证封装着密钥分量的区块在传输到 SMN_{Dest} 前敌手无法截获密钥数据。根据节点挖掘区块相关性质，本文认为应将各项交易以随机顺序插入封装区块中，封装后区块将采用区块挖掘算法交由全体 SMN 成员进行挖掘，并将挖掘区块结果广播回网络。密钥传输信息流过程如下。

步骤 1 SMN-A sends transactions to SMN-Cloud:

$$Enc\{info\}_{PK_{SMN-Dest}} + Dest_{SMN} + Sig\{Cipher + Dest_{SMN}\}_{SK_{SMN-A}}$$

步骤 2 SMN-Cloud returns the mined block to SMN-A

3.2.2 交易信息格式及区块格式

交易信息格式旨在安全封装从 $SMN-Scr$ 到 $SMN-Dest$ 的密钥传输信息，尽可能安全地保持分量密钥数据以密态传输。本文模型的事务标题参考传统比特币的区块体交易内容，其中包含 7 个字段，如表 2 所示。

交易格式中的交易信息 $info$ 包含合法成员的密钥分量及相关信息，使用 $SMN-Dest$ 的 $pk_{SMN-Dest}$ 将 $info$ 加密得到 $Cipher = Enc\{info\}_{pk_{SMN-Dest}}$ 。 $info$ 在基于区块链传输过程中除了 $SMN-Dest$ 和 $SMN-Scr$ 外，其他 SMN 是无法获取的，恶意节点在没有 $sk_{SMN-Dest}$ 的情况下无法从 $Cipher$ 解密读取加密的任何信息，保证了交易信息的机密性和安全性。使用 $SMN-Scr$ 的私钥 $sk_{SMN-Scr}$ 对密文和目的 $SMN-Dest$ 进行签名计算，保证交易完整性不可否认性。敌手在没有 $sk_{SMN-Dest}$ 的情况下无法

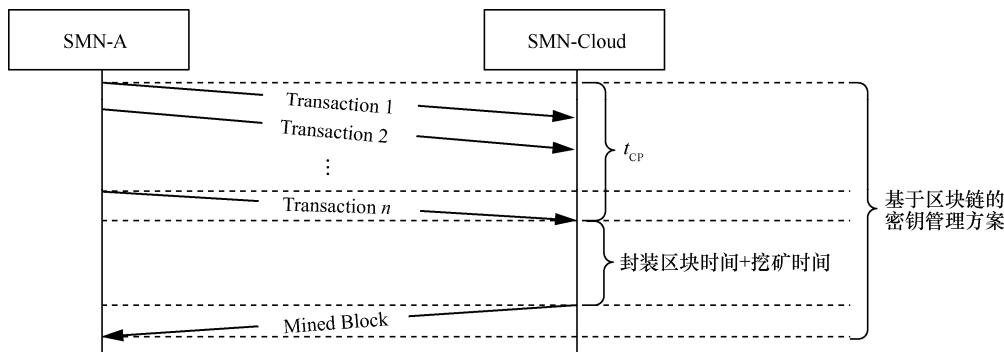


图4 KMSBoB 密钥管理过程

表 2

交易信息格式

内容	说明
Hashed result of the transaction	交易内容杂凑值
Position of this transaction in block	区块交易位置
Current security domain number SMN-this	本安全域网络编号
Beacon	网络信标, 标明是否进行跨 AD 密钥传输
Identity materials	身份相关信息, 包括 ID_{SMN-A} 和 $Cert_{SMN-A}$
Signature of this transaction	交易签名值 $Sig\{Cipher + Dest_{SMN}\}_{sk_{SMN-Src}}$
Payload: {Encrypted Transaction Information}	$Cipher = Enc\{info\}_{pk_{SMN-Dest}}$

伪造签名, 无法冒充正常的 SMN 成员窃听和篡改跨域密钥传输和密钥生成过程。

本文借鉴传统比特币格式^[9], 设计了适应于本文方案的区块结构。区块中 $Hash_{prev}$ 将本区块链与父区块相链接, 是区块链的链式结构根源。区块中的所有事务都以杂凑值的形式合并到 Merkle 根中^[10], 任何交易值的变动都会导致 Merkle 根的值产生巨大差异, 保证了交易的完整性和不可篡改性。时间戳 $Timestamp_{now}$ 用来防止时间篡改。 $Hash_Target$ 的作用是在区块挖掘过程中, 不断调整 $nonce$ 值直到区块头的杂凑值^[11]小于以 n_{zeros} 个数字零开始的 $Hash_Target$ 。SMN 在 t_{CP} 内收集交易, 以随机顺序将这些交易进行排序并封装形成区块, 通过这种方式区块可以聚合跨自治域或同自治域密钥传输交易

请求。封装后的区块统一交由网络中矿工成员进行挖矿操作, 验证区块及其交易内容的合法性及正确性。

3.2.3 挖矿和工作量证明算法

在区块链中, 工作量证明机制是一种数字收据, 其伪造区块是困难的, 但易于网络中节点验证区块合法性。挖矿过程是全体节点执行工作量证明机制的过程, 区块链网络中全体 SMN 成员对区块头进行 $doublehash()$ 穷举杂凑运算, 通过枚举 $nonce$ 重复运算 $doublehash$ (BlockHead)值, 直到结果小于 $HashTarget$ 。基于此交易随机排序方法可以迫使 SMN 对不同的区块头进行穷举杂凑运算, 以不同的时间长度挖掘相同的事务集合, 使整个系统预先阻止区块间竞争记账权冲突。算法 1 描述了本文设计的改进的工作量证明过程, 算法 2

表 3

本文设计的区块格式

区块内容	隶属区块部分	说明	字符长度/bit
ID	区块头	用户标识符 ID, 为公钥的 Base58 匿名值	8
$Hash_{prev}$	区块头	前一区块头杂凑值	32
$Timestamp_{now}$	区块头	时间戳, 基于用户系统生成	4
$BLC_Version$	区块头	区块链版本号, 本文为 1.0	4
$Root_{Merkle}$	区块头	交易内容的 Merkle Root 值	32
BLC_Height	区块头	区块高度	4
$Hash_Target$	区块头	目标难度值	8
$Nonce$	区块头	工作量证明机制调整随机数值	8
$BLC_Payload$	区块体	区块体负载	—
$Transaction_{No1}$	$BLC_Payload$	交易项 Tx_{No1}	—
$Transaction_{No2}$	$BLC_Payload$	交易项 Tx_{No2}	—
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$Transaction_{Non}$	$BLC_Payload$	交易项 Tx_{Non}	—
$Data_Payload$	$Transaction$	交易项中包含相关数据负载, 通常为密钥分量	—

描述了全体 SMN 验证区块合法性过程。

算法 1 基于改进的工作量证明过程 (IPoW, improved proof of work)

输入 待验证区块头相关元素: BlockHead_{temp}
 $\{\text{BLC_Version}, \text{Hash}_{prev}, \text{Timestamp}_{now}, \text{HashTarget}, \text{BLC_Height}, \text{Trans}=[\text{null}]\}$
 待写区块体相关数据集 $\text{Trans}=[\text{null}, \text{null}, \dots, \text{null}]$

待写入区块的隐私数据负载 $\text{PayloadSecretData}_{No1}, \text{PayloadSecretData}_{No2}, \dots, \text{PayloadSecretData}_{Non}$

输出 Nonce 值: nouce ; 区块高度 BLC_Height

1) **Start**

2) 转换 $\text{secretdata}_{No.i}$ 为相应数据负载

for $i=[1, n]$

$\text{Cipher} = \text{Enc}\{\text{PayloadSecretData}_{No.i}\}_{pk_{SMN-Dest}}$;

$\text{Tx}_{Noi} = \text{Sig}\{\text{Cipher} + \text{Dest}_{SMN}\}_{sk_{SMN-Src}} + \text{To}\{\text{ID}_{SMN-Dest} + \text{Address}_{SMN-Dest}\} + \text{From}\{\text{ID}_{SMN-Src} + \text{Address}_{SMN-Src}\}$;

Tx_{Noi} 写入区块体数据集, $\text{Trans}=[\text{Tx}_{No1}, \text{Tx}_{No2}, \dots, \text{Tx}_{Non}]$;

3) 初始化布尔变量值 $\text{gotAns}=\text{False}$;

4) **while** (NOR gotAns) **do**

5) 转换交易编号 $[1, n]$, 生成交易顺序 $a_{rand} = \text{randPerm}(n)$;

6) 依据交易顺序 a_{rand} 计算 Merkle 根值 Root_{Merkle} ;

7) 计算区块头临时杂凑值 $\text{Hash}_{temp} = \text{BLC_Version} || \text{Hash}_{prev} || \text{Root}_{Merkle} || \text{Timestamp}_{now} || \text{HashTarget} || \text{BLC_Height}$;

8) 封装新产生区块 $\text{Block}_{new} = \{\text{BlockHead}_{temp}, \text{Trans}\}$;

9) 初始化穷举数 $\text{nonce}=0$, 初始化杂凑输出值 $\text{result}=0$;

10) **while** (NOR gotAns & 没有从其他节点获得算法 1 的值) **do**;

11) $\text{result} = \text{doublehash}(\text{Hash}_{temp} || \text{nonce})$;

12) $\text{nonce}++$;

13) **if** ($\text{result} < \text{HashTarget}$ & 没有从其他节点获得算法 1 的值)

14) 将 $\text{nonce}-1$ 写入 Nonce 值域;

15) $\text{gotAns}=\text{True}$;

16) **return** ($\text{nonce}-1, \text{BLC_Height}$);

17) **else if** (如果从其他节点获得算法 1 的值)

18) $\text{gotAns}=\text{True}$;

19) **return** (Null, Null);

20) **end if**

21) **end while**

22) **end while**

23) **End**

算法 2 验证区块合法性 (PLOF, proof the legality of block)

输入 已挖掘区块头 BlockHead_{mined} ; 区块负载 BLC_Payload

输出 布尔值 isCorrect ; 更新区块链 $\text{Blockchain}_{update}$

默认共识值: $\text{Blockchain}_{previous}(\text{Height}-1)$

1) **Start**

2) 提取随机数值 $\text{nonce} = \text{getNonce}(\text{BlockHead}_{mined})$;

3) 基于计算历次 BLC_Payload 值的杂凑值累加和形成 Root_{Merkle} ;

4) 创建验证区块头 $\text{Hash}_{very} = \text{BLC_Version} || \text{Hash}_{prev} || \text{Root}_{Merkle} || \text{Timestamp}_{now} || \text{HashTarget} || \text{BLC_Height}$;

5) 产生待验证字符串: $\text{Input}_{verify} = \text{Hash}_{veri} || \text{nonce}$;

6) 形成待计算字符串杂凑值: $\text{result} = \text{doublehash}(\text{Input}_{verify})$;

7) **if** ($\text{result} < \text{HashTarget}$);

8) $\text{isCorrect}=\text{TRUE}$;

9) 构成区块 $\text{Block}(\text{BlockHead}_{mined}, \text{BLC_Payload})$;

$\text{Blockchain}_{update}(\text{Height}) = \text{Blockchain}_{previous}(\text{Height}-1) + \text{Block}(\text{BlockHead}_{mined}, \text{BLC_Payload})$;

10) **else**

11) $\text{isCorrect}=\text{FALSE}$;

12) **end if**

13) **return** ($\text{isCorrect}, \text{Blockchain}_{update}(\text{Height})$);

14) **End**

基于算法 2 使全体节点在网络可以得到相同的区块链记录结合, 多方验证和维护数据的合法性。全体节点对区块数据完整性和合法性负责, 全体 SMN 成员共同承担生成会话密钥安全风险。

3.2.4 会话密钥生成协议

本文结合文献[5,6]的会话密钥生成方案,将协议过程引入区块链中,提出基于 SMN 群组的区块链网络环境下的密钥协商算法。基于区块链的算法 1 和算法 2,本文方案屏蔽密钥生成和密钥管理策略上的差异,统一视为动态的新会话密钥生成过程。将全体异构网络环境下的 SMN 成员视为统一密钥管理域内,在全局范围内对 SMN 进行统一的域内编号 *Address*。默认网络中有 n 个 SMN 成员,SMN 成员间通过区块链进行隐私数据转发,并由全体 SMN 基于算法 2 验证区块链内数据的合法性,同时同步区块链数据。基于本文设计的协议四轮密钥协商过程中,全体 SMN 成员同步协商密钥数据,SMN 成员既是生成方也是验证方,全体 SMN 节点可以得到完全一样的会话密钥 K 。会话密钥生成协议具体过程如下。

Initialize

Step1: 全体 SMN 成员生成 G_1 和 G_2 为二阶为素数 q 的乘法循环群,令 $e:G_1 \times G_1 \rightarrow G_2, g$ 是 G_1 的生成元;

Step2: 全体 SMN 成员安全生成各自公私钥对。

Round 1st

Step1: 成员 SMN_i 生成 $Rand() \rightarrow s_i$;

Step2: 生成 g^{s_i} ;

Step3: 计算 $m_i = Enc(g^{s_i})_{pk_{SMN_j}}$;

Step4: $Hash_{SMN_i} = Hash(m_i)Signature_{SMN_i} = Sig(Hash_{SMN_i})_{sk_{SMN_i}}$;

Step5: 封装数据负载 $info_{SMN_i} = \{Signature_{SMN_i}, Hash_{SMN_i} \text{ 和 } m_i\}$ 中;

Step6: 一段时间内 SMN_i 统一封装区块头和包括 $info_{SMN_i}$ 在内的所有负载,并执行算法 1 将该区块发布至网络中,设置 $Dest = Address.j, Src = Address.i$;

Step7: 全网 SMN 成员根据接收到的区块,共同执行算法 2,集体验证区块内容和结果的合法性,集体备份区块链数据;

Step8: SMN_i 计算 $Position_{SMN_j} = Enc(Height)_{pk_{SMN_j}}$, 发送 $KeyGenerationRequest(Position, SMN_{Dest}, Round 1^{st})$ 至 SMN_{Dest} 。

Round 2nd

当 SMN_i 接收到 Round 1st 的 $KeyGeneration-$

$Request(Position, SMN_i)$ 后执行:

Step1: 成员 SMN_i 执行 $Dec(Position_{SMN_j})_{sk_{SMN_i}} \rightarrow Height$, 通过 $Height$ 检索相应区块;

Step2: **for** $j=[1, n]$;

提取区块数据负载

if $(Ver(Signature_{SMN_j})_{pk_{SMN_j}} == Hash_{SMN_j})$

$Dec(m'_j)_{sk_{SMN_j}} \rightarrow g^{s_j}$, 由此得到全网 SMN 成员的密钥分量 g^{s_j}

end for

Step3: 成员 SMN_i 生成 $Rand() \rightarrow r_i, P_i = g^{r_i s_i - 1}, Q_i = g^{r_i s_i - 1}$;

Step4: 计算 $m_i = Enc(P_i)_{pk_{SMN_{i-1}}}, m'_i = Enc(Q_i)_{pk_{SMN_{i-2}}}$;

Step5: 计算 $Hash_{SMN_i} = Hash(m'_i), Hash'_{SMN_i} = Hash(m_i), Signature'_{SMN_i} = Sig(Hash'_{SMN_i})_{sk_{SMN_i}}, Signature''_{SMN_i} = Sig(Hash_{SMN_i})_{sk_{SMN_i}}$;

Step6: 封装数据负载 $info_{SMN_i} = \{Signature'_{SMN_i}, Hash'_{SMN_i}, m'_i\}, info''_{SMN_i} = \{Signature''_{SMN_i}, Hash_{SMN_i}, m_i\}$;

Step7: 一段时间内 SMN_i 统一封装区块头和包括 $info_{SMN_i}$ 和 $info''_{SMN_i}$ 在内的所有数据负载,并执行算法 1 将该区块发布至网络中。设置 $Dest' = Address.i-1, Dest'' = Address.i-2, Src = Address.i$;

Step8: 全网 SMN 成员根据接收到的区块,共同执行算法 2,集体验证区块内容和结果的合法性,集体备份区块链数据;

Step9: SMN_i 计算 $Position'_{SMN_j} = Enc(Height')_{pk_{SMN_j}}$, 发送 $KeyGenerationRequest(Position, SMN_{Dest'}, SMN_{Dest''}, Round 2^{nd})$ 至 $SMN_{Dest'}$ 和 $SMN_{Dest''}$ 。

Round 3rd

当 SMN_i 接收到 Round 2nd 的 $KeyGeneration-Request(Position, SMN_{Dest'}, SMN_{Dest''}, Round 2^{nd})$ 后执行:

Step1: 成员 SMN_i 执行 $Dec(Position'_{SMN_j})_{sk_{SMN_i}} \rightarrow Height'$, 通过 $Height'$ 检索相应区块;

Step2: 提取区块数据负载

if $(Ver(Signature'_{SMN_j})_{pk_{SMN_j}} == Hash'_{SMN_j} \&\& (Ver(Signature''_{SMN_j})_{pk_{SMN_j}} == Hash_{SMN_j})$

$Dec(m'_i)_{sk_{SMN_i}} = g^{r_i s_i - 1} Dec(m''_i)_{sk_{SMN_i}} = g^{r_i s_i - 1}$

Step3: 计算 $k_i = e(g^{r_i + 1 s_i s_i - 1}, g^{r_i + 2 s_i s_i - 1})^{r_i} = e(g, g)^{r_i^2 + 1 r_i + 2}$;

Step4: 计算 $m_i''' = Enc(k_i)_{pk_{SMN_j}}$;

Step5: 计算 $Hash'''_{SMN_i} = Hash(m_i''')$,

$Signature_{SMN_i}''' = Sig(Hash_{SMN_i}')_{sk_{SMN_i}'};$

Step6: 封装数据负载 $info_{SMN_i}''' = \{ Signature_{SMN_i}''', Hash_{SMN_i}', m_i'' \};$

Step7: 一段时间内 SMN_i 统一封装区块头和包括 $info_{SMN_i}'''$ 在内的所有数据负载, 并执行算法 1 将该区块发布至网络中, 设置 $Dest''' = Address.j$, $Src = Address.i$;

Step8: 全网 SMN 成员根据接收到的区块, 共同执行算法 2, 集体验证区块内容和结果的合法性, 集体备份区块链数据;

Step9: SMN_i 计算 $Position_{SMN_j}''' = Enc(Height_{SMN_j}')_{pk_{SMN_j}'}$, 发送 $KeyGenerationRequest$ ($Position$, SMN_{Dest}''' , Round 3^{rd}) 至 SMN_{Dest}''' 。

Round 4^{th}

当 SMN_i 接收到 Round 3^{rd} 的 $KeyGenerationRequest(Position, SMN_{Dest}''', Round 3^{rd})$ 后执行:

Step1: for $i = [1, n]$

$Dec(Position_{SMN_i}')_{sk_{SMN_i}'} \rightarrow Height'''$, 通过 $Height'$ 检索相应区块

提取区块数据负载

if ($Ver(Signature_{SMN_i}')_{pk_{SMN_i}'} = Hash_{SMN_i}')$

$Dec(m_i')_{sk_{SMN_i}'} = k_i$

end for

Step2: 生成会话密钥

$$K = \prod_{i=1}^n k_i = e(g, g)^{r_1 r_2 r_3 + r_2 r_3 r_4 + \dots + r_n r_1 r_2}$$

基于区块链多方认证的过程, 可以将密钥更新过程, 视为全体 SMN 成员再次生成新的密钥, 协议过程不变, 简化了密钥管理过程。

4 安全性分析与效率测试

4.1 安全性分析

本文设计的基于区块链密钥管理策略的会话密钥生成过程中, 每次生成的会话密钥的密钥分量是由每个 SMN 成员随机数生成的, 所以每次基于协议过程所生成会话密钥都是唯一的。且当 SMN 成员组重新执行协议, 各个 SMN 成员的密钥分量都会不一样, 所以本文设计协议满足已知密钥安全性、前向保密性和密钥不可控制性。此外, 本文设计的协议策略中的区块封装过程设计了签名认证和验签过程, 故能抵抗密钥泄露攻击。基于区块链多方竞争记账权过程, 全体 SMN 成员都会备份区

块链所记录的密钥数据分量, 满足密钥共享可知性。

由于区块链数据是面向所有成员公开的, 如果存在敌手通过冒充 SMN 备份区块链从 Round 2^{nd} 中获得 P_i 和 Q_i , 那么进行如下分析。在获得 P_i 和 Q_i 的情况下, 由于敌手在不知道 Round 1^{st} 中随机数产生的密钥分量的情况下, 敌手所能计算的密钥分量仅为 $k_i = e(g^{r_i r_i + 1 s_i}, g^{r_i r_i + 2 s_i}) = e(g, g)^{r_i r_i + 1 r_i + 2 s_i^2}$, 而全体 SMN 节点所产生的密钥分量为 $k_i = e(g^{r_i + 1 s_i s_i - 1}, g^{r_i + 2 s_i s_i - 1})^{r_i} = e(g, g)^{r_i r_i + 1 r_i + 2}$, 故敌手无法得到最终的会话密钥, 保证了会话密钥在公开网络环境下安全性。

4.2 效率分析

分析全体 SMN 进行单次密钥协商协议过程: Round 1^{st} 中, 全体 SMN 成员各计算一次, 共发起 $n-1$ 次交易; Round 2^{nd} 全体 SMN 成员计算 P_i 和 Q_i , 发送了 2 次交易; Round 3^{rd} 中全体 SMN 计算一次自己的密钥分量, 发送了 $n-1$ 次交易; Round 4^{th} 中, 全体 SMN 成员在解密出全体 SMN 成员的分量后, 计算合成最终会话密钥。因此单个 SMN 节点在一次密钥协商过程中通信轮数为 3 轮, 产生交易量为 $2n$, 计算次数为 4 次。

4.3 性能测试

本文对 KMSBoB 性能进行测试, 并对所得结果进行分析。性能测试分为 3 个部分。第 1 部分对 KMSBoB 处理相关密码学运算时间进行测试; 第 2 部分对比测试 KMSBoB 与集中式密钥管理方案和分布式密钥管理方案通信开销; 第 3 部分测试了 KMSBoB 处理交易数量的性能。

4.3.1 仿真环境

本文采用 OMNeT++ 4.5^[12] 与专用网络仿真数据分组^[13] 仿真。仿真实验部署在 5 台物理服务器上 and 1 台式计算机上。每台服务器配备 2 张独立网卡, 每张独立网卡都设置有 50 个虚拟 IP 地址。将这些虚拟 IP 地址设置为相应节点的 ID (共计 5 000 个节点), 并进行初始化操作。服务器的配置如下: 处理器采用 Intel Xeon Processor E5-2670 (20 M Cache, 2.60 GHz, 8.00 GT/s Intel QPI), 双 CPU; 内存为 8 GB; 内存显卡 GeForce M150 (2G); 操作系统采用 Ubuntu 16.04.4 LTS server, 内核版本号为 4.4.0-31-generic, 5 台服务器分别命名 node1, node2, ..., node5。服务器

设备可以完成每秒 250×10^4 次的杂凑计算。网络通信中,区块设计中 Cipher 默认长度为 64 byte,其中,48 byte 用于存储 info。选取 OpenZeppelin 开源架构作为底层智能合约框架,网络通信采用 IEEE 802.11 a/c 协议。基于默认 SMN 成员是可信实体的事实,本文认为这个安全级别足以保证网络的安全。

两部分模拟的表现集中在区块交易处理时间上,结果仅取决于交易的总数。因此,本文进行以下模拟假设:① 在每个 t_{CP} 结束时,网络中会涌入一定数量的密钥更新事务;② 每个 SMN 密码运算的处理时间结果,是测试来自 SMN 的数据平均结果;③ 为测试 KMSBoB 处理交易数量的性能,将获得结果的放大视图,从 0 到 100 的交易编号。设置在区块封装最多包含 500 个事务,并设置模拟交易 5 000 笔。

4.3.2 密码运算的处理时间测试

由于密钥传输时间需要考虑方案相关密码学运算时间,本文模拟了 KMSBoB 的密码学运算的时间成本,包括加密、解密、签名、验证、挖矿和区块封装等密码学运算时间。图 5 显示了在密钥传输过程中密码运算的性能,除去全部节点挖矿操作外,其余运算时间均是随交易次数的增加而线性增加。由于挖矿算法对象是挖掘区块头,采矿处理时间是多次模拟的平均值。网络体系中只接受最快 SMN 成员挖掘的区块,因此测试的时间低于平均值。加密和解密方案花费相近的处理时间,签名和验证算法花费计划间最长的计算时间,这 4 组运算随交易数量呈比例线性增长。表 4 记录每个密码方案的平均处理时间。

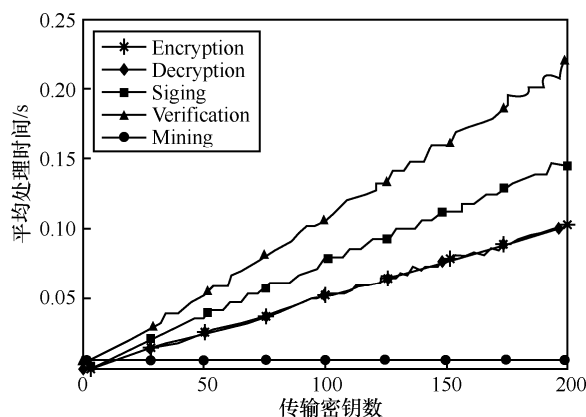


图5 基于不同交易数量的密码学运算平均时间

表4 平均密码学运算时间

密码学运算项目	平均处理时间/milliseconds
Enc()	0.489
Dec()	0.832
Sig()	0.495
Ver()	1.276
区块挖矿	5.331

图 6 给出不同区块交易数量下区块封装时间。由结果可知,随着待认证交易项数量的增长,区块封装时间呈指数增长。单个区块包含 300 个交易项内处理时间缓慢增加,交易项大于 400 个处理时间超过 0.1 s。当超过 1 000 个交易项,区块封装时间达到近 1 s。

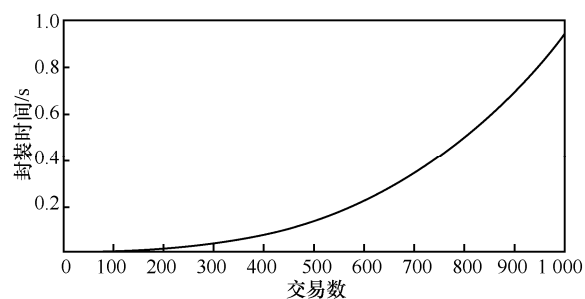


图6 不同区块交易数量下区块封装时间

4.3.3 方案开销对比

为了评估 KMSBoB 性能,将 KMSBoB 与文献[1, 6]中的传统方案进行比较,将文献[1]中的方案的数据结果被用作仿真实验对比基准,其中文献[1]提出了经典密钥管理方案,文献[6]提出了典型分布式密钥管理方案,本文以文献[1]的方案为基准对比。针对 2 种情况下进行对比实验。

- 1) 平均密钥生成时间计算开销。
- 2) 在同一 KMC 管理域内的 2 个 SMN 成员间的密钥传输通信开销。
- 3) 2 个 KMC 管理域下,2 个独立的安全域之间密钥传输通信开销。

实验 1)的结果如图 7 所示,KMSBoB 方案的密钥生成时间平均为 1.18 s,小于文献文献[6]和文献[1]平均密钥生成时间,说明 KMSBoB 方案在密钥生成速度上具有一定的优势,其可以满足分布式网络环境下的密钥生成时间开销。基于密钥协商过程,认为密钥生成时间开销节约主要是在密钥传输过程中产生的,本文方案只有一次密钥协商过程,较大地减少了数据传输的开销。

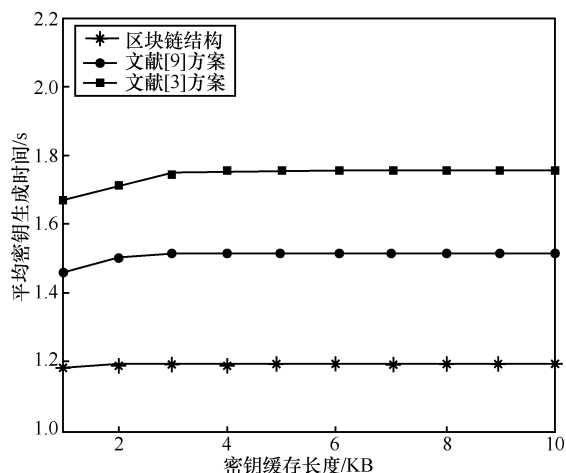


图7 平均密钥生成时间开销

图8和图9描述了KMSBoB在传输密钥数量方面的性能。如图8所示，实验2)中的同域密钥传输开销对比。完成传输500次交易，本文方案时间成本大约为1.1s，文献[6]结构中处理同样数量的交易量需要花费近两倍的时间，文献[1]成本是本文方案密钥传输时间的3倍。但是本文方案传输时间呈亚指数式增长，在交易数量为1500时本文方案和文献[1]两条曲线相交，在交易数量大于1500时较文献[1]处理时间增加。但是在交易数量少于1500时，本文方案相对于集中式密钥管理方案和分布式密钥管理方案提供了更好的可扩展性。另外，基于区块链的方案节省了近一半的处理时间，交易数量1500到2000时，本文方案低于文献[1]的基准量。

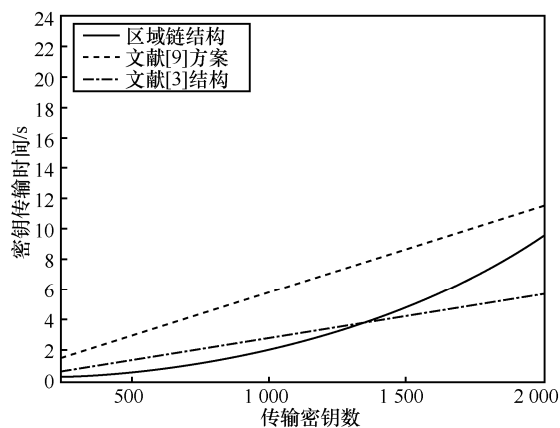


图8 同域密钥管理环境下密钥传输时间对比

图9中显示了实验3)跨域密钥管理的传输时间开销对比。KMC在传统方案中将密钥分量从一个安全域转换为另一个安全域，2个KMC需要相互通信以完成密钥传输。因此，KMC之间的额

外握手过程会导致繁琐的密钥传输时间。当文献[1]和文献[6]交易数量分别超过800和1800时，密钥传输时间成本均超过10s。而本文设计的基本区块链密钥管理方案的时间成本要少很多。因此，由于较少的处理时间成本，区块链结构在实验2)中相对于传统结构具有较好的可扩展性和较少的时间代价。

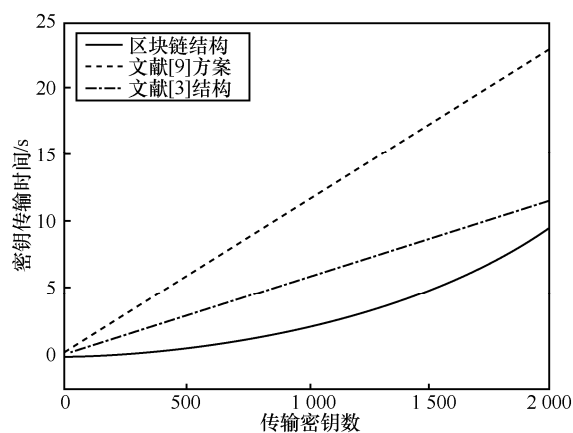


图9 跨域密钥管理环境下密钥传输时间对比

4.3.4 方案性能评估

方案性能评估主要是评估节点在交易收集期 t_{CP} 内，SMN允许的最大密钥传输交易事务收集量。 t_{CP} 时间长度决定了涌入SMN网络的交易量。图10显示了基于不同 t_{CP} 内的密钥传输的结果。观察曲线可知，基于不同 t_{CP} 曲线的密钥传输时间，当 t_{CP} 超过0.8s时，曲线有明显上升趋势。 t_{CP} 的增加使SMN负载更多的交易量，但同时也会增加密钥传输所需时间。可观察到交易数量与网络节点数量呈正比， t_{CP} 越长，SMN捕获的交易越多。密钥传送时间全都小于0.8s，密钥数据传输代价是可以接受的。

5 结束语

本文提出了KMSBoB密钥管理的方案，将MTI-CO协议引入基于区块链的密钥管理算法中，简化了密钥管理策略。通过实验测试，所提方案在批量密钥方面相比传统密钥管理方案具有更少的通信开销、更高的密钥传输效率和更好的动态适应性。下一步工作将研究基于区块链模式下高效地存储密钥数据方案以及区块链快速检索技术。

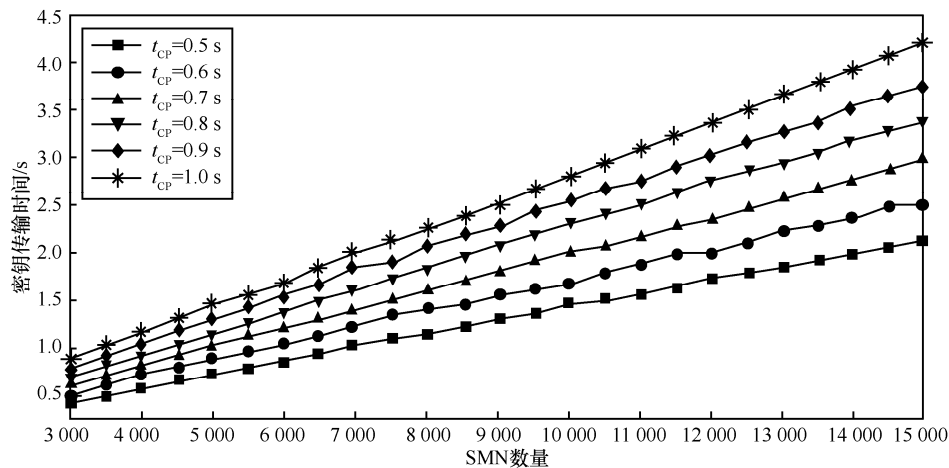


图10 基于不同交易收集期的密钥传输时间

参考文献:

- [1] SUN Y, TRAPPE W, LIU K J R. A sKMClablemultiKMCst key management scheme for heterogeneous wireless networks[M]. IEEE Press, 2004.
- [2] ZHOU W, XU Y, WANG G. Decentralized group key management for hierarchiKMCI access control using multilinear forms[J]. Concurrency & Computation Practice & Experience, 2016, 28(3): 631-645.
- [3] 张志军, 郭渊博, 刘伟, 等. 基于 m 叉树与 DH 协议的组密钥协商协议[J]. 计算机工程, 2010, 36(1): 161-163.
ZHANG Z J, GUO Y B, LI W, et al. Group key agreement protocol based on m-tree and DH protocol[J]. Computer Engineering, 2010, 36(1): 161-163.
- [4] YU S C, REN K, LOU W J. Attribute-based on-demand multicast group setup with membership anonymity[J]. Computer Networks, 2010, 54(3): 377-386.
- [5] HE D, CHEN C, CHAN S, et al. Secure and efficient handover authentication based on bilinear pairing functions[J]. IEEE Transactions on Wireless Communications, 2012, 11(1):48-53..
- [6] HE D, CHANS, GUIZANI M, Handover authentication for mobile networks: security and efficiency aspects[J]. IEEE Network, 2015, 29(3): 96-103.
- [7] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system[J]. Consulted, 2008
- [8] 荆继武, 林璟铨, 冯登国. PKI 技术[M]. 北京: 科学出版社, 2008:6.
- [9] JING J W, LIN J Q, FENG D G. PKI technology[M]. Beijing: Science Press, 2008:6.
- [9] Bitcoin blockchain size[EB/OL] <http://blockchain.info/charts/blocks-size>.
- [10] MERKLE R. C, A digital signature based on a conventional en-cryption function[C]//Advances in CryptologyCRYPTO87. Springer, 1987, 369-378
- [11] AITZHAN N. Z, SVETINOVIC D., Security and privacy in de-centralized energy trading through multi-signatures, blockchain

and anonymous messaging streams,[J]IEEE Transactions on De-pendable and Secure Computing, 2016,99:1-1.

- [12] VARGA et al. The Omnet++ discrete event simulation system[C]// Proceedings of the European simulation multiconference (ESM2001), 2001.
- [13] SOMMER C, GERMAN R, DRESSLER F. Bidirectionally coupled network and road traffic simulation for improved ivc analysis[J]. Mobile Computing, IEEE Transactions on, 2011, 10(1): 3-15.

[作者简介]



戴千一(1994-), 男, 陕西西安人, 战略支援部队信息工程大学硕士生, 主要研究方向为区块链安全、区块链分布式应用。

徐开勇(1963-), 男, 河南信阳人, 战略支援部队信息工程大学研究员, 主要研究方向为信息安全与可信计算。

郭松(1985-), 男, 河北保定人, 博士, 战略支援部队信息工程大学讲师, 主要研究方向为信息安全。

蔡国明(1976-), 男, 湖北云梦人, 博士, 战略支援部队信息工程大学副研究员, 主要研究方向为密码学与信息安全。

周致成(1992-)男, 河南郑州人, 硕士, 61660 部队助理工程师, 主要研究方向为区块链网络安全。