LIGHT · FAST · SWEET

TrustNote 技术白皮书

TrustNote 基金会

2018 年 3 月

内容说明：

此文档是 TrustNote 技术白皮书 V1.1.7 版本，由 TrustNote 基金会委托开发团队撰写，主要介绍 TrustNote 的背景、技术特点和应用场景等内容。未来我们会持续升级此文档，使其与技术实现保持一致。欲了解 TrustNote 的最新资讯、技术白皮书、软件发布、开发者社区等信息，请访问官方网站 www.trustnote.org。

联系我们：

基金会：foundation@trustnote.org

社区和技术：community@trustnote.org

版权声明：

免责声明：

技术在不断发展，区块链也在不断进步。TrustNote 开发团队未来会在基金会的监管下，根据需要改进技术方案，并持续更新技术白皮书，但基础代币的发行量和分发规则保持不变。另外，近期发现个别人员以私募的名义欺骗投资者。TrustNote 基金会在此敬告通过非官方渠道参与数字加密货币投资的机构和个人，注意防范风险，TrustNote 基金会和开发团队不承担通过非官方渠道参与投资所引发的任何后果。

摘要：

TrustNote 针对现有区块链普遍存在的交易拥堵、交易费高、交易确认时间长等问题，以“极轻、极速、极趣”为目标，构建世界领先的支持挖矿的 DAG 公有链，支持海量并发交易且交易确认更快。TrustNote 具有创新的双层共识机制和 TrustME 共识算法，利用该算法周期性地选出若干超级节点，赋予它们数据单元的公证权，并会根据它们发出有效的公证单元获得 Coinbase 奖励。TrustNote 聚焦于打造简单易用的去中心化数字通证底层区块链，利用增强表达能力的声明式智能合约，用户可自由创建和发布数字通证，而无需编写复杂的智能合约代码。TrustNote 拥有可扩展的钱包，为数字通证、区块链游戏和社交网络提供安全保障和丰富的应用接口，让新奇特的想法在区块链上流畅地运行，让使用区块链成为一种生活方式。

目录

1 背景

2009 年 1 月 3 日至今，比特币已经安全运行了接近 9 年，这在计算机网络历史上堪称奇迹。比特币的成功开启了一个规模庞大的、充满想象空间的数字加密货币生态。作为比特币的开发者，中本聪创造性地提出了区块链这种基于哈希函数的链式数据结构，并成功构建了能够良好运转的去中心化 P2P 网络，开启了数字加密代币的新纪元。区块链像一台飞速运转的机器，影响并改变着所到之处，用不断释放的能量激发人们的创新活力。

区块链提供了一种去中心化的信任机制，已经成为当今世界数据保护和价值交换的全新模式和重要方法论。区块链技术正处在蓬勃发展期，各种技术不断与区块链融合，各类场景也在探索如何利用区块链的技术特性，其应用方向已从数据防篡改和价值交换扩展到数字通证和社交等领域。区块链的应用场景和模式正在不断扩展，这对区块链技术提出了许多挑战，需要更强的安全性、更高的交易并发量和更短的交易确认时间。

在比特币的区块链中，数据块是以有向单链的形式相互链接的，由于存在区块大小和共识机制等限制，交易量非常有限且交易确认时间较长，促使交易费不断上涨，经常发生交易拥堵等情况。为解决上述问题，比特币开发者社区提出了区块扩容、隔离见证和闪电网络等解决方案，但这些方案都不完美，要么只是缓解问题，要么会牺牲安全性或一致性，并未在社区和生态中就此达成一致意见。近期出现的多个比特币分叉，被戏称为“IFO”，将这一问题的争论推向白热化。

传统区块链的块链式结构是阻碍区块链提高并发性的瓶颈，技术极客们不断寻找更高效的数据块链接形式，提出有向无环图（Directed Acyclic Graph，DAG）与区块链相结合的解决方案，以下称为“DAG 链”。DAG 链没有区块的概念，所以没有区块容量的限制。另外，DAG 链通过新交易验证并引用旧交易的方式进行交易确认，允许用户的账本之间存在临时性的微小差异，通过短时间内弱化数据块全网一致性的方式，达到不发生交易阻塞的效果。DAG 链网络节点规模越大、交易量越大，则交易确认时间越短。

IOTA 和 Byteball 在 2016 年构建出各自的 DAG 公有链，以适应高频次交易的应用场景。美中不足的是，DAG 链支持高频交易，但在交易频次较低的情况下，旧交易无法获得足够多新交易对其的验证和引用，致使旧交易无法被及时确认，极端情况下交易可能永远不会被确认。为此，IOTA 提出一种临时性的中心化角色，该角色称为协调者，它用于在交易量较小的时候保护网络，但并没有公开协调者的设计细节；而 Byteball 引入十二位见证人，由见证人发送公证交易实现交易确认，虽然宣称用户有权选择自己信任的见证人，但其交易引用规则使得用户很难主动更换见证人。

2 TrustNote 是什么

TrustNote 是支持挖矿的 DAG 公有链，具有创新的双层共识机制，面向数字通证发行、区块链游戏和社交网络等应用场景，基础代币称为“TTT”。TrustNote 的宗旨是“极轻、极速、极趣”。“极轻”是指 TrustNote 拥有轻量化的架构和智能合约系统，支持轻量级应用扩展和微钱包；“极速”是指 TrustNote 支持高并发交易、交易确认速度快、dApp 开发简单上线快；“极趣”是指 TrustNote 打造的应用生态体系，让新奇特的想法在 TrustNote 上流畅地运行，让使用区块链成为一种生活方式。

2.1 关键特性

- 拥有双层共识机制，是支持挖矿的 DAG 公有链
- 支持高并发交易，交易极速确认
- 支持高级声明式智能合约
- 数字通证发行平台
- 哈希算法：BLAKE2
- 签名算法：EdDSA
- 多平台钱包、轻钱包、微钱包，支持第三方扩展

2.2 有向无环图

若一个有向图无法从某顶点出发经过若干条边回到该点，则称该图为有向无环图（Directed Acyclic Graph，DAG）。使用 DAG 数据结构存储区块链账本数据的模式，正逐步受到更多开发者的关注。已有 IOTA 和 Byteball 等多个项目利用 DAG 技术成功构建了能够稳定运行的公有链，证明了 DAG 链的可行性。

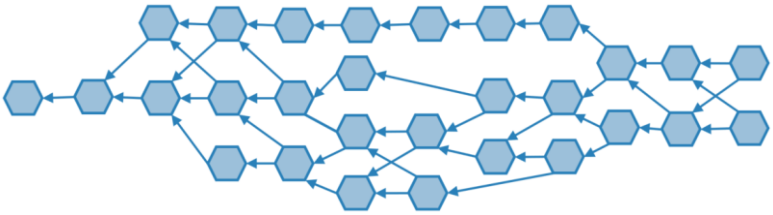

图 2-1 有向无环图

在 TrustNote 中，交易被视为一种消息，支持多种类型的消息，多个消息可组合成一个数据块，该数据块称作一个单元（Unit），单元与单元之间相互链接组合成一个 DAG 图。由于单元可以链接到任意一个或多个之前的单元，不需要为共识问题付出更多的计算成本和时间成本，也不必等待节点之间数据强同步，甚至没有多个数据单元拼装区块的概念，因此可以极大提高交易的并发量，并把确认时间降低到最小。

TrustNote 使用以下方案解决双花问题（一份数字货币被花了两次）。首先，尝试在 DAG 图上找到一条以创世单元为起点的主链（Main Chain，MC），并给位于主链上的单元分配索引，创世单元索引为 0，创世单元的子单元索引为 1，以此类推。然后，对于不在主链上的单元，定义其索引等于引用此单元的第一个主链单元的索引。最终，DAG 上的每笔交易都拥有了一个索引。如果两笔交易尝试使用同一笔输出，只需要比较其索引（Main Chain Index，MCI）的大小，小的有效，大的无效，由此解决双花问题。例如，图 2-2 中存在两笔双花交易，当它们的 MCI 确定后，其中一笔双花的 MCI 是 11，而另一笔双花的 MCI 是 15。所以，我们可以认定 MCI 为 11 的交易有效，而拒绝 MCI 为 15 的交易。

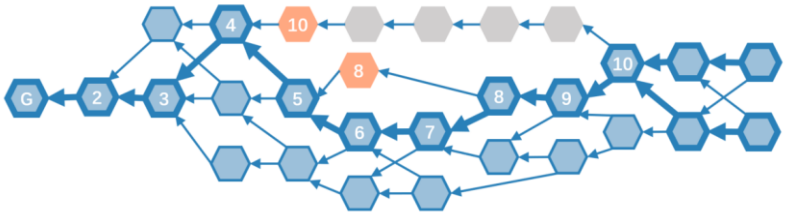

图 2-2 主链

安全性方面，不同于比特币的区块链以巨大算力作保障，基于 DAG 的 TrustNote 依靠交易的快速推进，以及交易之间关联关系的不确定性作为“防火墙”，使整个系统毫无规则，无从攻击。TrustNote 具有创新的双层共识机制和 TrustME 共识算法，利用该算法周期性地选出若干超级节点，赋予它们数据单元的公证权，并会根据它们发出有效的公证单元获得 Coinbase 奖励。

2.3 项目对比

TrustNote 站在巨人的肩膀上，吸纳现有区块链项目的优点，解决它们的突出问题，构建更加繁荣的应用生态。TrustNote 采用创新的双层共识机制，具有高安全性的密码算法，以挖矿的方式使超级节点参与单元公证。TrustNote 与当下知名 DAG 链（IOTA 和 Byteball）进行了横向对比，对比结果如表 2-1 所示。

表 2-1 方案对比

	IOTA	Byteball	TrustNote
代币	IOTA	Byte	TTT
市值	140 亿美元	4 亿美元	--
共识机制	PoW 权重累加	12 名公证人	去中心化 TrustME 共识
智能合约	不支持	声明式合约	高级声明式合约
奖励机制	无	交易引用和公证	交易引用和挖矿
节点分类	全节点 轻节点	全节点 轻节点	超级节点 全节点 轻节点 微节点
交易费	无	有	有
双花问题	PoW 权重比较	MainChain 定序	MainChain 定序
低交易量问题	中心化的协调者	弱中心化的公证人	TrustME 公证人

3 数据结构

3.1 单元

当节点发起交易或发送消息时，首先创建一个新的数据块并广播给 peer 节点，称为“单元（Unit）”。一个单元可以包含多个不同类型的消息，单元中包含以下内容：

- 头部：当前单元所引用的之前单元（父单元）的哈希值。
- 消息组：一个单元可以包含多个消息（Message），消息有多种不同类型，每种消息都有自己的数据结构定义。
- 签名：对所创建单元的一个或多个用户的数字签名，单个用户可以拥有多个地址，地址利用 BIP-0044 算法生成。

单元字段定义参见表 3-1。

表 3-1 单元字段定义

字段	含义	说明
version	协议版本号	例如：’1.0’
alt	代币标识	例如：‘1’
messages	消息数组	详见后续消息字段及消息类型描述
authors	作者数组	单元作者地址组成的数组
parent_units	父单元哈希数组	父单元的哈希组成的数组

其中，messages 存储单元的实际数据，是包含一个或多个消息的数组。TrustNote 支持存储多种类型的消息，通过 app 字段区分。

表 3-2 消息字段定义

字段	含义	说明
app	消息类型	‘payment’、‘text’等，详见后续消息类型说明
payload_location	消息主体所在位置	‘inline’表示消息主体保存在当前消息中 ‘uri’表示通过 uri 地址获取消息主体

		‘none’表示无消息主体
payload_hash	消息主体的哈希	--
payload	消息主体	存在多种不同类型消息用于保存不同类型的数据内容，例如交易消息包含交易输入和输出
payload_inputs	消息输入数组	包括产生该输入的单元哈希、消息索引、输出索引等内容
payload_outputs	消息输出数组	包括接收者的地址、交易金额等内容

3.2 消息类型

TrustNote 支持多种消息类型，并可根据需要进行类型扩展。不同类型的消息拥有不同的解析规则，用于存储不同类型的数据。在 TrustNote 中，通过消息中的 app 字段区分不同的消息类型。

- 公证权证明（app = TrustME）

公证权证明类型消息只能由公证节点生成，该消息用于存储某节点拥有的公证权证明结果。如果某单元的第一个消息是公证权证明消息，则该单元是公证单元。公证权消息内容包含：共识轮次、公证单元序号、最新稳定共识轮号、最新稳定共识轮的 Coinbase、种子、难度、公证权证明、公证优先级等内容。关于公证节点和公证权获得方式的说明，请参见“共识机制”。

```
messages:[{
  app:'TrustME',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    round: 'round number',
    sequence: 'sequence number in current round',
    last stable round: 'number of the last stable round',
    coinbase of the last stable round: [
      {address:'...',amount:1.2 GN},
      {address:'...',amount:2800 MN},
      ...
    ],
    seed: 'string of seed',
    difficulty: 'difficulty number',
    proof: 'consensus result',
    priority: 'priority of notary'
  }
}]
```

- 交易（app = payment）

交易类型消息用于保存各类数字通证的交易信息。一个交易消息中可包含多个输入和输出。对于用户自己定义的资产，需要在消息体中指定该资产定义所在单元的哈希值。一个标准的交易消息如下：

```
messages:[{
  app:'payment',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    inputs:[{
      unit:'...',
      message_index:0,
      output_index:0
    }],
    outputs:[
      {address:'...',amount:1200},
      {address:'...',amount:2800}
    ]
  }
}]
```

- 文本（app = text）

  文本类型消息用于存储任意的可显字符串数据。

```
messages:[{
  app:'text',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:'any text'
}]
```

- 结构化数据（app = data）

  结构化数据类型消息用于存储任意的结构化数据。

```
messages:[{
  app:'data',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    any structured data
  }
}]
```

- 数据供给（app = data_feed）

  数据供给类型消息可由某个可信第三方发出，用于触发智能合约。

```
messages:[{
  app:'data_feed',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    'data feed name':'...',
    'another data feed name':'...'
  }
}]
```

- 修改地址定义（app = address_definition_change）

  修改地址定义类型消息用于修改地址定义并保留旧地址。

```
messages:[{
  app:'address_definition_change',
  definition_chash:'...'
}]
```

- 资产定义（app = asset）

资产定义类型消息用于定义新的资产。

```
messages:[{
  app:'asset',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    cap:1000000000,
    is_private: true,
    is_transferrable: true,
    auto_destroy: false,
    fixed_denominations: false,
    issued_by_definer_only: true,
    cosigned_by_definer: false,
    spender_attested: false,
    attestors:[...]
  }
}]
```

各字段含义：

- cap：资产最大发行量。
- is_private：指定资产交换是私有的还是公开的。
- is_transferrable：指定资产是否可以不经过发行者就可以在第三方之间转移。如果设置为 false，那么发行者必须参与转移。
- auto_destroy：指定资产发送给发行者后是否销毁。
- fixed_denominations：指定资产是否可以按照任意整数发送，或者只能按照固定金额（例如：1、2、5、10、20 等）。
- issued_by_definer_only：指定资产是否只能由发行者发行。
- cosigned_by_definer：指定是否每次资产转移都需要发行者的联合签名。
- spender_attested：指定资产的消费者消费前是否需要认证，如果他无意间收到资产但还没有认证，他必须通过定义中罗列的认证者认证后，才能消费，这个需求对于管理资产很有帮助。
- attestor：指定资产发行者指定的认证者地址列表（仅当 spender_attested 为 true 时），发行者可以通过发送一个 asset_attestors

消息修改认证人列表。

- denominations：仅当 fixed_denominations 为 true 时，列出允许的面额，以及每种面额的发行数量。
- transfer_condition：定义资产允许转移的条件，这个定义的语法与地址定义相同，除了不能引用认证的数据，例如“sig”。默认情况下，除了已经定义在其它字段的条件，没有其它限制。
- issue_condition 与 transfer_condition 相同，针对发行交易。

- 修改认证人列表（app = asset_attestors）

资产定义者可以通过此类型消息修改资产的认证人列表。

- 个人信息（app = profile）

个人信息类型消息用于个人披露自身信息，通过可信第三方发送的证实消息可验证其真实性。

```
messages:[{
  app:'profile',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    name:'Tim',
    emails:'[tim@example.com]',
    twiter:'tim'
  }
}]
```

- 证实（app = attestation）

证实类型消息用于可信第三方公开与某地址关联的个人信息，利用证实消息可以实现某些业务模式中所需的用户认证。

```
messages:[{
  app:'attestation',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    address:'Address of the subject',
    profile:{
      name:'Tim',
      emails:'[tim@example.com]'
    }
  }
}]
```

- 发起投票（app = poll）

  发起投票类型消息。

```
messages:[{
  app:'poll',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    question:'...',
    choices:['A','B']
  }
}]
```

- 参与投票（app = vote）

  参与投票类型消息。

```
messages:[{
  app:'vote',
  payload_location:'inline',
  payload_hash:'hash of payload',
  payload:{
    unit:'hash of the unit where the poll was defined',
    choice:'A'
  }
}]
```

4 共识机制

TrustNote 采用双层共识机制，该共识机制包括基础共识和公证共识。基础共识是指节点发送的新交易单元对旧交易单元的验证与引用，也称“DAG 共识”；公证共识是指根据公证节点提交的公证单元严格确定其它单元的顺序，也称“TrustME 共识”。这种双层共识机制有利于提供较高的交易吞吐量和较短的交易确认时间，能有效解决过度分叉和双花问题。

为使 TrustNote 生态更加健壮，我们设计了两种 TrustME 共识方案。TrustNote 将在早期采用基于工作量证明（Proof of Work，PoW）的 TrustME 共识方案，该方案称为“TrustME-PoW”；未来，TrustNote 计划采用基于拜占庭协商（Byzantine Agreement，BA）的 TrustME 共识方案，该方案称为“TrustME-BA”。无论哪种方案，若超级节点参与共识并被选作公证节点，它将会获得 TTT 作为奖励。

在 TrustME-PoW 中，超级节点通过证明自己的运算能力获得公证权，而在 TrustME-BA 情况下，利用伪随机算法选择超级节点并赋予其公证权。无论哪种 TrustME 共识方案，公证节点发出的公证单元都同样遵守单元引用规则，并且不会影响其它单元之间已经建立的引用关系。而只有在公证单元成为主链上的稳定单元后，才能最终证明某个公证节点做了有益于维护 TrustNote 的工作，并因此获得公证奖励。另外，两种 TrustME 共识机制都鼓励所有节点公平参与，相比中心化和弱中心化的方案更加公平、可信、安全。

4.1 节点

TrustNote 节点有四种类型，分别为超级节点（super node）、全节点（full node）、轻节点（light node）和微节点（micro node），节点对比如表 4-1 所示。

表 4-1 节点对比

	超级节点	全节点	轻节点	微节点
账本	全账本	全账本	轻账本	无账本
交易	√	√	√	委托交易
DAG 共识	√	√	间接	×
TrustME-PoW	√	×	×	×
TrustME-BA	√	×	×	×

	超级节点	全节点	轻节点	微节点
微节点托管	√	×	×	×
安装环境	矿机 云主机 服务器 PC	云主机 服务器 PC	智能手机 平板电脑	MCU 智能卡

TrustNote 具有与比特币相似的 P2P 网络，每一个节点可选择一组随机的 peer 节点，并利用 Gossip 协议来传播消息。为了保证消息不被伪造，每一个消息都被原始发送者的私钥签名，其它节点在转发它之前要验证签名是否有效。为了避免消息传播出现前向回环，节点不会转发重复的消息。

成为 TrustNote 超级节点，需满足以下条件：

- 有资源：具有良好的网络带宽、存储空间和运算能力，最好拥有公网 IP；
- 有币：在多个共识周期内平均持有一定数量的 TTT；
- 有信用：之前提交的所有单元都被认为是有效的。

成为超级节点之后，可参与 TrustME 共识，进而成为公证节点。公证节点可通过发送公证单元获得公证奖励并能赚取普通单元的公证费。

4.2 单元引用

TrustNote 中每个单元可引用多个彼此之间没有父子继承关系的单元，而且新的子单元会优先链接有更多父单元的单元。如果沿着某个父单元向子单元方向推进，当一个单元被多个子单元引用时，我们将观察到许多分叉，而当多个父单元被同一个子单元引用时，这些单元将逐渐汇聚。

引用父单元的目的是建立单元之间的模糊次序。引用父单元之前，需要验证父单元有效性，如签名是否有效、单元引用是否合规等。TrustNote 不要求节点之间强同步，不同节点看到的 DAG 链图存在临时性的不一致，但这并不会破坏已经建立的父子关系，只可能会使一个父单元有多个子单元。而正因为不追求节点之间的单元数据强同步，TrustNote 可以容忍高交易吞吐量和网络延迟。

为减少 DAG 中可能出现的垃圾单元，所有节点提交新单元时需要支付交易

费。交易费分成两部分，分别支付给将该单元作为父单元的节点以及为该单元提供公证的节点。若一个单元被多个子单元引用，那么发送拥有最小哈希值单元的节点获得引用奖励。同时，限制获得引用奖励的子单元的 MCI 值应等于或略大于其父单元的 MCI，激励节点尽可能快地引用更多最新的父单元，以获得更多引用奖励。这有助于 DAG 快速收敛，减少分叉。

4.3 主链

主链是在指定单元所见的 DAG 图中沿着子-父链接找到一个单链，可以把所有单元都关联在一起。所有的单元要么直接在这条主链之上，要么从主链上的单元沿着 DAG 的边缘通过少量的跳跃可以到达。如果我们从另一个顶点开始，我们会构建另一条主链。如果以相同的规则在两个不同的顶点选择主链，这两条主链在回溯过程中一旦相交，它们会在交点之后完全重合。最坏的情况，两条主链在创世单元相交。虽然节点生成单元的过程中是相互独立的，也不存在任何协调的可能，我们仍期望主链之间的交点可以尽可能的靠近顶点。

一旦某单元选择出一条主链（MC），它就可以在两个冲突的无序单元之间建立总序。首先，给直接位于主链上的单元做个索引，创世单元索引为 0，创世单元的子单元索引为 1，以此类推，沿着主链给主链上的所有单元分配索引。对于不在主链上的单元，我们找到第一个直接或间接引用此单元的主链单元。这样，就给每一个单元分配了一个主链索引（MCI）。然后，给定两个单元，拥有较小 MCI 的单元被认为是更早生成的。如果两个单元的 MCI 恰好相同并且存在冲突，则拥有较小哈希值的单元有效。TrustNote 会保留双花的所有单元，包括最终认定无效的单元。

主链构建过程实际上是父单元选择算法的递归调用过程。通过参与 TrustME 共识，超级节点有机会成为具有公证权的节点，可以发送公证单元。通过比较可选路径中公证单元的数量，父单元选择算法能在给定的所有可选父单元中选出一个作为“最优父单元”。对于不同节点，主链构建过程是完全独立的，仅依赖于节点自身能看到的 DAG 图。从 DAG 图的无子单元开始，沿着最优父单元的路径做历史回溯，节点可以构建一条能通达创世单元的主链。

4.4 交易确认

当获得新的单元时，每一个节点会持续追踪自身的当前 MC，好像他们将要基于当前的所有无子单元构建新单元。不同节点各自的当前 MC 也许不同，因为它们有可能看到不同的非稳定单元集合。而当新单元到达时，当前 MC 会不断变化。然而，当前 MC 的足够老的那部分会保持不变。

未来所有的 MC 在回溯时将会汇集某个 MC 单元，这个 MC 单元以及之前的所有 MC 单元都是稳定的，不会因为新单元的到来而改变。事实上，创世单元是一个天然的初始稳定节点。假设我们已经基于当前的非稳定单元集合构造了一条当前 MC，并且这条链上已经有一些之前认定稳定的节点，也就是说未来的当前 MC 都被相信会在这个点或早于这个点汇集，然后就沿同一条路径回溯。如果我们能找到一个方法，把这个稳定点向远离创世单元的方向推进，就可以根据数学归纳法证明这个稳定点存在。而被这个稳定点所引用的单元将获得确定的 MCI，包含在这些单元中的所有消息也将被确认。

4.5 交易费与挖矿奖励

发布交易需要支付交易费，节点根据生成单元的字节数计算交易费。交易费被分成两部分，60%作为单元引用费，40%作为公证费。引用费将被该单元的子单元获得，而公证费将被累加到主链中 MCI 值最接近的公证单元所在共识轮的公证奖金池。公证单元也需要支付交易费，交易费计算方法与普通单元相同。公证单元包含较多信息，通常比普通单元占用更大的存储空间，因而交易费也相对较高，激励其它节点引用公证单元。

在 TrustNote 中，DAG 链的增长与 TrustME 共识是异步的，每轮共识会选举出多名公证人，这些公证人有权提交公证单元。在主链没有稳定的情况下，无法确定哪些公证单元在主链上，也无法计算公证单元的有效引用，所以没办法像比特币一样直接给出 Coinbase。当某共识轮的所有公证节点发出的所有公证单元都成为稳定单元后，才能确定该共识轮的每个公证节点分别可以获得多少公证奖金。需要说明的是，普通单元也可能会出现在主链中，它们可能会获得父单元支付的引用费，但不会分享公证奖金。

TrustME 共识周期性地执行，每轮都会选出一定数量的公证节点。如果某轮公证节点发送的所有公证单元都成为稳定单元，则称该轮共识稳定。每个公证单元的第一个消息是公证节点自身的公证权证明。每次达成 TrustME 共识后，当前轮公证节点生成的首个公证单元必须包含最新稳定共识轮的 Coinbase。需要说明的是，最新稳定共识轮的 Coinbase 在进行新一轮 TrustME 共识时已经确定，是 TrustME 共识算法的输入变量之一。在同一共识轮内，由不同公证节点生成的其它公证单元不再包含上述内容，而是通过验证并引用首个公证单元的方式确认稳定共识轮的 Coinbase 内容。这样做可以进一步削弱公证节点的能力，防止恶意的超级节点通过多次获得公证权干扰稳定共识轮公证节点的 Coinbase 收益。

4.6 TrustME-PoW

TrustME-PoW 是基于工作量证明选择公证节点的共识机制，每轮共识将选择少量节点成为公证节点，并确定相关公证节点的优先级。TrustME-PoW 共识算法每五分钟执行一次，每次达成共识会选出不超过二十个超级节点作为公证节点，这些公证节点有权发送公证单元并以此获得公证奖励。

TrustME-PoW 基于 Equihash 算法，以 BLAKE2 为底层哈希函数，降低使用 ASIC 挖矿的优势，更多超级节点可以公平参与挖矿，使超级节点成为公证节点的概率分布更加合理。Equihash 算法的输入为当前轮数、种子和难度系数等内容。当前轮数起始为 0，每轮加 1；每轮共识的种子根据上一轮共识的种子和取得的共识结果计算得出，种子可被公开获得及验证；难度系数根据全网平均算力推算得出，通过调整难度系数可以控制达成共识的平均时间间隔。

公证单元必须满足前文所述的单元引用规则。公证单元只能引用非稳定单元，并且必须验证引用单元及其“子-父”链的正确性，直到验证到稳定的 MC 单元。鼓励公证单元引用多个非稳定状态的最优父单元，加速单元稳定，促使 DAG 链向前推进并收敛。

只有公证单元成为主链上的单元，才能获得相应的公证奖励。同一共识轮内，所有主链上公证单元按照各自的有效引用单元数量计算当前共识轮公证奖金的分配比例。每个公证单元在成为主链单元并稳定之后，计算出该公证单元的有效

引用数量。

4.7 TrustME-BA

TrustME-BA 是一种基于可验证随机函数（Verifiable Random Function，VRF）和 BA 算法构建的共识机制，该共识机制能够随机选出少量超级节点作为公证节点，并确定公证节点的优先级。

TrustME-BA 每一分钟执行一次，每次达成共识将随机选出若干超级节点作为公证节点，公证节点有权发送公证单元，公证单元须满足 DAG 共识中的父子引用规则。公证节点发送的公证单元成为稳定主链的单元后，该公证节点可以获得公证奖励。当交易活跃时，新单元不断产生，则公证节点会及时获得公证奖励；当交易不活跃时，极端情况下一分钟内没有新单元产生，已经发送公证单元的节点在发送的公证单元成为稳定主链单元时获得公证奖励，没有发送公证单元的节点不获得公证奖励。

4.7.1 设计目标

TrustME-BA 共识机制可实现以下两个目标：

- 安全目标

  很大概率上，所有的超级节点在选出的公证节点集合上意见一致。也就是说，如果大部分诚实的超级节点接受某共识结果，那么未来的任意共识过程都可以追溯到该共识结果。TrustME-BA 假设诚实的超级节点持有超过总量 2/3 的基础代币，并且攻击者可以参与共识并获得相应的奖励。这种假设的合理性在于，想要成功攻击 TrustME-BA，攻击者必须在其中投入足够多的基础代币。TrustME-BA 假设攻击者可以控制一定数量的目标超级节点，但他不能控制大量的超级节点，以至于他所控制的超级节点所持有的基础代币的数量超过了总量的 2/3。

- 保活目标

  在安全目标之外，TrustME-BA 具有关于网络可达性的假设，以严格地

确定公证节点之间的优先级。TrustME-BA 的目标是超级节点能在一分钟时间内对选出的公证节点集合达成共识。为达到保活，TrustME-BA 做了一个强同步假设，即所有诚实的超级节点能在一个已知的时间范围内将消息发送给其它大部分诚实节点。这个假设允许攻击者控制一部分诚实超级节点的网络，但他不能大范围的控制整个网络，也不能对网络进行分割。

4.7.2 两种共识状态

TrustME-BA 有最终共识和临时共识两种状态。

如果一个超级节点达到最终共识，意味着任何其它超级节点也达到了最终共识或者在同一轮中的临时共识必须同意这一共识结果，而无论强同步假设是否成立。而临时共识意味着其它超级节点可能在其它公证单元上达到了临时共识，没有超级节点已经达到了最终共识。所有公证单元都必须直接或间接引用之前生成的公证单元，这可以确保 TrustME-BA 的安全性。

TrustME-BA 产生临时共识有两种情况。首先，如果网络是强同步的，一个攻击者可以以一个很小的概率让 TrustME-BA 达到临时共识。此情况下，TrustME-BA 不会达成最终共识，也不能确认网络是强同步的。但经过几轮后，很大概率上会达到最终共识。第二种情况是，网络是弱同步的，整个网络都被攻击者控制。此情况下，TrustME-BA 将达到临时共识，选举出不同的公证节点集合，形成多分叉共识。这能够避免 TrustME-BA 达到最终共识，因为超级节点被分成了不同的组，各组之间并不同意对方。为了恢复活性，TrustME-BA 将被周期性地执行，直到消除意见分歧。一旦网络恢复到强同步状态，将会在短时间内达成共识。

4.7.3 抽签算法

抽签算法是基于可验证随机函数（VRF）构造而成的，可根据每个参与 TrustME-BA 共识的超级节点的权重选出这些节点的随机子集。某超级节点被选中的概率约等于自身权重与总权重的比值。抽签的随机性源于 VRF 函数和一个可公开验证的随机种子，每个超级节点可根据随机种子验证自己是否被选中。

VRF 函数定义：给定任意字符串，VRF 函数输出哈希值和证明结果。

$$\langle hash, \pi \rangle \leftarrow VRF_{sk}(seed \,||\, role)$$

哈希值 *hash* 由私钥 *sk* 和给定的字符串(*seed*||*role*)唯一确定，在不知道私钥 *sk* 的情况下，输出的哈希值 *hash* 与随机数之间不可区分。证明结果 π 使得，知道私钥所对应公钥的节点可以验证哈希值 *hash* 和字符串 *seed* 之间是否关联。种子 *seed* 是随机选择并且可以被公开获得的，每一轮运算的 *seed* 由前一轮运算的 *seed* 生成。抽签算法支持角色指定，如选出协商过程中某一步骤的参与者。

所有超级节点执行抽签算法来确定自己是否被赋予公证权，被选中的超级节点通过 P2P 网络向其它超级节点广播自己的抽签结果。需要说明的是，抽签选择超级节点的概率与超级节点自身权重成正比，以抵御 Sybil 攻击。权重大的超级节点可能会被选中多次，为此抽签算法会输出超级节点被选中的次数。如果一个超级节点被多次选中，那么它就被当成多个不同的超级节点。

4.7.4 拜占庭协商

拜占庭协商（BA）能为每一个被选中的超级节点确定公证优先级并提供公证优先级的证明。达成拜占庭共识需要执行多个步骤，BA 算法会被执行多次。每次协商都从抽签开始，所有超级节点都去查看它们是否被选中成为当前 BA 的参与者。参与者广播一个包含选择公证优先级的消息。每一个超级节点用它们收到的公证优先级消息去初始化 BA 算法。上述过程将被不断重复执行，直到某轮协商有足够多的超级节点达成共识。在不同超级节点之间，BA 算法并不是同步的，每个超级节点发现之前的步骤结束后应立即查看新的参与者选举结果。只有超级节点在某轮协商中投票并最终达成共识，它才可以参与下一轮协商。

BA 算法的一个重要特征是，参与者不需要维护私有状态，仅存私钥，所以参与者每个步骤之后都可以被更换，以减少对参与者的攻击。当网络是强同步的，BA 算法保证如果所有的诚实超级节点以相同内容进行初始化，那么可以在很少的交互步骤之内达到最终共识。此情况下，即使存在少量攻击者，所有的诚实超级节点也将在有限交互步骤步下在达到最终共识。

5 智能合约

TrustNote 拥有非图灵完备的声明式智能合约系统，支持布尔运算、数学运算、合约数据存储，不支持栈和跳转指令。这种智能合约语言能直接描述合约期望的目标，表达能力强，易于理解，安全性高。与以太坊相比，TrustNote 的智能合约系统具有复杂度低、轻量化和高性能等优势，同时还降低了合约编写难度和出错概率。

TrustNote 没有账户的概念，TTT 以 UTXO 的形式存储在不可篡改的分布式账本的数据单元中。在 TrustNote 的智能合约语言里，地址定义是一个布尔表达式，可计算出是 true 或者 false 的结果。智能合约的所有表达式，最终都会计算出一个布尔值，多个子表达式可以通过布尔操作联合起来。例如，下面这个定义需要两个签名。

```
["and", [
   ["sig", {pubkey: "one pubkey"}],
   ["sig", {pubkey: "another pubkey"}]
]]
```

为了花费与以上定义地址的资金，必须同时提供两个签名。我们使用 JSON 来编写表达式，可以使用具备很好支持的、已经过优化的 JSON 解析器。

"Or" 操作可以用来描述需要提供任意一个公钥所对应私钥的签名。

```
["or", [
   ["sig", {pubkey: "laptop pubkey"}],
   ["sig", {pubkey: "smartphone pubkey"}],
   ["sig", {pubkey: "tablet pubkey"}]
]]
```

可以用以上定义实现在三个不同设备上可以控制同一个 address，这三个设备可能是你的电脑、手机和平板。

地址定义的指令可以嵌套，如：

```
["and", [
    ["or", [
        ["sig", {pubkey: "laptop pubkey"}],
        ["sig", {pubkey: "tablet pubkey"}]
    ]],
    ["sig", {pubkey: "smartphone pubkey"}]
]]
```

地址定义可以要求一个集合中参与者必须达到某个阈值，例如 2-3 签名。

```
["r of set", {
    required: 2,
    set: [
        ["sig", {pubkey: "laptop pubkey"}],
        ["sig", {pubkey: "smartphone pubkey"}],
        ["sig", {pubkey: "tablet pubkey"}]
    ]
}]
```

上面的表达式意味着任意两个签名便可以使表达式为真。如果一个密钥丢失了，这个地址仍然可用，并且能够修改这个定义，给丢失的密钥设置一个新值。另外，不同条目可以赋予不同的权重，并可以设定一个最小权重要求。

```
["weighted and", {
    required: 50,
    set: [
```

```
        {weight: 40, value: ["sig", {pubkey: "CEO pubkey"}] },
        {weight: 20, value: ["sig", {pubkey: "COO pubkey"}] },
        {weight: 20, value: ["sig", {pubkey: "CFO pubkey"}] },
        {weight: 20, value: ["sig", {pubkey: "CTO pubkey"}] }
    ]
}]
```

地址定义可以引用其它地址。

```
["and", [
    ["address", "ADDRESS 1 "],
    ["address", "ADDRESS 2"]
]]
```

这种定义意味着把签名委托给其它地址，这对于构造共同控制的地址是很有用处的。这种语法给予了用户很多方便，他们能根据自己的意愿去改变他们自己有权管理的那部分地址的定义，而不会影响别的用户。

地址定义可以用来配备添加到 DAG 中的数据。

```
["in data feed", [
    ["ADDRESS1", "ADDRESS2", …],
    "data feed name",
    "=",
    "expected value"
]]
```

如果由某地址添加到 DAG 的数据反馈结果与期望值相等，则表达式的结果为 true。通过指定数据反馈的来源，可以实现链上预言机功能。利用链上预言机可以扩展出非常强大的功能。

```
["or", [
  ["and", [
    ["address", "ADDRESS 1"],
    ["in data feed", [["EXCHANGE ADDRESS"], ["EURUSD", "+", "0.200"], ">", "1.1500"]]
  ]],
  ["and", [
    ["address", "ADDRESS 2"],
    ["in data feed", [["TIMESTAMPER ADDRESS"], "datetime", ">", "2016-10-01 00:00:00"]]
  ]]
]]
```

上面的表达式依赖两个预言机，一个会发布欧元/美元的汇率，另外一个会发布时间。首先，双方为这个表达式定义的地址准备资金，向这个地址支付各自相应的份额。然后，如果由兑换地址公布的欧元/美元汇率加上 0.200 曾经超过 1.150，则地址 1 将得到全部资金。如果在 2016 年 10 月 1 日之前，以上情况没有发生，则地址 2 将得到全部资金。另外一个有趣的例子中，消费者向商人购买货物，但是他不是很信任商人，如果货物没有发给他，希望钱可以退还给自己。此时，消费者可以把钱付给用以下方式定义的一个共享地址。

```
["or", [
  ["and", [
    ["address", "MERCHANT ADDRESS"],
    ["in data feed", [["FEDEX ADDRESS"], "tracking", "=", "123456"]]
  ]],
```

```
["and", [
  ["address", "BUYER ADDRESS"],
  ["in data feed", [["TIMESTAMPER ADDRESS"], "datetime", ">", "2016-10-01 00:00:00"]]
]]
```

这个定义有效的前提是 FedEx 会在链上存储包裹的单号。如果货物发放了，根据第一个条款，商人可以解锁资金。如果在商定的日期之前，货物没有发放，消费者可以拿回自己的资金。

地址定义可以实现交易查询。假定用户想购买至少 1200 个单元的数字资产，但是只愿意支付 1000 TTT，而且他不愿意一直在线等待卖家。他可以只在交易平台发布一个订单，当匹配的卖家出现时，自动完成交易。他可以按照以下方式创建地址，并发送 1000 TTT 到该地址。

```
["or", [
  ["address", "USER ADDRESS"],
  ["and", [
    ["address", "EXCHANGE ADDRESS"],
    ["has", {
      what: "output",
      asset: "ID of alternative asset",
      amount_at_least: 1200,
      address: "USER ADDRESS"
    }]
  ]]
```

]]

第一个或条件的含义是用户在任何时候都可以撤销订单，拿回他的 TTT。第二个或条件当满足条件的交易出现后，将 TTT 付给委托交易平台，授权它来花费资金。交易平台会公开发布订单信息，卖家可以查看订单列表，生成一个交换资产交易，并且和交易平台一起签名。

6 生态和应用场景

区块链代币的发行与交易是区块链技术核心的应用场景之一，但是从实际操作的角度来看，任何一种代币的交易都需要一个底层的钱包软件作为支撑，一旦需要发行新的币种，或者添加新的交易类型，就必须升级钱包软件，因为旧的软件很可能无法交易新的币种或支持这些新功能。以太坊是一个很好的平台，可以提供发行代币，定义交易类型等功能，但是以太坊智能合约的编写需要较高的专业技能，图灵完备的智能合约对于一些简单的资产类应用（例如，发行和交易）显得过于笨重而且容易出错，“The Dao”事件的发生不是偶然，智能合约的安全性和编写应用的易用性需要得到更高的关注，近期发生的“以太猫”拥堵也证明了以太坊的交易性能还有待提高。

TrustNote 团队会坚定的沿着项目初期制定的技术路线图前进，以技术创新为驱动力，以应用生态为目标，依托开源生态体系推动 TrustNote 持续进行技术创新，打造极轻、极速、极趣的公有链。

- 主链和钱包：主链搭建，基础版钱包功能开发并上线，代币发行。
- 项目开源：项目将在 Github 中开源代码，开源时间定在 2018 年 Q1，届时会成立开发小组，不局限于 TrustNote 内部开发人员。同时，开发者可以贡献代码，经审核后由开发小组提交。
- 开发者社区：与开发者零距离互动，收集问题，发布技术文章和路线图，与开发者交流，融合世界先进的区块链技术。

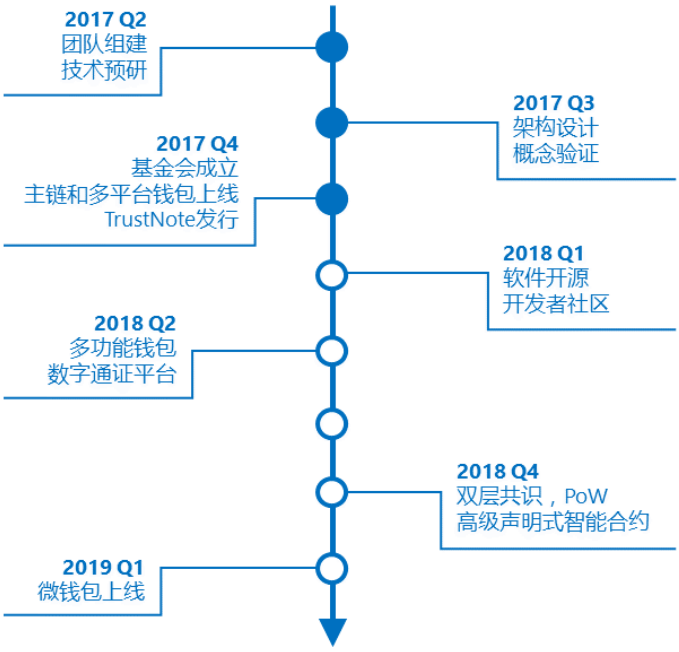

图 6-1 项目路线图

- 钱包革命：
  - ✓ 多平台：采用跨平台开发语言 Node.js 开发，支持多种平台 Windows、Mac、Linux、Android、iOS，浏览器 Chrome、Firefox 应用，支持在浏览器中安全的使用钱包软件；
  - ✓ 微钱包：TrustNote 团队还致力于物联端侧钱包的开发，将打造应用于物联端侧设备的微钱包协议和微钱包应用客户端，采用强调安全、并发、高效的系统编程语言 Rust 开发微钱包协议和钱包客户端，从根本上解决物联端侧设备计算、存储资源不足，设备之间无法完成自主价值交换的难题，赋能物联网。
- 数字通证平台：
  - ✓ 定制代币发行、分发、交易、销毁，管理数字通证的整个生命周期；
  - ✓ 去中心化币-币交易所，无需登记，匿名交易；
  - ✓ 支持比特币、以太坊、莱特币等主流数字代币钱包；
  - ✓ 用户发行的资产可以采用统一代币钱包（TrustNote 钱包），交易消

耗 TrustNote 基础代币 TTT，交易的实际费用即交易所占字节数；

- ✓ 采用双层共识机制，保证安全的同时，支持交易极速确认，并发量越高，交易确认时间越短；
- ✓ 第三方应用扩展，通过智能合约和第三方可信数据源提供的数据创建资产类应用使用场景，可以通过钱包应用市场下载扩展并使用。

图 6-2 数字通证平台框架图

数字通证平台聚焦于打造去中心化的数字通证创建、发行和运营综合平台。用户可以很方便地定制个人数字通证，轻松发起资产众筹或首次资产发售，无需编写复杂的智能合约代码，实现数字通证发行零门槛，也可以通过相应的数字通证网关安全地管理比特币、以太坊、莱特币等主流数字代币钱包，实现数字代币钱包软件跨操作系统和入口统一，无需再多个不同的钱包软件间不断切换。

- 典型应用场景：
  - ✓ 虚拟资产：游戏装备，直播打赏；
  - ✓ 条件支付：知识付费，API 调用，去中心化保险等；
  - ✓ 加密撮合：预测性市场，公益等；
  - ✓ 场外交易：币-币交易；
  - ✓ 社交交易：群红包，群收款；
  - ✓ 资源共享：充分利用闲置资源。

7 发行量和分发规则

- TTT 是 TrustNote 基础代币的简称，单位是 Mega Notes (MN)。
- 发行总量：1,000,000,000 MN，无增发，500,000,000 MN（占比 50%）为初始分发，投资兑换的方式分发；500,000,000 MN（占比 50%）为公证奖金，可挖矿获得。支持 PoW 挖矿的主链预计于 2018 年 Q4 正式上线，届时用户可以下载和使用挖矿客户端，并申请成为超级节点，通过积极参与主链共识获得公证权，通过发布有效的公证单元获得公证奖励。公证奖励策略为首年提供总公证奖金的 6.79%，之后公证奖金逐年衰减，衰减情况参见图 7-1。其中，公证奖金的 90%分配给提供有效公证单元的超级节点，公证奖金的 10%分配给基金会，用于社区运维、项目孵化、贡献者奖励等。

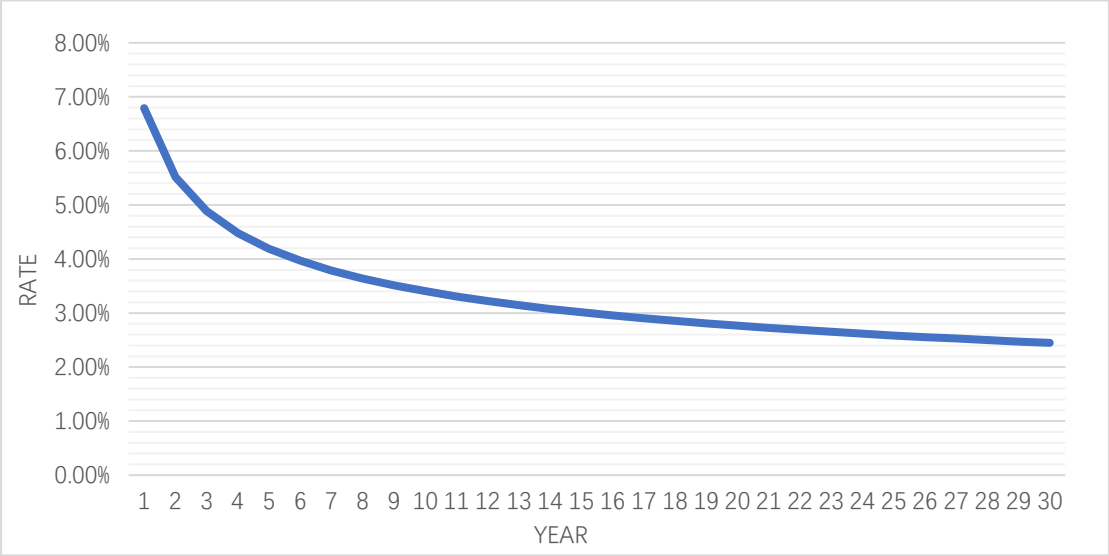

图 7-1 公证奖励比例

- TrustME-PoW 共识平均每 5 分钟一轮，每年共约 10 万轮，每轮公证奖金总量 = 公证奖金*90% + 单元公证费。
  - ✓ 第一年，每轮公证奖金约为 323.04 MN；
  - ✓ 第二年，每轮公证奖金约为 262.39 MN；
  - ✓ 第三年，每轮公证奖金约为 232.34 MN。

8 参考文献

[1] Bitcoin Computation Waste, http://gizmodo.com/the-worlds-most-powerful-computer-network-is-being-was-50403276. 2013.

[2] Bitcoin wiki. Proof of Stake. http://www.blockchaintechnologies.com/blockchain-applications. As of 11 Aug 2017.

[3] Coindesk.com. Bitcoin: A Peer-to-Peer Electronic Cash System.

[4] http://www.coindesk.com/ibm-reveals-proof-concept-blockchain-powered-internet-things/ As of 11 Nov 2017.

[5] Ethereum. Ethereum. https://github.com/ethereum/. As of 12 Nov 2017.

[6] IOTA. IOTA. https://github.com/iotaledger/. As of 10 Nov 2017.

[7] Byteball. Byteball. https://github.com/byteball/. As of 10 Sep 2017.

[8] Bernstein, Daniel J, et al. High-speed high-security signatures. Journal of Cryptographic Engineering 2.2(2012), 77-89.

[9] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, Louisiana, USA,1999, pp. 173–186.

[10] Biryukov, Alex, and D. Khovratovich. Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem. Network and Distributed System Security Symposium 2016.

[11] Gilad Y, Hemo R, Micali S, et al. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. The Symposium 2017, 51-68.

[12] C. Decker and R. Wattenhofer. Information Propagation in the Bitcoin Network. 13-th IEEE Conference on Peer-to-Peer Computing, 2013.

[13] D. Dolev and H.R. Strong. Authenticated algorithms for Byzantine agreement. SIAM Journal on Computing 12 (4), 656-666.

[14] A. Kiayias, A. Russel, B. David, and R. Oliynycov.. Ouroburos: A provably secure proof-of-stake protocol. Cryptology ePrint Archive, Report 2016/889, 2016. http://eprint.iacr.org /2016/889.

[15] S. King and S. Nadal. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake, 2012.

[16] S. Micali, M. Rabin and S. Vadhan. Verifiable Random Functions. 40th Foundations of Computer Science (FOCS), New York, Oct 1999.