计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(8):16−23 [doi: 10.15888/j.cnki.csa.007465]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于国密算法的区块链架构①

杨　洵, 王景中, 付　杨, 王宝成

(北方工业大学 信息学院, 北京 100144)
通讯作者: 杨　洵, E-mail: 1179234467@qq.com

摘　要: 本文设计并提出了基于国密算法的区块链架构—“国密链”, 以国密算法 SM2、SM3 替换国际通用密码算法的 ECC、SHA-256, 实现了区块链架构的自主可控. 同时, 针对当前区块链架构面临的共识算法妥协的现状, 设计了“可插拔共识”协议, 解决了当前区块链架构面临的共识算法的不可更改的问题.实验结果表明, “国密链”与普通区块链架构在一致性、有效性相近的情况下, 拥有更高的共识效率、更低的资源开销.

关键词: 国密链; 国密算法; 可插拔共识

引用格式: 杨洵,王景中,付杨,王宝成.基于国密算法的区块链架构.计算机系统应用,2020,29(8):16–23. http://www.c-s-a.org.cn/1003-3254/7465.html

Blockchain Architecture Based on Domestic Cipher Algorithm

YANG Xun, WANG Jing-Zhong, FU Yang, WANG Bao-Cheng

(School of Information Science and Technology, North China University of Technology, Beijing 100144, China)

Abstract: This study designs and proposes a block chain architecture based on the domestic cipher algorithm — “DCchain”, and replaces ECC and SHA-256 of the international general cryptographic algorithm with SM2 and SM3, which realizes the autonomous control of the block chain architecture. At the same time, according to the current status of consensus algorithm compromise faced by block chain architecture, a “pluggable consensus” protocol is designed to solve the unalterable problem of consensus algorithm faced by block chain architecture. Experimental results show that “DCchain” and ordinary block chain architecture have higher consensus efficiency and lower resource cost under the condition of consistency and effectiveness.

Key words: DCchain; domestic cipher algorithm; pluggable consensus

区块链作为一个可靠的分布式数据库, 被广泛应用于版权保护、数据共享、大数据确权等领域. 例如: 吴健提出了以区块链技术为核心的数字版权保护方案[1], 指出区块链技术在版权保护中具备安全性高、成本低、可溯源等优势. 薛腾飞等提出基于区块链的医疗数据共享模型[2], 该架构在医疗方面解决了不同层级机构的数据共享难题. 王海龙将区块链应用于数据确权, 以完整性和持久性为中心, 采用区块链实现证据与确权结果的强一致性, 与传统确权手段相比, 提高了技术的可信度, 并降低了被篡改的风险[3]. 但是, 上述文章只是将区块链架构应用于不同的场景, 忽略了区块链架构面临的共识算法的不可更改的问题.

区块链架构按应用场景划分为两种, 第一种为需要全社会达成共识的公有链架构; 另一种只需要合作伙伴之间达成共识的联盟链架构. 前者采用 POW、DPOS[4] 等算法, 可以在任意用户数量的环境下达成共识, 但挖矿会造成大量资源浪费, 且达成共识周期较长; 后者采用 PBFT[5] 算法, 在少量节点的环境下可以快速、高效的达成共识, 但在多用户环境中效率较低, 无法满足多节点情况下对于共识速度方面的需求. 虽然

① 收稿时间: 2019-12-03; 修改时间: 2019-12-27; 采用时间: 2020-01-02; csa 在线出版时间: 2020-07-29

两类共识算法各有利弊, 但由于目前没有一种共识算法可以兼顾多节点与共识效率, 因此没有一个区块链架构可以满足不同的应用需求.

区块链技术是 P2P 网络协议、分布式数据库、密码学技术等多种技术的集成创新, 以密码机制和共识机制最为核心[6].

目前, 国密算法 SM2、SM3 的可证安全性已经达到了密码算法的最高安全级别, 其实现效率相当或略高于国际标准的密码算法, 相较于比特币、超级账本等现有区块链架构选用的国际通用密码算法 ECC、SHA-256, 拥有安全、稳定、高效等优势. 文章以 SM2、SM3 为基础, 提出了基于国密算法的国密链架构, 实现更高层面的自主可控. 另外, 针对当前区块链架构共识算法不可更改的现状, 本文设计了“可插拔共识”协议, 即通过设计通用共识接口的方式, 使开发者可以灵活选用适当的共识算法, 以满足系统需求.

1 预备知识

1.1 SM2/SM3 算法

SM2 基于求解离散对数问题, 是我国自主研发的椭圆曲线密码算法, 相较于 ECC-256 算法, SM2 算法在解密正确性判断、明文编码问题、对待加密数据长度的限制及加密计算等方面具有更高的效率, 在实际应用中, SM2 算法具有速度快、损耗低的特点, 比 ECC-256 算法更具有优势[7].

SM2 签名算法主要分为 3 个步骤实现消息的签名: (1) 密钥对生成; (2) 用户哈希生成; (3) 数字签名生成.

设用户 A 为签名者, 通过以下步骤获取待签名消息 M 的数字签名 (r, s), 其中参数说明如表 1 所示.

表 1 SM2 参数对照表

符号	含义
M	待签名消息
r,s	数字签名
(d_A,p_A)	用户A的私钥、公钥
(x_A,y_A)	公钥坐标
G	椭圆曲线的基点
x_G,y_G	基点坐标
ID_A	用户A的身份标识
$ENTL_A$	ID_A的长度标识
Z_A	用户A的哈希值
$\bar{M}$	预处理后的待签名消息

(1) 密钥对生成

① 用随机数发生器生成私钥$d_A \in [1, n-2]$.

② G 为基点, 生成公钥:

$$p_A = [d_A]G = (x_A, y_A) \tag{1}$$

(2) 用户哈希生成

① 计算用户 A 的哈希值:

$$Z_A = SM3(ENTL_A||ID_A||a||b||x_G||y_G||x_A||y_A) \tag{2}$$

(3) 数字签名生成

数字签名生成分为消息预处理, 签名计算两部分, 具体过程如图 1 所示.

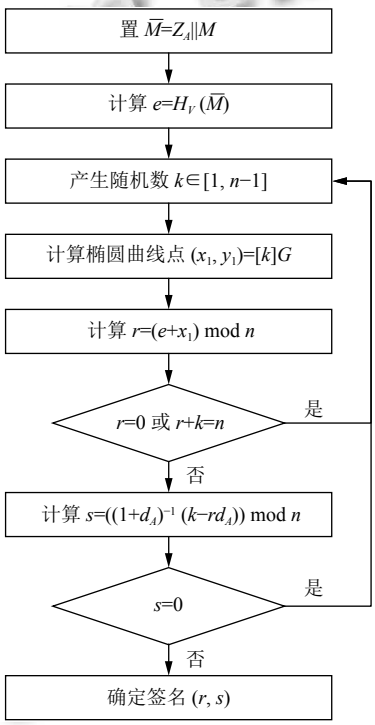

图 1 SM2 签名过程

如图 2 所示, SM2 验签算法通过用户哈希 Z_A、用户公钥 p_A 对数字签名 (r, s) 进行验签计算, 以确定签名是否通过, 完成签名的验证过程.

SM3 是我国自主研发的散列密码算法. 在国密算法体系中, SM3 主要用于数字签名及其验证、消息认证码生成及其验证和随机数生成. 具体算法流程如图 3 所示.

SM3 的压缩函数与 SHA-256 的压缩函数具有相似的结构, 但 SM3 设计更加复杂, 如 SHA-256 压缩函数的每一轮都使用 2 个消息字[8]. SM3 在避免高概率局部碰撞方面效果显著, 能够有效抵抗强碰撞性“差分分析”、碰撞性“线性分析”等密码分析方法[9].

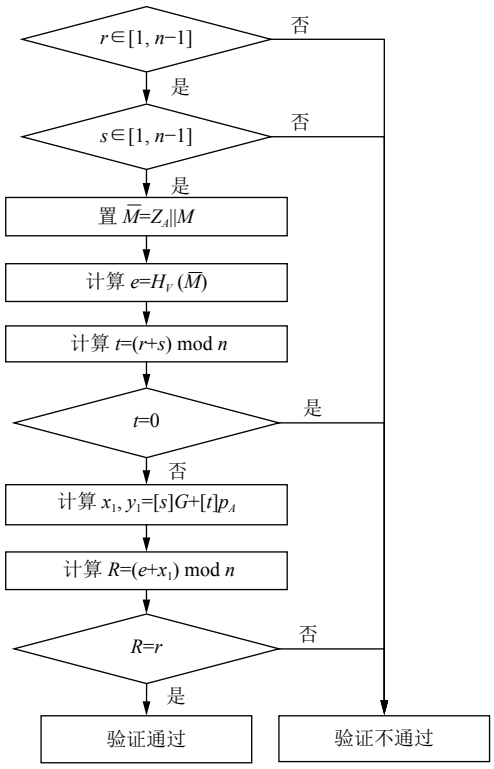

图 2 SM2 验签过程

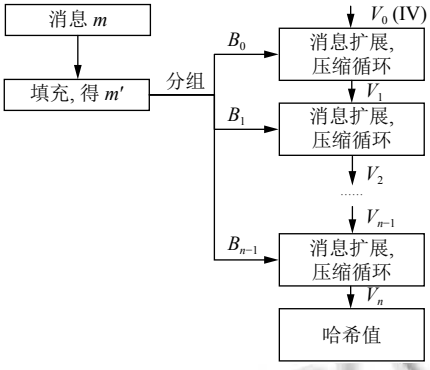

图 3 SM3 算法流程

1.2 区块链

《2018 中国区块链产业白皮书》[10] 指出, 区块链作为一项重点前沿技术, 在供应链、征信、溯源、版权等领域广泛应用, 在一定意义上促进了技术变革. 区块链本质是一种结合散列算法、公钥密码体制、分布式一致性等技术所形成的安全机制, 具备了弱中心化、防篡改、高度透明和可溯源等优良特性. 从数据存储层面上来讲, 区块链作为一种非关系型数据库, 各个区块中包含事务信息, 在打包事务信息产生新区块之前, 利用散列算法的计算机制连接前一区块, 在发生篡改时将引起一系列波动, 并且块与块之间形成单点式对应, 从而可以验证整个链式结构的完整性和有效性. 如图 4 所示, 在比特币中, 当前区块里面包含上一个区块的哈希值, 后面一个区块又包含当前区块的哈希值, 以这样连接起来, 形成一个数据块链条.

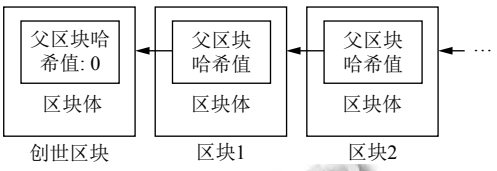

图 4 比特币中的区块链架构

链式结构增加了区块链数据的篡改难度. 区块链是一个被所有节点共同维护的数据链条, 若某节点修改其中某一区块的内容, 其余节点验证区块时会发现问题, 数据不会被全网接收为合法数据. 为了使篡改数据合法, 则需要掌握全网 50% 的算力下, 修改后续所有区块内容, 其代价远超篡改数据后所得到利益.

区块链中的每一条交易数据以散列值的形式记录在区块中, 一方面提供了数据可追溯性, 另一方面也间接保证了链上数据的公开透明.

1.3 共识机制

作为区块数据达成一致的核心, 共识机制主要体现为“在对等网络中, 互不信任的节点通过预置规则校验数据, 达到数据一致性”的过程. 区块链作为一个分布式账本, 共识机制是衡量架构设计优劣的关键因素. 区块链典型的共识算法有工作量证明 (POW) 和实用拜占庭容错算法 (PBFT) 等.

(1) 工作量证明算法 (POW)

工作量证明机制由中本聪于 2009 年提出[11], 比特币中, 全网节点通过 POW 达成对交易池中待确认交易的共识过程. P2P 网络中所有对等节点通过完成工作量证明机制竞争记账权, 只有当某个节点完成共识过程并提出当前阶段区块数据后, 全网节点才可以继续尝试完成工作量证明, 提出新区块.

工作量证明通常包含 3 个阶段: 第 1 步, 生成新区块挖矿难度 (Difficulty); 第 2 步, 采用挖矿算法尝试解决难度; 最后, 比对难度目标, 验证难度是否被满足. 工作量证明的本质是利用节点算力来解决数学难题, 在比特币系统中是通过找到一个满足条件的 Hash 值来证明节点的工作量. 具体步骤描述如下:

Step 1. 动态调整难度生成算法, 生成当前挖矿难度 Difficulty;

Step 2. 采用挖矿算法计算当前阶段区块的 Hash 值, 通过改变难度目标 (Nonce) 的方式改变 Hash 值结果, 直到新区块哈希值小于难度 Difficulty;

Step 3. 验证算法校验新区块哈希值是否小于难度直至满足, 将新的区块数据写入本地区块链. 开始下一轮共识过程.

比特币中通过工作量证明机制实现交易在全网节点中的共识过程. 当区块链较长时, 除了链条尾部少数区块外, 其余区块均以得到确认, 实现了一致性. 同时, 全网节点可以自由加入区块链, 且单节点操作不会对区块链一致性造成影响. 每个节点通过消耗算力完成工作量证明机制, 保证了攻击者无法通过创建多身份标识的方式削弱备份数据效力, 可以有效抵御 Sybil[12]. 在诚实节点的算力占多数的情况下, 可以有效解决数字货币的“双花”问题, 保证全网数据安全.

在共识效率方面, 工作量证明机制存在一些问题: 首先, 共识效率较低. 区块链的生成需要消耗时间, 每个新区块需要被后续区块确认后才能被认定有效, 这个过程需要消耗更多的时间, 影响了工作量证明机制的共识效率. 例如在比特币网络中, 系统动态调整难度目标, 保证系统每隔 10 分钟生成一个区块, 且新区块需要被后续 6 个区块确认. 在比特币系统中, 完成一笔交易要等待至少 60 分钟才可能达成共识. 其次, 在数据安全方面存在隐患: 为保证区块数据的安全, POW 算法要求恶意节点的算力不能超过全网算力的 50%, 然而目前比特币矿池中算力排名前列的矿池总算力所占比例已经过半[13], 理论上可以完成对整体比特币网络的控制, 对公平性与安全性造成影响. 第三, 算力对资源的损耗. POW 基于算力, 造成资源与能源的大量消耗, 即使后来提出的有用的工作量证明机制 (Proof Of Useful Work, POUW)[14], 仍无法解决效率等问题.

(2) 实用拜占庭容错算法 (PBFT)

实用拜占庭容错算法 (Practical Byzantine Fault Tolerance, PBFT) 源于拜占庭协定, 主要是指如何在异步模型下, 多节点分布式系统中可以容忍一定错误 (即不确定行为的未知节点) 并保持正常工作实现共识. 假设系统中存在恶意节点 f 个, PBFT 可以实现节点数不小于 $3f+1$ 个的分布式系统的分布式一致性. 相比于 BFT (Byzantine Fault Tolerance, 拜占庭容错算法), PBFT 更高效的达成了一致性, 避免资源浪费; 且在一个阶段, 仅有一个节点提出新区块, 剩余节点则进行区块数据的校验工作, 可以有效提高交易与区块的共识效率.

然而, PBFT 在网络节点扩展性方面存在一定缺陷. 算法的共识效率依赖于当前区块链网络的节点个数, 当节点数量增加时, 数据的广播与确认周期变长, 导致共识效率降低. 因此, 该协议不适用于节点数量较大的区块链系统, 扩展性差. 此外, 一轮是否可以取得共识也依赖于主节点是否诚实, 若主节点提出无效区块, 则本轮不会产生区块, 影响效率[15].

2 国密链

2.1 架构模型

“国密链”架构模型分为 4 个逻辑层次: 应用层、接口层、区块链服务层与存储层, 架构模型如图 5 所示.

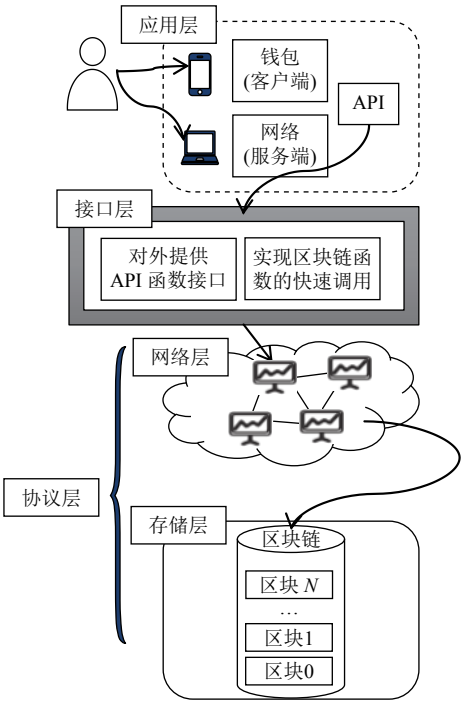

图 5 国密链架构模型

应用层: 应用层主要为去中心化应用程序, 例如版权保护系统、数据存证系统.

接口层: 接口层为上层应用提供 API, 用于去中心化应用实现如数据上链、查询等业务逻辑.

区块链服务层: 区块链服务层主要包括数据存储、网络共识、账户管理等模块. 数据存储模块实现从交易池中读取交易, 创建并向其他节点发送区块等功能; 网络共识模块实现全网新区块共识、接收并存储缺失区块等功能; 账户管理实现账户的创建与保存,

实现交易的签名与验签等功能.

存储层: 存储层主要包括数据缓存与数据库服务. 缓存操作用于缓存内存中的临时信息, 包括从用户接收的交易信息、支持系统运行的临时数据等; 数据库操作是以非关系型数据库为基础实现的数据存储.

2.2 算法设计

2.2.1 SM3

“国密链”中, SM3 算法应用于 Hash 值与 Merkle 根的计算. 区块头由 3 组区块元数据组成, 数据结构如表 2 所示. 首先是一组引用前一区块哈希值的数据, 这组元数据用于将该区块与区块链中前一区块相连接. 第 2 组元数据是 Merkle 树根, 用以记录当前区块交易信息. 第 3 组元数据, 即难度、时间戳和 nonce, 用以共识算法为 POW 时的挖矿操作.

表 2 区块头结构

字段	大小(B)	描述
版本	4	版本号, 用于跟踪软件/协议的更新
父区块哈希值	32	引用区块链中父区块的哈希值
Merkle根	32	该区块中交易的Merkle树根的哈希值
时间戳	4	该区块产生的近似时间(精确到秒的Unix时间戳)
难度目标	4	该区块工作量证明算法的难度目标
计数器	4	用于工作量证明算法的计数器

表 2 中前两组元数据为通用数据结构, 第 3 组与挖矿有关的元数据的必要性依赖于共识算法. 例如当前系统采用 PBFT 算法, 难度目标、随机数数据项与共识过程没有联系, 则由系统随机填充数据.

(1) 区块 Hash 值

“国密链”中, 通过 SM3 算法对区块头数据进行二次哈希计算而得到 32 字节的区块哈希值, 用以唯一标识一个区块.

(2) Merkle 根

区块头中的 Merkle 根由区块体中所有交易的哈希值生成. 如图 6 所示: 从下往上, 两两成对, 连接两个节点哈希, 将组合哈希作为新的哈希, 重复该过程, 直到仅有一个根哈希, 则被作为整个区块交易的标识, 保存到区块头.

2.2.2 SM2

“国密链”中, SM2 算法应用于交易的签名、验签操作. 交易是国密链系统中最重要的部分, 系统中任何其他的部分都是为了确保交易可以被生成、能在国密链网络中得以传播和通过验证, 并最终添加入全网交易总账簿.

国密链交易的本质是数据结构, 主要包含交易哈希、交易数据、交易签名等 3 部分内容, 如表 3 所示.

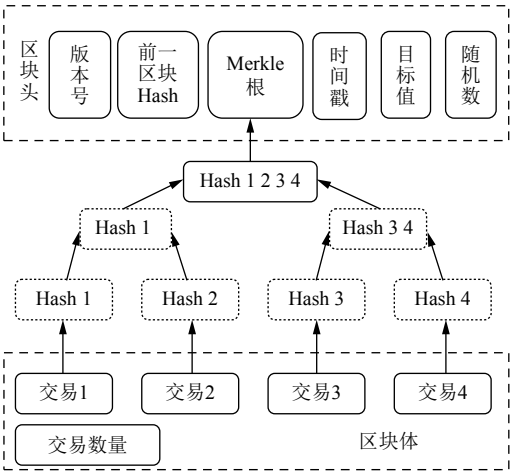

图 6 生成 Merkle 根

表 3 交易结构

字段	大小(B)	描述
交易哈希	32	随机生成的交易标识
交易数据	可变	欲链上存储数据
交易签名	64	SM2算法生成的交易签名

交易签名验签流程:

设节点 A 创建一笔交易 T_A, 包含交易数据 D.

Step 1. 节点注册

系统首先通过 SM2 算法生成节点 A 的私钥 d_A 和公钥 p_A, 计算用户 A 的哈希值 Z_A, 将 d_A、Z_A 加密存储于本地.

Step 2. 交易签名

对数据 D 进行预处理, 得到数据 $\bar{D}$. 通过 SM3 算法对数据 $\bar{D}$ 进行散列运算得到其散列值 e. 利用 SM2 中的签名算法对交易数据散列值 e 进行签名, 得到交易的数字签名 $SIG_T=(r,s)$. 将公钥 p_A、签名 SIG_T 与交易绑定, 得到交易 $T_A=\{T_A,p_A,SIG_T\}$, 向全网广播.

Step 3. 交易验证

具有生成区块权利的节点, 即公有链中的挖矿节点、联盟链中的主节点, 负责收集交易池中未确认的交易, 通过 SM2 验签算法校验交易中绑定的用户公钥 p_A 与交易签名 SIG_T 合法性. 若验证通过, 则认定交易 T_A 合法, 并将其存储至新区块中.

2.3 可插拔共识

“国密链”模块化交易共识过程, 通过自定义接口的方式, 对外提供交易处理与共识模块的方法调用, 以

达到“可插拔共识”的目的. 其中共识算法选取与相关共识操作均由使用者完成, 实现了共识算法的灵活可变.

2.3.1 接口定义

接口类定义如下:

(1) Consensus: 共识算法接口类

```
class Consensus {
public:
virtual void Create(Transaction transaction ) = 0;
virtual void Broadcast(Transaction transaction ) = 0;
};
```

方法介绍:

Create() 方法负责创建一笔交易;

Broadcast() 方法在交易创建后负责将交易向全网广播.

(2) Manage: 交易管理类

```
class Manage {
public:
Manage() {};
~Manage() {};
vector<CTransaction> TransactionPool();
void SetConsensus (Consensus consensus);
void Start();
void Inject(Transaction transaction);
void Halt();
}
```

方法介绍:

TransPool() 方法会返回一个 vector 容器用以存储交易;

SetConsensus () 方法负责设置共识对象;

Start() 方法负责启动 Manager 线程;

Inject (Transaction transaction) 方法会在 Transaction-Pool() 收到事件之后将事件发放给 Dispose进行处理;

Halt() 方法负责停止 Manager 线程.

2.3.2 可插拔共识协议

“国密链”将交易共识过程拆分为交易管理、交易共识两部分. 其中共识算接口类 Consensus 实现了共识的基本逻辑与相关流程, 并对外提供共识方法接口; 交易管理类 Manage 调用接口, 实现交易的共识操作.

Step 1. 使用者首先继承接口类 Consensus 实现共识算法, 调用 NewConsenter 方法初始化共识引擎, NewConsenter 可插拔;

Step 2. 交易 T_A 被账户 A 创建, 由交易管理类 Manage 调用方法 SetConsensus 指定交易实际处理对象 Consensus, SetConsensus 可插拔;

即所有与共识有关的参数传递全部可插拔, 并且对外提供了相应接口. 使用者通过自定义 Consensus 初始化共识算法, 在交易处理时 SetConsensus, 即可实现“可插拔共识”.

以 PBFT 算法为例, 演示“可插拔共识”的实现过程. 整体实现可插拔共识分为两个阶段: 共识算法预准备 (Pre-Consensus)、交易共识 (Tra-Consensus).

设系统网络结构如图 7 所示, 其中, N_0 为主节点, 负责建块相关操作; N_1、N_2 为分布式节点, 负责交易验证; N_3 为宕机节点, 不参与共识过程.

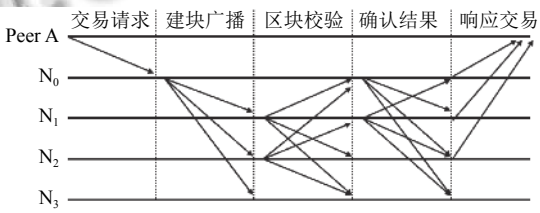

图 7 系统网络结构

(1) Pre-Consensus:

① 调用共识算法接口 Consensus, 实现 PBFT 的共识逻辑.

Order(): 排序服务函数, 实现待确认交易排序服务;

Commit(): 区块确认函数, 实现新区块数据确认过程;

StateUpdate(): 区块状态更新函数, 新区块校验通过时, 实现新区块上链操作.

② 调用 NewConsenter 方法初始化共识引擎.

(2) Tra-Consensus:

① 全网选出一个主节点 N_0, 负责新区块的生成.

② 每个节点将交易向全网广播, 主节点调用 Order() 函数将收集到的待确认交易进行排序后存入列表, 并将该列表向全网广播, $N_0 \rightarrow N_1/N_2/N_3$.

N_1、N_2 接收到交易列表后, 调用 SM2 算法验证交易签名. 所有交易验证完毕后, 基于交易结果计算新区块散列值, 向全网广播, $N_1 \rightarrow N_0$、N_2、N_3; $N_2 \rightarrow N_0$、N_1、N_3. N_3 作为拜占庭节点不负责交易验证与广播.

④ N_1、N_2 分别接收到其它节点发来的区块散列值, 调用 Commit() 函数将该散列值与本地结果进行校验, 并记录校验通过次数, 若次数大于 $2f$ (f 为可容错的拜占庭节点数,此时 f=1) 个,则向全网广播一条 Commit 消息.

⑤ 若 N_1、N_2 分别接收到大于 $2f$ 条 Commit 信息, 则调用 StateUpdate() 函数将新区块追加到本地区块链

和状态数据库, 完成共识过程.

3 性能分析

3.1 SM2/SM3

“国密链”中, 密码算法主要应用于交易的签名验签, 且 SM2、ECC 也分别采用 SM3、SHA-256 进行哈希运算, 因此本文将对 SM2、ECC 设计测试实验, 以比较国密链相较于普通区块链架构的优势.

3.1.1 差异分析

(1) SM3 与 SHA-256

SM3 与 SHA-256 都是哈希算法, 具有类似的结构与流程. 例如: 两者数据最大输入均为 264 位, 输出为 256 位的哈希值; 消息分组均为 512 位, 每个分组采用 64 轮循环迭代, 且压缩函数具有相同的结构, 计算流程基本一致. SM3 的优势在于将消息扩展为 132 个字, 相较于 SHA-256 算法将消息扩展为 64 个字, 位之间相关性更强, 拥有更优良的抗碰撞性能.

(2) SM2 与 ECC-256

SM2 是 ECC 椭圆曲线算法的一种, 其私钥长度为 256 位. 主要分为签名验签、加密解密、密钥交换 3 个功能模块. 表 4 显示了 SM2 与 ECC-256 签名算法的不同之处.

表 4 SM2 与 ECC-256 算法差异

处理内容	SM2	ECC
消息预处理	$\overline{M}=Z_A\|M$	$\overline{M}=M$
哈希算法	SM3	MD5、SHA-1、SHA-2
签名计算	$r=(e+x_1)\bmod n$ $s=((1+d_A)^{-1}(k-rd_A))\bmod n$	$r=(x_1)\bmod n$ $s=k^{-1}(e+rd_A)\bmod n$

可以看出, SM2 与 ECC-256 的签名算法主要差别是计算消息 M 的哈希值e时, 需要额外计算一个 Z_A 值. Z_A 长度为 256 位, 是根据签名者自己的 ID、椭圆曲线参数、自身公钥 p_A 等信息的哈希值, 使得签名拥有更全面的用户信息, 签名值拥有更强的效力, 最终提高签名的可信性.

3.1.2 性能测试

(1) 测试环境

两种算法使用基于素数域的椭圆曲线, 在素数域 Fp 上的方程为:

$$y^2=x^3+ax+b \tag{3}$$

其中, SM2 采用选用国密推荐标准, 参数如下:

a =0xFFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF00000000FFFFFFFFFFFFFFFC

b =0x28E9FA9E9D9F5E344D5A9E4BCF6509A7F39789F515AB8F92DDBCBD414D940E93

ECC 采用 seck256k1 曲线, 参数如下:

a =0x0000000000000000000000000000000000000000000000000000000000000000

b =0x0000000000000000000000000000000000000000000000000000000000000007

(2) 测试方法

选取长度为 32、64、128 字节的字符串, 分别通过 SM2、ECC 算法完成签名验签操作, 重复 10 000 次, 计算签名验签的时间开销与内存开销.

(3) 测试结果

① 时间开销

由表 5 可以看出, SM2 签名验签算法具有更快的执行速度, 相较于比特币中采用的 secp256k1 曲线, 时间节约了大概 20%.

② 内存开销

由表 6 可知, SM2+SM3 实现签名验签比 ECC-256+SHA-256 算法占用的内存更少, 节约内存大概 10%.

表 5 SM2 和 ECC-256 签名验签时间开销 (单位: s)

算法类型	签名时间	验签时间
SM2	981	1975
ECC-256	1145	2321

表 6 SM2 和 ECC-256 签名验签内存开销 (单位: Byte)

算法类型	ROM	RAM
SM2	10 895	3015
ECC-256	10 157	3202

3.2 可插拔共识

“国密链”设计的可插拔共识协议, 支持不同共识机制. 本节从以下 3 个维度分析可插拔共识协议:

(1) 安全性

“可插拔共识”协议以纯分布式 P2P 作为基本网络架构, 即新节点随机与某个节点建立连接. 通过这种随机拓扑的方式, 与普通区块链结构中的结构化 P2P 网络相比, 拥有以下优点:

① 使用泛洪的方式通信, 节点不依赖分布式哈希表等节点发现机制查找对等节点, 节点可以任意的加

入或退出国密链网络, 使得网络可扩展性强;

② 节点在加入或退出国密链网络时只会对临近节点造成影响, 对整体网络影响不大, 使得网络更稳定;

③ 可以有效避免集中式网络单点性能瓶颈和单点故障的问题, 使得网络鲁棒性更强;

④ 为了完成交易的共识过程, 国密链中账户地址、交易详情等隐私数据需要保证一定的公开性. 通过随机拓扑的方式可以有效地弱化账户地址与实际网络地址之间的联系, 可以有效抵御基于节点数据的分析溯源攻击, 使得国密链中账户数据更为安全.

(2) 扩展性

“可插拔共识”协议定义了通用共识接口, 使国密链架构支持共识算法的扩展.

当前典型的联盟链、公有链架构, 如超级账本、比特币, 分别采用实用拜占庭容错算法 (PBFT)、工作量证明 (POW) 作为唯一共识机制, 实现交易的共识操作. 由于二者均不具备改变共识机制的条件, 当系统应用范围从最开始的少量用户节点发展至大量用户节点时, 必须分别选择以上两种共识机制, 即需要两倍的开发工作量来满足当前的系统需求.

不同于普通区块链架构中共识算法不可更改的情况, 当系统应用范围扩大时, 基于国密链的应用开发者可重新设计并实现满足需求的共识算法, 实现共识的灵活应用.

(3) 性能效率

“可插拔共识”协议提供共识接口可以简洁高效地实现不同的共识算法, 以满足不同应用环境的性能需求.

国密链的共识效率取决于开发者选取的共识算法, 共识过程最关键的步骤是 P2P 节点对新区块交易的校验, 主要操作为验证交易的签名是否合法. 与普通的区块链架构采用 ECC-256+SHA-256 椭圆曲线密码算法实现交易签名验签不同, 国密链采用更安全高效的国密 SM2+SM3 算法. 上节实验表明, 基于国密算法的国密链架构相较于普通区块链架构, 在占用内存更少的情况下, 拥有更快的签名验签速度, 间接提高了共识效率.

4 结论

本文以国密算法 SM2、SM3 为基础, 设计并实现了“国密链”架构, 使区块链技术更为自主可控. 结合“可插拔共识”协议, 有效解决了区块链架构面临的共识算法妥协问题. 经过分析, “国密链”架构相较于普通区块链架构, 在交易签名验签方面拥有更低的时间开销与内存开销, 而且增加了共识算法的可扩展性, 实现了共识算法的灵活应用.

参考文献

1 吴健, 高力, 朱静宁. 基于区块链技术的数字版权保护. 广播电视信息, 2016, (7): 60–62. [doi: 10.3969/j.issn.1007-1997.2016.07.022]

2 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. 自动化学报, 2017, 43(9): 1555–1562.

3 王海龙, 田有亮, 尹鑫. 基于区块链的大数据确权方案. 计算机科学, 2018, 45(2): 15–19, 24. [doi: 10.11896/j.issn.1002-137X.2018.02.003]

4 张永, 李晓辉. 一种改进的区块链共识机制的研究与实现. 电子设计工程, 2018, 26(1): 38–42, 47. [doi: 10.3969/j.issn.1674-6236.2018.01.008]

5 Castro M, Liskov B. Practical Byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems, 2002, 20(4): 398 –461. [doi: 10.1145/571637.571640]

6 不止思考. 揭秘区块链的核心技术之 [哈希与加密算法]. https://www.cnblogs.com/jsjwk/p/9476175.html. (2018-08-14).

7 任子荣. 基于移动终端和 PKI 技术的第三方身份认证服务系统 [硕士学位论文]. 上海: 上海交通大学, 2014.

8 申延召. SM3 密码杂凑算法分析 [硕士学位论文]. 上海: 东华大学, 2013.

9 胡卫, 吴邱涵, 刘胜利, 等. 基于国密算法和区块链的移动端安全 eID 及认证协议设计. 信息网络安全, 2018, (7): 7–15. [doi: 10.3969/j.issn.1671-1122.2018.07.002]

10 工业和信息化部信息中心. 2018 中国区块链产业白皮书. http://www.miit.gov.cn/n1146290/n1146402/n1146445/c6180238/part/6180297.pdf. (2018-05).

11 Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/en/bitcoin-paper. 2009.

12 Douceur JR. The Sybil attack. Proceedings of the 1st International Workshop on Peer-to-Peer Systems. Cambridge, UK. 2002. 251–260.

13 区块. 全球算力分布. https://www.walian.cn/zixun/news/522.html. (2018-04-28).

14 Ball M, Rosen A, Sabin M, *et al*. Proofs of useful work. http://eprint.iacr.org/2017/203.pdf. (2017-02-27).

15 韩璇, 刘亚敏. 区块链技术中的共识机制研究. 信息网络安全, 2017, (9): 147–152. [doi: 10.3969/j.issn.1671-1122.2017.09.034]