DOI:10.19729/j.cnki.1009-6868.2018.06.008
网络出版地址:http://kns.cnki.net/kcms/detail/34.1228.TN.20181127.0926.002.html

一种基于区块链的身份识别技术

An Authentication Technology Based on Blockchain

苏宣瑞/SU Xuanrui
邹秀清/ZOU Xiuqing
丁勇/DING Yong
(桂林电子科技大学,广西 桂林 541004)
(Guilin University of Electronic Technology, Guilin 541004, China)

中图分类号:TN929.5 文献标志码:A 文章编号:1009-6868 (2018) 06-0041-008

摘要: 基于以太坊的智能合约技术提出一种新型身份识别系统的设计方案,其中平台架构设计分为数据层、网络层、共识层和接口层,采用点对点(P2P)技术将数据分布存储在各个节点,实现了分布式认证,可抵抗黑客的攻击;采用椭圆曲线数字签名算法(ECDSA)和改进的Merkle树,确保数据的真实性。本设计具有高拓展性、高可靠性、高安全性的特点,支持不同平台进行统一的身份认证。

关键词: 区块链;以太坊;智能合约;身份识别

Abstract: A new authentication system design based on Ethereum smart contract technology is proposed. The platform architecture includes a data layer, a network layer, a consensus layer, and an interface layer. The point to point (P2P) technology is used to distribute data to each node, then a distributed authentication mode can be implemented, and hacker attacks can be resisted; the elliptic curve digital signature algorithm (ECDSA) and the improved Merkel tree are applied to ensure the authenticity of the data. This design is highly scalable, high reliability and high security, and supports unified authentication on different platforms.

Keywords: blockchain; Ethereum; smart contract; authentication

随着互联网行业的发展,越来越多新型的网络平台融入到了人们的生活,人们日常生活都要用到淘宝、京东等交易平台,使用支付宝、小米钱包、微信等来支付。这些平台都有一套独立的注册、登录、认证和权限管理的系统,每一个用户在系统中都充当不同的角色,并拥有不同的权限,这种中心化系统给人们带来的弊端有以下几点:

(1)如果有多个系统需要进行登录、认证,管理员的维护和管理的成本将会大幅增加,并随着规模的增加,维护难度会增加。

(2)用户必须记住自己的多个账号、密码,如果账号很多,使用起来将非常不便捷。

(3)容易被黑客攻击,带来损失。

(4)中心化系统不可信,管理员可以随时篡改数据。

区块链技术是一种利用去中心化共识的机制维护一个完整的、分布式的、不可篡改的账本数据库的技术,它能够让区块链中的参与者在无需建立信任关系的前提下实现一个统一的账本系统。近年来,区块链以集成分布式数据存储、点对点(P2P)传输、新型加密算法和共识机制等技术的特点,已越来越成为许多国家政府和国际组织研究讨论的热点,依靠互联网的产业也纷纷加大了对其投入的力度[1],但是目前全球还没有政府大力推广将该技术应用于物联网的身份识别系统。

如今新型的区块链技术给人们带来了解决方案:区块链建立了动态的P2P网络,没有了中心化服务,帐本均分布在每个节点中,所有的节点一同维护;帐本上记录了该区块链自创建以来的记下的所有交易记录,通过密码学的安全机制,使得所有记录不可修改、真实可信;每个人都是一个节点,通过彼此之间的信任来建立区块链的信任。区块链网络没有传统的中心管理员,整个网络的运作由线上的电脑共同进行维护,使得运营成本大幅降低。

本设计组成的框架主要包括四大模块:数据层模块、网络层模块、共识层模块和接口层模块。通过接入到同一个区块链网络中,使用统一的接口层进行交互,同时接口层还能和网络层和共识层通过底层协议进行交互,网络层负责发现区块链网络中的P2P节点和数据的传输,共识层负责身份认证,数据层负责存储数据。

1 关键技术

1.1 区块链技术

1.1.1 区块链技术基本原理

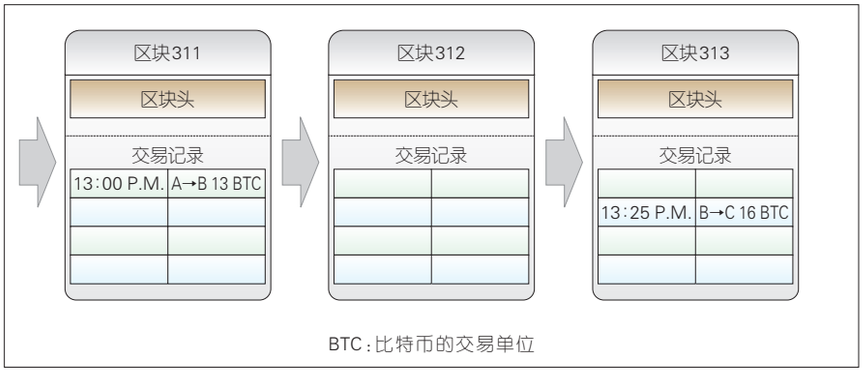
假设Bob要在互联网上向Alice转

收稿日期:2018-10-30
网络出版日期:2018-11-27

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P1

账，每次转账都会产生交易记录，将所有的交易记录进行连接，生成总帐单，总账单包含每个人的余额。记账时，应需保持公平、诚信的态度，使得双方能够相互信任；但记账人可能会作假，使得双方的信任程度降低，这是很典型的欺诈行为。区块链技术则可以很好地解决这一问题，没有人可以作假。

现把账单模型进行缩小，规定：每次更新、修改数据必须在原有的账单中进行，并且新账单包含时间戳、前一个账单的哈希值等数据。将这些帐本累加起来称为总帐本，总帐本将所有的块链接起来，组成区块链，图1表示一条区块链中的3个区块。



▲图1 一条区块链中的3个区块

1.1.2 区块链的去中心化

在2008年的金融危机中，比特币的创始人中本聪发明了比特币，它成为了第一个去中心化的数字货币。区块链是一个分布式网络，每个节点都会存放所有交易的副本，并自动同步。节点可以是用户的电脑、手机，或是其他设备。如图2所示，区块链网络节点是扁平化的，每个节点的地位相等、公平，并以扁平拓扑的方式向相邻节点进行数据交互[2]。

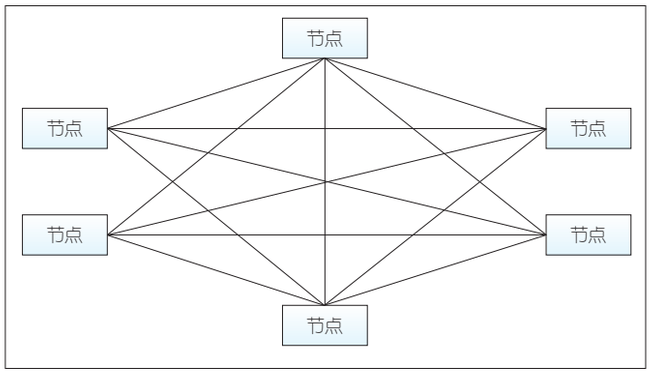


图2▶ 去中心化的节点图

假设Bob想给他人转账，Bob就得向全网广播他要转账的消息，并需全网达成共识，才能认为他的消息是合法的，且每个节点都会保存他转账信息。全网没有中心服务器，没有人能拥有管理的权力，只要规则定好了，就必须照着规则做，没有人可以改变，这其实就是区块链去中心化的魅力所在。

所有节点都相当于“校验员”，它们无时无刻不在检查区块中的交易信息是否正确，并且在检查交易的时候，不断尝试产生随机数，计算哈希值，使数据具有很强的安全性，黑客无法入侵，无法修改账户余额。随着用户的增加，越来越多的后续节点（用户）加入到了比特币网络中，共同完成共识的过程[3]；而整个过程中网络中每个节点的地位是相同的。比特币不是凭空产生，而是通过消耗了电力、物力并应运算而产生的，因此具有价值，可以兑换成现有的货币。

1.2 加密哈希函数和哈希指针

哈希函数也叫做哈希算法，不论输入字符串的长短，生成签名的长度都是固定的，因此可以作为一段数据的数字指纹，便于区分每个消息。生成的摘要可作为签名，以确保数据的真实性。账户的创建需要一个非对称加密密钥对，以太坊选择的是椭圆曲线加密算法（ECC）中的Secp256k1，依据速度、安全性等参数确定账户的地址，具体的方法如下：

（1）创建一个随机私钥，由64个十六进制的字符构成；

（2）从私钥中导出公钥；

（3）从公钥中导出地址。

使用Secp256k1生成256位公钥/密钥，然后编译成64位长度的十六进制字符串；采用公钥的Keccak-256哈希算法，得到一个32字节的字十六进制字符串，接下来对该字符串进行取截，取字符串的最后20个字节（即删除前12个字节），得到了40个字节的字符串，在签名加上0x前缀，就可以得到一个42个字节长的地址，该地址就是以太坊用户全网唯一的账号。

1.3 Merkle树

Merkle树在区块链中尤其重要，相当于用大量的数据块来进行哈希运算。Merkle树将2个相邻的认证请求进行哈希计算，逐步堆积到Markle树根。这种哈希算法的好处就是让历史数据不可篡改、真实可信[4]。节点的值是它相连2个叶子节点的哈希，这就导致整个Merkle树中的数据都是互相关联的，改动其中一个数据，将会彻底改变整个区块的结构，因此给身份的证明提供了一个非常简洁的机制。

Merkle树的最初应用是在比特币中，即使用了Merkle树来存储每个区

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P2

块的交易。每个Merkle树从块到根都是由哈希的分支组成，如图3所示。由于Merkle树采用了非常强的哈希算法，且哈希后的摘要求逆几乎不可能实现，因此Merkle树提供了真实可信的数据验证方法。

每个区块头包含如图4所示的内容。将相连的数据区块的数据进行相连，通过上一区块的哈希值和当前区块的哈希值将所有的区块请求进行关联。如果修改了其中一个数据，将影响所有在当前区块链网络上的区块，因此数据不可能被篡改，所有的认证请求不可伪造，极大提高了区块链的安全性。

1.4 ECDSA

ECDSA是数字签名算法（DSA）的其中一个例子。和非对称加密算法（RSA）进行对比，在相同的安全强度下，ECDSA可以使用的密钥更短，从而节省网络和存储空间，具有较高的研究价值[5]。

在本设计方案中，首先要避免数据明文传输的极大不安全因素，同时要保证交互双方的身份真实性，因此需要利用公钥加密算法中非对称加密的优势。使用本设计方案进行数据传输时，将服务器的公钥输出在客户端，客户端使用公钥加密，在信息交互时数据以密文方式传给服务器端，再由相应私钥得到明文数据[6]。

Alice将要给Bob发送一条消息，要求消息包含数字签名来进行身份识别，那么可以定义一组参数$(CURVE,G,n)$，其中$CURVE$表示椭圆曲线的点域以及它所使用的几何方程，G表示椭圆曲线基点，大素数n是椭圆曲线的阶数[7]。接下来我们介绍数字签名的具体过程和验证数字签名的具体过程。

（1）数字签名的过程

如果Alice要发出认证请求，她希望能对消息m进行签名，因此将椭圆曲线的参数设计为$D=(p,a,b,G,n,h)$，其中对应的密钥对为(k,Q)，Q为公钥，k为私钥。Alice将按照如下步骤进行签名：

1）产生一个随机数$d,1\leqslant d\leqslant n-1$；

2）计算$dG=(x_1,y_1)$，将x_1转化为整数$\overline{x_1}$；

3）计算$r=\overline{x_1}\bmod n$，若$r=0$，则转向第1步；

4）计算$d^{-1}\bmod n$；

5）计算哈希值$H(m)$，并将得到的比特串转化为整数e；

6）计算$s=d^{-1}(e+kr)\bmod n$，若$s=0$，则转向第1步；

7）(r,s)即为Alice对消息m进行的签名。

（2）验证数字签名的过程

如果Bob收到消息m之后，他需要验证消息m的签名(r,s)，在得到椭圆曲线参数和Q之后，将按以下步骤操作来验证数字签名[8]：

1）首先验证r和s是区间$[1,n-1]$上的整数；

2）计算$H(m)$并将其进行转化变为整数e；

3）计算$w=s^{-1}\bmod n$；

4）计算$u_1=ew\bmod n$以及$u_2=rw\bmod n$；

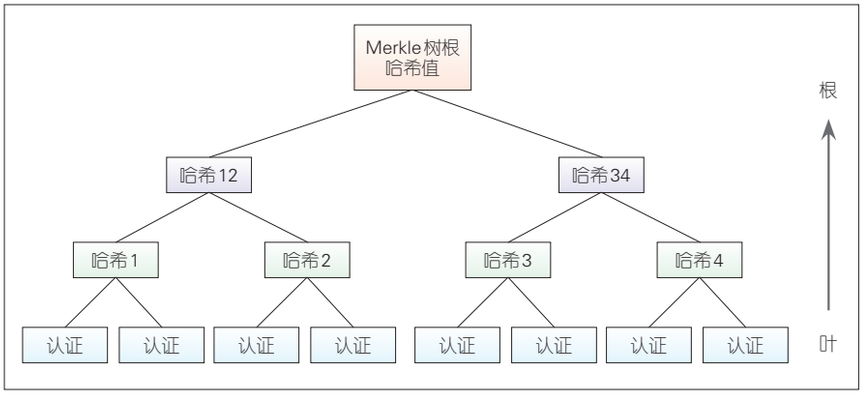
5）计算$X=u_1G+u_2Q$；

6）若$X=0$，则拒绝该签名的有效性，否则将X的x坐标x_1转化为整数$\overline{x_1}$，并计算$v=\overline{x_1}\bmod n$；

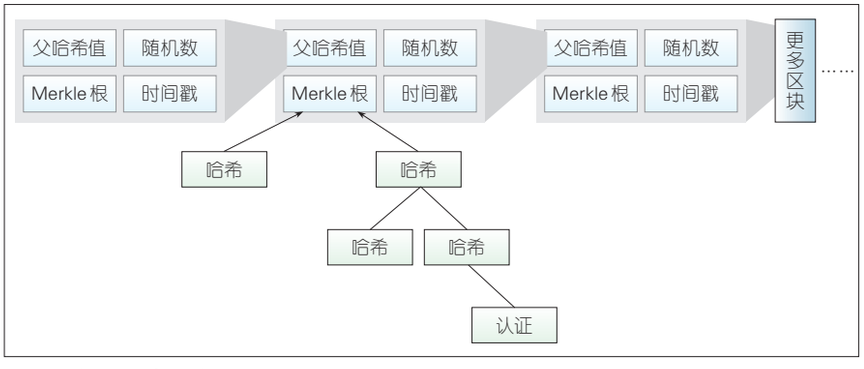
7）当且仅当$v=r$时，签名验证可以通过。

利用ECDSA算法，将认证信息进行数字签名，确保了每条认证消息都是由正确的用户发表的，防止他人假冒，还可以保证数据的完整性。在整个认证请求中，将不会有人对数据包进行恶意篡改。

每个节点用户再发起认证请求时，都会利用自己的私钥签名，其他节点收到认证请求，也会一同参与签名认证的操作。如果认证成功，将会将账单记录下来，完成认证；否则将



▲图3 Merkle树的结构



▲图4 数据区块头的结构

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P3

拒绝认证请求。

1.5 共识方法

1.5.1 Gas——以太坊系统计算工作量的单位

Gas是以太坊系统中执行交易所需要的计算工作量单位。所有的交易不论是转账交易，还是执行智能合约，都要消耗Gas。Gas的价格由交易的发起人和矿工的工作量决定，交易打包进区块中需要矿工们进行哈希运算，矿工们付出了劳动，因此需要收取一定的费用。如果交易发起人设置的Gas价格过低，矿工们基本不会将交易打包进区块里；如果交易设定较高的Gas，该交易将会得到较高的优先级。

1.5.2 以太坊的工作量证明和挖矿原理

工作量证明（PoW）的目的是阻止网络攻击，如当今网络环境下常出现的分布式拒绝服务攻击（DDoS），就是用来发送许多假的请求以耗尽计算机网络系统资源，导致服务器宕机，真正的用户则无法登录到中心服务器上[9]。

PoW被定义为花费计算机算力来进行数据校对的要求，俗称“挖矿”。挖矿的目的有以下几点：

（1）验证交易的合法性，避免出现多重交易的情况；

（2）用来奖励矿工所做出的计算工作；

（3）维护以太坊系统的正常安全运转。

通过挖矿的方式解决PoW的数学难题具有不可逆的特征。从技术角度来说，挖矿的过程就是一个不断进行的哈希运算过程，它通过尝试产生随机数，找到满足条件的随机数后立即将区块进行打包并全网广播，找到该随机数的节点也是赢得本轮记账权利的节点。该区块将在整个区块链网络广播，进行共识的达成。如果达成共识，每个节点将会将该区块添加到自己的区块链中，同时该矿工将会得到以太币奖励。

随机数的条件取决于系统设定的难度，例如：要求整个区块加上随机数计算出的哈希值要小于给定的值才算成功；而哈希值的产生没有规律可循，只有算力越高的计算机才能更快得到符合条件的随机数。

2 核心框架

2.1 数据层

2.1.1 数据区块和链式结构

首先，区块是以太坊网络的核心，所有的交易、数据存储都是在区块头中进行的。不同的区块头之间通过头指针（ParentHash）函数指向前一个区块的头指针，将它们串联起来，形成单项链表。

区块结构分为区块头和区块的数据部分这2个部分，源码在以太坊的/core/types/block.go中，数据层的函数关系图如图5所示。

其中，区块的结构体定义为：

```
type Block struct
{
//表示一次交易的结构体
  header *Header          //区块头
  transactions Transactions   //交易
}
```

区块头的结构体定义为：

```
type Header struct
{
   ParentHash  common.Hash    //头指针，指向前一个区块
   Number  *big.Int          //代表当前区块的编号
}
```

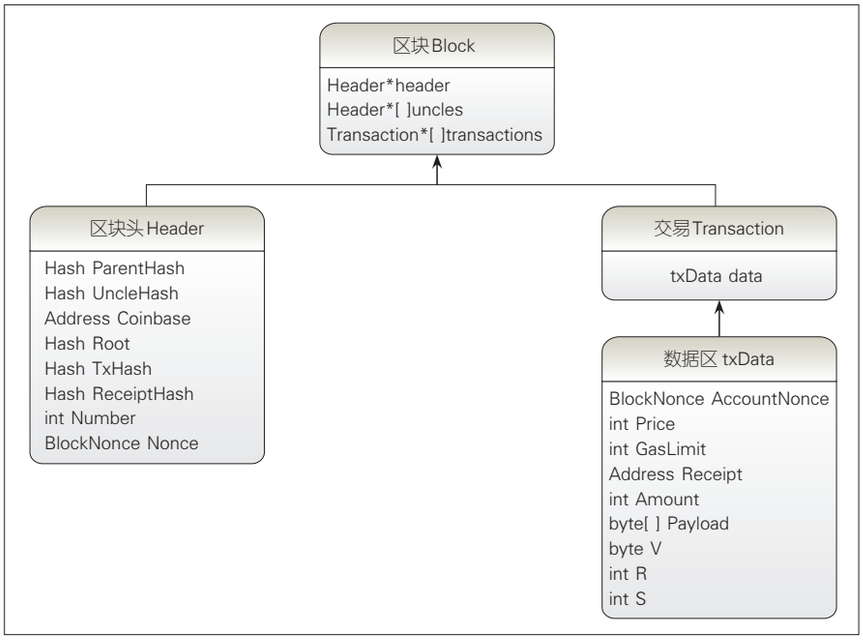
在上面的结构体中提到了交易指针，简称tx，表示一次以太坊交易的结构体。相应的代码在/core/types/transaction.go中。

交易的结构体定义为：

```
type Transaction struct
{
   data  txdata              //交易的存储数据
   Hash, size, from atomic.Value    //在内存使用
}
```

数据区（txdata）的结构体定义为：

```
type txdata struct
```



▲图5 数据层函数关系图

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P4

```
{
    AccountNonce uint64    //随机数
    Price *big.Int         //交易所产生的费用
    GasLimit *big.Int      //当前以太坊网络gas的限制
    Recipient *common.Address    //转账转出方的账户地址
    Amount *big.Int
    Payload []byte
    V, R, S *big.Int       // 数字签名的值
    Hash *common.Hash      // Hash的封装处理
}
```

2.1.2 Merkle树的使用

在以太坊中使用的是Merkle树的改进树(MPT),也是二叉树的一种。节点的值是它相连2个叶子节点值的哈希。Merkle树用于所有交易正确性的验证,而MPT则大大提高了查找效率。树的构造代码在以太坊源码的trie/trie.go中,关键代码如下。

首先是从根节点进行遍历:

```
func (t *Trie) HashRoot(db DatabaseWriter) (node, node, error)
{
    if (t.root == nil) {...}
    //如果节点不存在子节点了,直接返回
    h := newHasher(t.cachegen, t.cachelimit)
    //否则会对节点进行折叠,继续遍历
    return h.Hash(t.root, db, force: true)
}
```

通过调用下面的函数跟父节点进行交互,进行数据存储:

```
func (t *Trie) Commit() (root Hash, error)
{
    if (t.db == nil) {...}  //如果数据已经存在数据,返回
    return t.CommitTo(t.db) //父节点没有数据,就调用下面的函数写数据
}
func (t *Trie) CommitTo(db DatabaseWriter) (root common.Hash, error)
{
    Hash, cached, error := t.HashRoot(db)  //对数据进行Hash,存放到父节点中
    t.root = cached        //把当前遍历的位置存放到内存中
    ...
}
```

查找、插入、删除都是在trie/trie.go里进行使用的。

2.1.3 数据存储的实现

以太坊的数据存放在StateDB中。StateDB是以太坊的数据库,负责本地存储数据及业务,还负责连接到底层的数据库,它使用二级缓存机制来存储账户的相关数据。

StateDB的相关代码在core/state/statedb.go中,其定义的结构体以及作用为:DataBase类型的DB用于存放数据,Tire类型的tire用于存放MPT树,stateObject表示以太坊账户,其中在stateObject中也有二级缓存机制,主要用来缓存和更新以太坊帐户。

整个以太坊网络的运作结构如图6所示。

2.2 网络层

2.2.1 网络层传输协议

当一个节点有新的数据区块产生,该节点将会进行全网广播,其他收到请求的节点将会进行验证。一个节点创建一个新的区块,该新区块很快会被发到网络上所有的节点,然后每个节点都要验证这个新的区块,验证其真实性。经验证后,每个节点才会添加这个新的区块到区块链,区块链网络中的所有节点达成共识,一起决定哪个区块有效而哪个无效,擅自篡改的区块会被网络上其他节点拒绝[10]。

在节点之间传播数据时,采用加密网络和传输协议(RLPx)加密握手协议。该协议在网络层的上层,在以太坊网络中新的节点建立后,首先进行端口监测侦听、节点间连接及通信交互,当节点间都建立了连接,将会通过Msg的格式进行通信。每次在通信的过程中,都会做出如图7所示的判断,以确保握手协议运作正常,如果运作不正常,将会失去对该节点的连接。

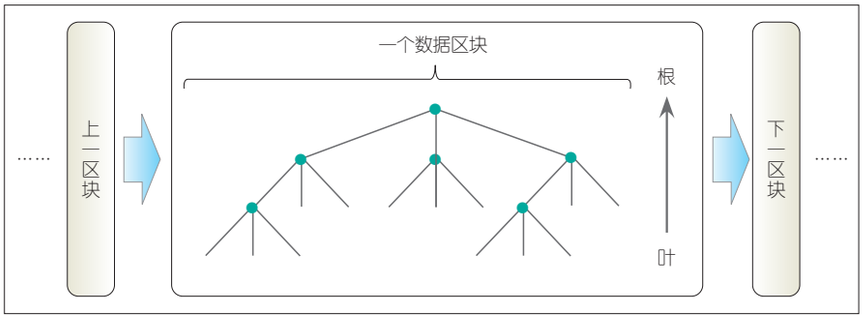
RLPx加密握手协议的具体流程如图8所示。

2.2.2 数据验证机制

P2P网络中的每个节点,只要在线,都在随时监测侦听其他节点的认证请求,验证区块数据的具体一些步骤如下:

(1)负责识别网络中广播的数据和区块;

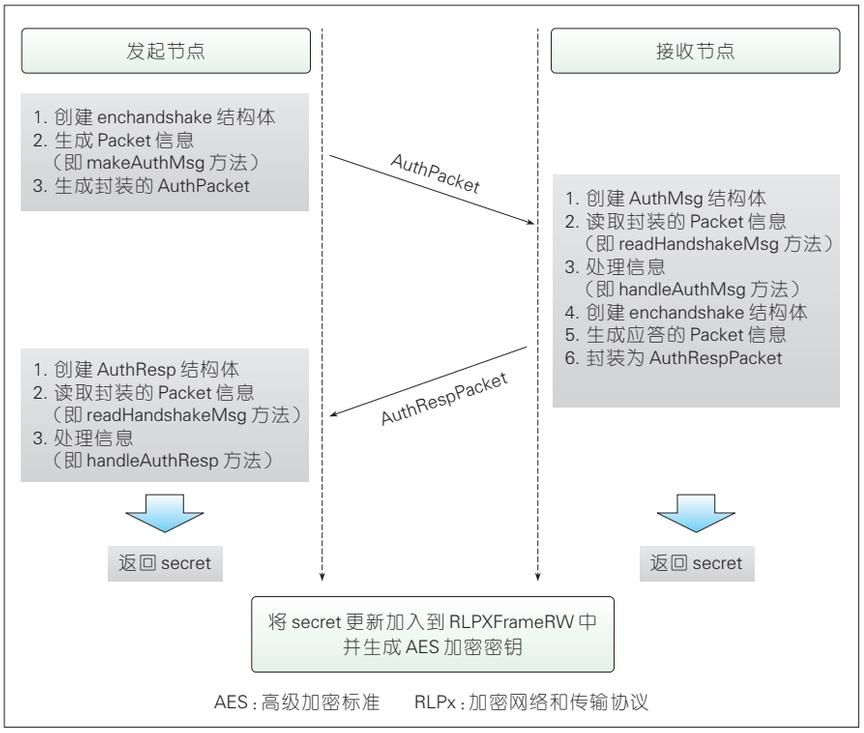
(2)如果接收到相邻节点发来的认证,将会对认证请求进行分析,将检查数据的完整性、语法的规范性、数字签名是否正确等方面来校验交易数据是否有效;



▲图6 以太坊网络的运作结构

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P5

▲图 7 RLPx 加密握手协议的建立



▲图 8 RLPx 加密握手协议的流程

（3）如果数据有效，将会把数据放入存储池中，即将认证请求记录到本地，同时向相邻节点转发数据；

（4）如果数据无效，将会立即放弃该数据，确保数据不会在区块链网络中传播。

2.2.3 P2P 网络的具体实现

以太坊的 P2P 网络主要使用以下几个工具实现。

（1）Discover：使用了 Kademlia 协议，用于使用 UDP 的 P2P 节点发现的协议；

（2）discv5：用于发现新节点；

（3）nat：网络地址转换工具；

（4）netutil：有关网络连接的工具；

（5）simulations：P2P 网络测试工具。

P2P 网络非常复杂，如果要使用它，就必须要包含节点查找、节点维护、节点建立连接的功能。在 database.go 文件中，newNodeDB 函数用来存储节点数据，存储节点数据采用了 Keccak-256 的签名哈希算法。以下是一些关键函数。

（1）查找节点的函数：func(db *nodeDB)node(id NodeID)*Node；

（2）插入数据的函数：func(db *nodeDB)updateNode(node*Node)；

（3）删除数据的函数：func(db *nodeDB)deleteNode(id NodeID)。

连接超时的处理办法，即如果发现有个节点接收消息的时间超出了设定的值，那么就删除节点不再连接，关键代码如下：

```
func (db *nodeDB) expireNodes() error
{
  threshold := time.Now().Add(-nodeDBNodeExpiration)
//将节点最后接收的时间和当前的时间进行比较
  it := db.lvl.NewIterator(nil, nil)
  defer it.Release()  //初始实例化，先将指针指向头部，接下来准备开始遍历
  for it.Next()  //循环查找节点
  {
   ……
   if !bytes.Equal(id[:], db.self[:])  //找到对应的节点
   {
    if seen := db.lastPong(id); seen.After(threshold)
                    //如果节点在规定时间内更新，继续查找
     Continue
    db.deleteNode(id)          //否则释放该节点，不再连接
   }
  return nil                  //返回最新的节点
  }
```

2.3 共识层

2.3.1 PoW 的机制

PoW 通过计算机进行数学运算得到记账权，但是每次要达成全网共识，都需要全网一起参与运算。本设计系统作了以下规定，使认证步骤准

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P6

确进行：

（1）所有连接到以太坊网络的地址都应该分为已被认证的和未被认证的；

（2）已认证的节点可以变成没有经过认证的，没有经过认证的节点也可以变成已认证的；

（3）一条认证请求包括认证的地址、认证的状态等；

（4）认证成功之后不能再进行第2次认证。

2.3.2 共识记账的设计方案

本设计方案的认证流程如下：

（1）请求的生成。以太坊的客户端持续监测侦听，如果网站调用了认证请求，那么客户端将会向全网进行广播。Alice使用她的私钥对认证请求进行签名，并在认证请求的末尾处添加签名，以便能够让其他节点来进行校验。

（2）请求的广播。Alice将认证的请求向全网节点广播，其他的节点将会收到并将共同参与数字签名的校验。若正确，则将其纳入到矿工自己的区块中；若不正确，则丢弃。

（3）区块的生成。每当间隔一段时间，所有的节点通过挖矿进行PoW，通过解决数学难题来赢得记账的权利，此过程也是所有节点进行区块同步校对的过程。

（4）区块的广播。如果有节点通过算力找到了符合条件的随机数，将会向全网广播，该节点将是下个区块的创建者，并会获得奖励；

（5）区块写入账本。将对所有节点成功解出数学难题的广播答案进行验证，如果正确，它会将该区块纳入自己的账本中，每个节点同步进行；否则，将丢弃该区块。

2.3.3 共识记账的实现

共识层的代码在consensus/路径中，本节具体介绍共识记账的实现。

prepare函数主要用来处理区块头部信息，其定义如下：

diff = (parent_diff + (parent_diff / 2048 * max(1 - (block_timestamp - parent_timestamp) / 10, - 99))) + 2^(periodCount - 2)

其中parent_diff表示上一区块的难度；block_timestamp表示当前区块的时间戳；parent_timestamp表示上一区块的时间戳；periodCount表示区块数量。

结合官方的文档，在测试阶段，调节区块难度的值为一个较低的值，让登录认证的交易尽快被矿工打包，避免用户长时间等待，同时方便调试和使用。难度设定需基于创世区块（创世区块是指区块链的第1个区块，它是构建整个区块链系统的基础）。

seal函数用于处理挖矿的工作，需要一直递归调用，直到解决问题，解决问题之后退出。seal函数具有以下几点作用：

（1）根据区块头部的信息中的挖矿难度系数来处理计算目标值。

（2）选取随机数和区块头的哈希值，进行哈希运算。如果结果小于目标值，那么表示挖矿成功，自动退出；否则，则继续循环进行哈希运算。

（3）如果从外部收到了这个块，表示其他人已经挖矿成功并且已经得到了块，那么就会马上放弃打包当前块。

（4）Finalize函数表示挖矿成功之后奖励的事，它可以计算矿工的奖励，使矿工得到奖励。

verifyHeader函数主要用来校验区块的时间戳、校验难度值、校验区块的gas。

VerifySeal函数主要用来验证区块头部的签名信息。

2.4 接口层

本设计系统使用的是以太坊的go-ethereum客户端来连接到自己搭建的以太坊私有网络，它提供的应用程序编程接口（API）可以给本设计系统进行调用，并用来创建新地址，验证数字签名、支付和转账、查看余额等。接口层包含了以太坊智能合约脚本、分布式计算、验证加密签名和数据存储的技术。所有的请求数据通过post传递，使用json参数传递。

解析一个请求的具体的实现步骤如下：

（1）首先要对json数据进行实例化，使用NewJSONCodec编码器；

（2）通过NewJSONCodec编码器将请求转换为jsonRequest，并且获取参数有关服务名（service_name）、服务方法（service_method）和数据片段（params）；

（3）通过服务名（service_name）和服务方法（service_method），查找已经注册的rpc服务；

（4）向rpc服务进行请求，之后的操作都在rpc服务中进行；

（5）rpc返回结果，接着对json序列化，返回结果值。

例如：本设计系统使用Ethereum客户端账，在向rpc服务发送请求时，设定service_name为指定以太坊的服务名，并设定service_method为sendRawTransaction，通过调用rpc服务，返回的结果是TxnHash字符串的json数据。

3 结束语

本设计方案最大的创新之处在于身份识别系统基于以太坊智能合约技术，立足于传统互联网行业的现状，解决了中心化管理的麻烦，以及用户信息容易被篡改、被黑客盗用，中心服务器被攻击等事关国家信息安全痛点的问题。

在本设计中，认证请求者向系统提出认证请求，服务器节点在收到请求后，采用认证方案对识别认证者的请求，同时将认证信息加入到认证区块链中。这个过程解决了分布式账本的一致性和安全性问题，不需要第三方中介的引入。

该系统具有以下创新特点：

（1）去中心化，防止伪造。根据当今互联网产业的需求，本系统使用P2P技术，改善了数据的存储。所有

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P7

的数据通过分布式存储保存在各个节点，每个用户都是一个节点，通过节点的共识，完成身份识别，不依赖第三方。

（2）数据校验，真实可信。结合以太坊改进之后的MPT树，以及通过RLPx加密握手协议，并充分利用ECDSA的非对称加密的优势，避免了黑客通过网络传输作弊的行为。整个认证过程由节点们共同完成，使数据真实可信。

（3）安全性高，黑客止步。基于区块链的去中心化特点，每个节点的地位都是对等的，即使某个或者部分节点被摧毁都不会影响整个系统的安全，也不会造成数据的丢失。黑客如果想篡改数据，需要攻击、修改一半以上的节点数据，这几乎是无法实现的。

（4）调用方便，拓展性高。本系统设计的接口层，通过json传递参数到以太坊客户端，基于接口层方便调用的特点，可快速搭建更多不同开发语言的网站和程序，同时客户端支持多个网站和程序，并调用获取账号的信息，实现了账号的统一身份认证。

（5）部署简单，用途广泛。本设计方案可以部署在企业、政府机构、教育结构等，例如：在企业中，多个部门可以使用同一套以太坊网络，无需在每个部门都进行部署，支持跨多个部门，适合部署在大型企业中。

本设计仍然有很多的可拓展之处，除了身份认证，还可以通过以太坊智能合约开发更多功能，例如：房屋出租、契约、贷款平台等，能给使用者带来显著的安全效益、经济效益、管理效益、科研效益。

参考文献

[1] 韦康博.解读区块链——重新定义未来经济[M].北京:人民邮电出版社,2017
[2] 申屠青春. 区块链开发指南[M].北京:机械工业出版社,2017
[3] 徐明星, 田颖, 李霁月.图说区块链[M].北京:中信出版社,2017
[4] 杨波.密码学中的可证明安全性[M].北京:清华大学出版社,2017
[5] MANN C, LOEBENBERGER D. Two-Factor Authentication for the Bitcoin Protocol[J]. International Journal of Information Security, 2017,16(2): 213 226.DOI:10.1007/s10207-016-0325-1
[6] 陈志德, 黄欣沂, 许力. 身份认证安全协议理论与应用[M]. 北京:电子工业出版社, 2015
[7] 庹小忠. 区块链在身份认证中的应用[J]. 科技经济导刊, 2017(3):26-27
[8] 邓迪. 区块链技术最新的认识和成果[J]. 新经济, 2016, (19): 90-91
[9] 康双勇. 区块链中的身份认证问题研究[J]. 保密科学技术, 2018(5): 32-35
[10] 陈烨, 许冬瑾, 肖亮. 基于区块链的网络安全技术综述[J]. 电信科学, 2018, 34(3): 8-16. DOI:10.11959/j.issn.1000-0801.2018135

作者简介

苏宣瑞，桂林电子科技大学在读本科生；主要研究方向为信息安全以及区块链等。

邹秀清，桂林电子科技大学在读硕士研究生；主要研究方向为信息安全、区块链等。

丁勇，桂林电子科技大学计算机与信息安全学院教授、副院长，广西密码学与信息安全重点实验室主任；主要研究方向为公钥密码理论、同态加密、密码安全协议、区块链等；主持国家自然科学基金、中国密码发展基金、国防预研基金、广西区自然科学基金等项目10余项；发表论文60余篇，其中SCI/EI检索30余篇，出版学术专著1部、工信部规划教材1部。

D:\MAG\2018-12-143/VOL23\F6.VFT——8PPS/P8