WXChain(WTOT)白皮书

目录

1、系统的介绍及描述 1
2 权重 2
3 系统的稳定性以及割集 3
总结： 5
累积权重通常增长的有多快? 5
总结 8
4、可能的攻击场景 8
4.1 寄生链攻击 和新的 tip 选择算法 10
4.2 分裂攻击 12
总结 13
5.抵抗量子计算 13
6.经济模型： 14
6.1 WXC 14
6.2 WTOT 14
7.免责声明： 14

1、系统的介绍及描述

比特币的提出以及成功足以证明区块链技术对现实世界的意义。然而，比特币却因自身存在许多缺点，而导致其无法被作为通用的加密货币平台进行使用。一个显著的缺点是，比特币中任何一笔交易都需要交易费。然而，随着物联网的快速发展，小额支付将越来越重要，如果交易费用超过了小额支付自身的额度，这明显是不合理的。此外，对于比特币来说，交易费用作为区块产生者的一种激励机制，是无法被轻易移除的。另外一个问题是系统的异质性，对于目前现有的数字加密货币技术来说，其系统会有两类不同的角色参与者，一类交易的发起者，一类是交易的证明者。这种系统设计会对某些参与者造成不可避免的偏袒，从而导致冲突，这样一来，系统的需要消耗大量资源来解决冲突。基于上述问题，我们需要寻找更为合理的解决方案。

本文主要讨论的是一种不包含区块链技术的创新方法。当然，该方法 已经被专门为物联网产业 而设计的,我们称之为 WXCHAIN 的数字加密货币 所实现。而本文的核心目的是专注于 tangle 通用的特性，以及讨论如何解决 区块链系统在维护分布式账本时所出现的问题。当然，本文不会讲解 WXCHAIN 具体的协议实现。

正常情况下，一个以 tangle 作为基础结构的数字加密货币其特点如下：首先，这里不是全局单链的结构，取而代之的时 tangle 这种 dag 结构。由节点提出的交易将作为 tangle 图形集的基础。而 tangle 中，点与点间的边按照以下方式生成：当一笔新的交易到来时，它必需选择 tangle 中的两笔交易进行认证，因此，一条有向边代表一个交易对另一个交易的认证。当交易 A 并没有通过 有向边直接关联交易 B,而是同过至少两条或以上的有向边关联时,我们则称之为 A 间接认证 B。当然，tangle 中有且仅由一个“创世”交易，被剩余的所有交易直接或间接认证。“创世”交易的具体描述如下：作为 tangle 的第一个交易，所创建的所有 taken 被包含于一个 指定地址的账户余额中。而创世交易会将这些 token 分别发送到另外几个“基础”的账户地址中。另外，需要强调的是，所有的 token 仅在创世交易中被创建，所有别的交易都无法创建 token，换言之，挖矿者将无法通过挖矿获取奖励，即系统将不再有挖矿者的角色。

这里在对一些概念进行简短的说明：tangle 中的所有顶点都代表一笔交易。而 WXCHAIN 网

络则进一步由 WXCHAIN 节点组成，而节点的核心作用则是发起交易，并认证交易。

tangle 的主要交想法如下：用户在发起一笔交易前，必须先认证部分交易。因此，当一笔交易被提出来时，如果该笔交易可以按照 WXCHAIN 的协议将选定的交易认证成功，并且自身也是有效交易，则该交易会被认为对 WXCHAIN 网络的安全有贡献。否则，该笔交易将被视为无效交易，并且不会被别的交易选定认证。

当一笔交易获得越多的交易认证时，那么它的可信度就会越高。换言之，想要系统接受双花交易时非常困难的。需要强调的是，我们不会通过任何强制的规则来让节点 选定交易进行认证。当然，一种比较好的方式是，如果大量的节点去按照某种交易选择规则进行运行，那么对于任意别的节点，最好也按照这种规则运行。这似乎是一个合理的假设，尤其是运行在带有预装固件的专用芯片 的 WXCHAIN 网络节点。

下面则是节点详细交易流程：

- 节点在发起一笔交易前，必须选择 tangle 中已有的两笔交易，并按照规定算法进行认证。
- 节点必须检查这两笔选定的交易不存在冲突，并确定被选择交易也不是无效交易。
- 节点在发布一笔有效交易前，需要进行类似比特币中的难度确认问题，即微型的 pow。其核心是通过 pow 运算，从而找到一个 nonce 值，该 nonce 的 hash 值 与自己交易中的某些数据组合起来时，具有一个特定的形式。

  需要注意的是 ，整个 WXCHAIN 网络都是异步的。通常来说，节点间并不需要看见相同的交易集，因为有些交易是无效交易。一方面，节点不必对 哪些是有效交易可以加入账本达成共识，这意味着所有有效的交易都可以加入 tangle 中。另一方面，当节点遇到两笔有冲突的交易时(如双花交易)，必须决定那笔交易是无效的。选定的具体规则如下：节点运行 tip 选择算法多次
  (cf. Section 4.1)，并观察这两笔交易中，那笔交易更有可能被直接认证。例如，在运行 100 次算法后，其中一笔交易被选择了 97 次 ，那么，我们则认为该笔交易的可行度为 97%。

  我们提前来讨论一下 4.1 节尾部所给出的评论：节点传播交易的动机是什么？首先，每个节点自身都会统计一些数据，其中一些数据就是有多少新交易是从邻居节点接收的。因为如果一个节点“非常懒惰”，就会被别的节点从邻居节点中删除。

  接下来，我们将在第二部分来详细说明 tip 选择算法，通过第三部分来讲解如何判断交易确认的，在第四部分来分析可能的攻击场景。

2 权重

在该部分中，我们将详细介绍交易权重的定义及其相关概念。交易权重与提出该交易的节点对其投入的工作量成正比。在 WXCHAIN 的当前的实现中，具体的权重值被假定为 3^n,其中 n 为非空的并可接受的整数值(具体请参考第四部分的分析)。目前，对于权重的详细计算是无关紧要的，我们只需要知道，权重大的交易笔权重小的交易更“重要”。为了避免垃圾邮式以及别形式攻击。我们假定没有一个节点能够在短时间内生成大量具有“可接受”权重的交易。

另一个我们需要讲解的概念是交易的累积权重:具体的定义为所有直接或间接认证该交易的权重以及自身权重之和。我们举 Figure1 中的一个交易作为例子。例子中的交易 F，它被交易 A、B、C、E 直接或间接认证，其累积交易权重为 9,具体的计算方式为 9 = 3(F) + 1(A) + 3(B) + 1(C) + 1(E)。

我们将所有未被认证的交易定义为“tip”。如 Figure1 中 before 用例中，A、C 均为 tip。然后，当一笔新的交易 X 到来，并成功认证了 A、C(Figure A 中的 before)，那么，更具累积权重的定义，除了交易 X 的所有交易的累积权重都增加 3。

我们在来介绍认证算法中与交易有关的另外两个参数：

- height:指定交易到创世交易的最长路径的长度。
- depth:指定交易 到 tip 交易的最长路径长度。
  我们已 Figure2 中的用例讲解。用例中，G 的 height 为 1（gensis->G） ，depth 为

4(G->D->B->A)。
接着介绍另外一个定义 score，代表着由该笔交易直接或间接认证的所有交易的权重之 以及自身权重之和。如 Figure2 中的 score(A) =1(A)+3(B)+1(D)+3(F)+1(G) = 9。

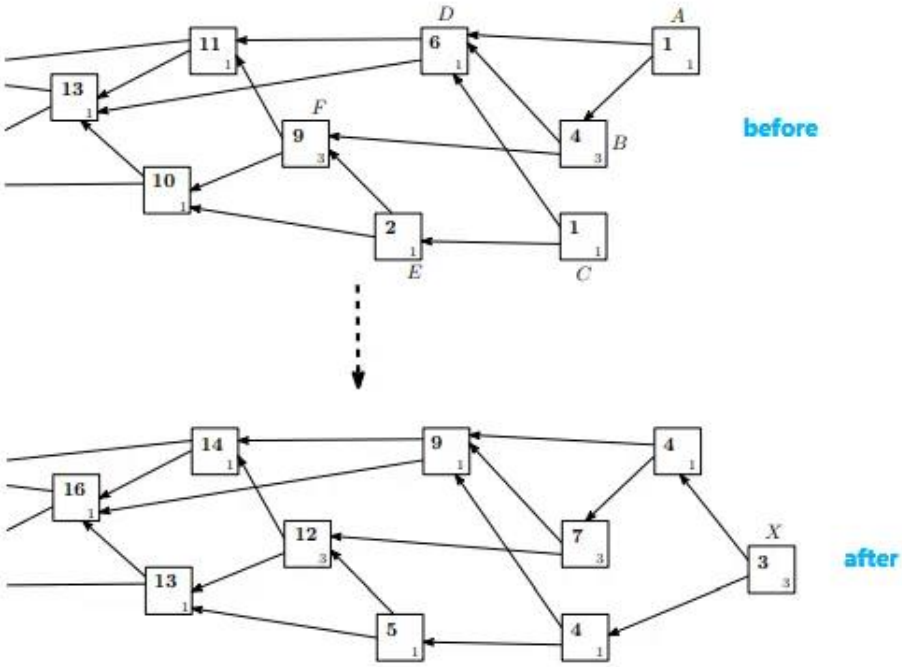

Figure1

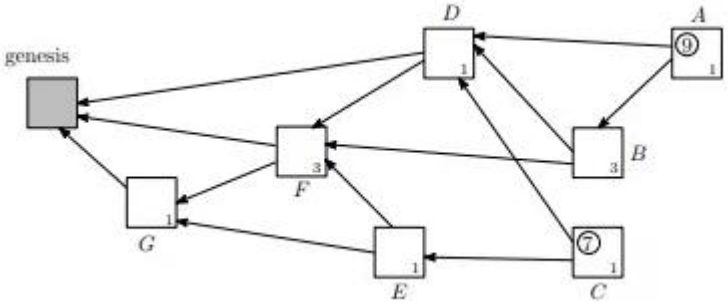

Figure2

3 系统的稳定性以及割集

我们定义 L(t)为，系统在时刻 t 所包含的全部 tips 总量。一方面，我们所期望的是，随机过程 L(t)可以保持稳定，更准确的说，是希望这个过程是积极的循序，具体的定义请参阅引用[11]的 4.4 和 6.5 节。特别地，正递归式表明，当 t→∞时，P[L(t) = k]的极限应存在，且当 k≥1 时，其值均为正。直观上，我们期望 L(t)应该在一个常数值附近波动，而不是趋向无穷大。当然，如果 L(t)的值趋向正无穷，这样一来，就会由许多 tip 交易将永远得不到认证。

为了分析 L(t)的稳定性，我们需要做一些假设。其中一个假设是，交易由大量独立的个体发出，因此，节点接受交易的流程可以用泊松点过程进行模拟(参见[11]的 5.3 节)。我们定义 λ 作为整个 泊松过程 速率，即平均的单位时间交易到达率。为了简单起见，我们假设这个速率在时间上保持不变。同时，假设所有设备的计算能力大致相同，并定义 h 为节点完成交易传播所需的平均耗时。然后，我们继续假设所有节点的行为如下:在发起交易前，节点需要随机选择两个 tip 并认证。这种

情况下，整个 WXCHAIN 网络并没有提供保护机制来抵御“惰性”或恶意节点(参见下面的 4.1 节)。因此，出于该模型相对简单，我们可以考虑更多复杂的 tip 选择算法来运行系统并持续观察之。

接下来，我们将做一个进一步的简化假设，即任何节点发起的交易，只有经过了 h 个单位时间后，才会被别的节点看见。同时，我们假设 tip 数量随着时间的推移也同样大致保持稳定，并且集中数字 L0 周围。

按照上述假设，我们来推导 L0 与 λ 和 h 的数学关系：给定时间戳 t，那么系统将由 λh 个 tip 是“不可见”的；同样，假定这里有 r 个“可见”tip(在 t-h 前加入 tangle 的交易，并且在时刻 t 任然是 tip 交易)，因此，可以得到 L0 = r + λh。接着，在时刻 t，有一笔新的交易到来，然后选择两笔交易进行认证，那么，选择到 tip 交易的概率为 r/(r+λh)(更具上述假设，r 才是正真可见的 tip，尽管节点会认为 λh 也是 tip)，所以，选择到 tip 的数量为 2r/(r+λh)。根据模拟的观察结果，在平稳状态下，所选 tips 的平均数量应该等于 1，因为平均而言，新交易不应该改变 tips 的数量。因此，解方程 2 r (r +λh) = 1 / r，得到 r =λh，进一步得到

$$L_0 = 2\lambda h. \qquad (1)$$

另外，我们还注意到，如果如果规则改为 新交易必须选择 K 个交易，而不是两个交易进行认证，那么将给出类似的计算(该公式具体的推导方式没给出，权且按照结论阅读)：

$$L_0^{(k)} = \frac{k\lambda h}{k-1}. \qquad (2)$$

当然，符合上述公式的前提条件是 $L_0^{(k)}$ 中的 λh as k → ∞。(基本上，剩下的 tip 对 WXCHAIN 网络是未知的)。

我们继续回到每笔交易只需选择两笔交易进行认证的情况进行讨论，这么一来，交易首次被认证的预估时间 为 h = L0/2λ。

需要注意的是，给定的任何时间 t，时刻 s(s∈[t, t + h(L0, N)]) 所包含的 tips 交易集通常会构造成一个割集。而任意从时间 t 以后所发出的交易到创世交易的路径都必须经过这批集合。而割集偶尔会变小，然后，我们可以使用这个变小的割集作为 DAG 的修剪工具以及任务检查点。

需要明确的是，上述的完全随机 tip 选择策略在实际运用中不是很好，因为它并不鼓励选择 tip 交易。而“懒惰”用户经常会选择旧交易进行认证，但这种行为对整个 WXCHAIN 网络是没有贡献得。此外，恶意节点可以通过发起大量交易来选择指定的交易对进行认证，从而使指定的交易会有更高的概率被选定，而“诚实”节点的 tip 交易将会被抛弃，这样一来，就会 tangle 就会按照作恶节点的意图发展。为了避免这类问题，我们必须采取一种偏向于“更好”tip 的策略。在下面 4.1 节中，会用一个例子来介绍这种策略。

在开始讨论交易在第一次获得认证的预期时间之前，我们需要区分状况。

- 低负荷：如 Figure 3 中的顶部用例，tip 数目很少，通常是 1。这种情况基本发生在交易的单位时间到达率非常低的情况下。此外，如果网络延迟非常低，且节点计算速度也很快的情况下，也不会有很多 tip。该情况下，新到的交易基本不会选择相同的 tip 进行认证。
- 高负载：Figure3 中的底部用例，有大量的 tip 交易。这种情况基本发生在交易到达率很高，网络延迟高且节点计算速度慢的情况下。这种情况下，不同交易可能会选择相同的 tip 进行认证。

当然，上述划分方式只是为了分析在极端情况下，系统会做出何种表现。

低负荷状态下的情况相对简单,交易在第一次获得认证的预期时间为 λ^{-1} 。

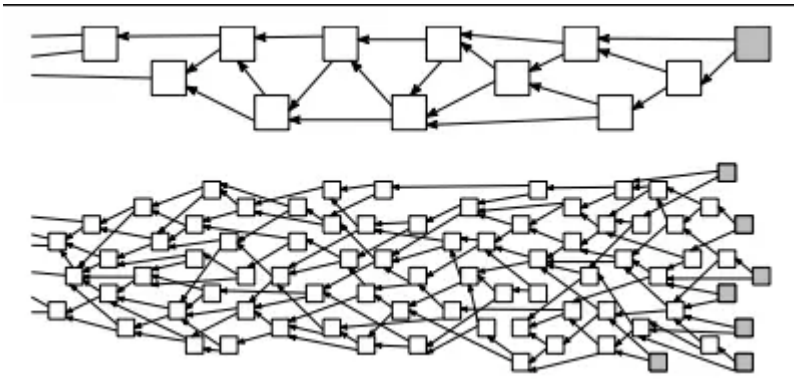

Figure3

现在让我们考虑高负载情况，即 L0 很大的情况。正如上面提到的，有人会认为泊松过程的交易选择是独立户不影响的，其近似值为 2λ/L0。因此，交易首次获得认证的预估时间是 L0/(2λ) = h (recall (1))。

然而，值得注意的是，对于被动的等待长时间直到交易被别的交易选定认证，这种方式并不是一个好的策略。因为“更好”的 tip 会出现，而且会更被偏向选定。相反，当一个交易在等待被选定的时间间隔 L0/2λ 大得多时，一个好的 tip 选择策略是需要促进这类交易被空交易选定(空交易时指不会发生 token 转账的交，该类交易有助于整个 WXCHAIN 网络)。换言之，网络种的任意节点都可以发起一笔空交易来选定并认证一些落后的交易，这样一来，这些落后的交易就有更大的机会获得确认。

事实证明，基于 heights 以及 scores 的交易选择策略更容易受到特定类型的攻击，详细参见 4.1 节，当然，该小节也会详细讨论防范此类攻击的策略。与此同时，一些简单的交易选择策略任然是值得思考的。因为简单的策略是容易分析的，并从中获取出 tangle 行为种定性 和定量的特征。

总结：

1.我们就低负荷和高负荷这两种状态讨论了 tangle 的行为。

2.在低负荷的情况下，tip 会非常少，一个 tip 首次获得认证的时间为 $\Theta(\lambda^{-1})$,其中 λ 为单位时间交易的到达率。

3.在高负载的情况下，tip 的数据取决于具体的交易选择策略。

4.如果我们使用完全随机选择策略，这对于 tip 的数量数量最大化是一个最优解，但这种策略是不实际的，因为它不鼓励 tip 被选择。

5.而复杂的 tip 选择策略需要处理各种各样的攻击。这类策略将在 4.1 节讨论。

6.在高负载的情况下,tip 首次被认证的时间为 Θ(h),其中 h 为 一个节点（算力/交易传播时间 ）的平均值。然而，当一个交易在超过了这个时间后还没有被认证，该交易的发起方或接受方最好使用一个空交易来促进该交易被选定认证。

累积权重通常增长的有多快?

我们假定整个 WXCHAIN 网络处于低负载状态。它的累积权重的增长速度为 λ，因为所有 新到来交易都会间接或直接认证它。

当网络处于高负载的情况下，较老的交易其累积权重的增长速度同样为 λ，因为基本上所有的交易基本上都会间接认证较老的交易。此外，当交易第一次加入到 tangle 时，它可能需要等待一段时间才被选定认证。在这个时间间隔内，交易的累积权重表现为随机方式。为了描述交易在获得

多个认证后其累积权，我们将将定义 H(t)为 指定交易在时间 t 时的预期累积权重,K(t) 为在时间 t 时，认证指定交易的预期 tip 数量，而 h := h(L0, N)。另外，我们在做了一个简单的假设，随着时间的推移，tip 的数量会大致保持不变，值为 L0。并且，我们继续假定每个交易选择两笔交易进行认证。预计其它合理的策略其定性行为大致相同。

我们大致回顾一下，处于节点在发起一笔交易前，必须先进行计算即认证，所以一个在时间 t-h 选择两笔交易认证的交易会在 时间 t 进入到整个 WXCHAIN 网络。综上，我们可以推导出，该笔交易所选定的认证交易中，至少有一笔时 tip 的概率为

$$1-(1-\frac{K(t-h)}{L_0})^2=\frac{K(t-h)}{L_0}(2-\frac{K(t-h)}{L_0})^{16}$$

。类似于引用[11] 中的示例 6.4，我们可以假定 δ>0：

$$H(t+\delta)=H(t)+\lambda\delta\frac{K(t-h)}{L_0}\Big(2-\frac{K(t-h)}{L_0}\Big)+o(\delta),$$

从而推导出以下微分方程：

$$\frac{dH(t)}{dt}=\lambda\frac{K(t-h)}{L_0}\Big(2-\frac{K(t-h)}{L_0}\Big). \qquad (3)$$

为了使用推导的公式(3),我们首先需要计算的是 K(t)。这可不是一项简单的任务,因为在时间 t-h 的是 tip 的 交易，可能在时间 t 就不是了 tip 了，并且，新到来的交易如果认证了 该交易，则认证该交易的总 tip 数加 1。据实验观察，在时间 t-h 的为 tip 的交易，在时间 t 仍然是 tip 的概率为 1/2（为了验证这一点，回顾一下第 3 节的讨论，正常情况下，任意时刻的 tip 数量为 λh，在时间间隔 h 内，新的 λh tip 将取代旧的 2λh 中的一半）。因此，在时间 t， 将近 K(t-h)/2 的 tip 任然是未被认证的状态,而剩下的 K(t-h)/2 获得了至少一个以上的新交易认证。设定 A 表示在时间 t-h 时, 数量 K(t-h)/2 的 tip 在时间 t 仍然为 tip 的 集合，B 为在时间 t-h 时，K(t-h)/2 为 tip，在时间 t 不为 tip 的集合。我们定义 p1 为，一个新到来的交易，至少从集合 B 选择一个，而不从集合 A 选择交易的概率；P2 为全部从集合 A 选择的概率。换言之，P1 和 P2 代表着当新交易到来时，当前 tip 数量是加或减 1 的概率。据定义，我们可以得到

$$p_1=\Big(\frac{K(t-h)}{2L_0}\Big)^2+2\times\frac{K(t-h)}{2L_0}\Big(1-\frac{K(t-h)}{L_0}\Big),$$
$$p_2=\Big(\frac{K(t-h)}{2L_0}\Big)^2.$$

为了得到第一个表达式，我们解读上图的公式，可以得到 p1 为 p2 加上 两倍于 第一个 tip 属于集合 B 并且第二个 tip 不属于 A ∪ B 的概率。类似于推导公式(3),得到 K(t)的微分方程：

$$\frac{dK(t)}{dt}=(p_1-p_2)\lambda=\lambda\frac{K(t-h)}{L_0}\Big(1-\frac{K(t-h)}{L_0}\Big). \qquad (4)$$

由于精确的解方程(4)式困难的，所以我们进一步简化假定。首先，根据实验观察，当 K(t) 达到固定 εL0 水平后,其中 ε>0,它将很快增长至(1-ε)L0。所以，当 K(t) 小于期望 L0 时，这时，公式(4)中 K(t-h)/L0 约为 0，而 λh/L0 =1/2，因此，可以得到进一步的简化公式：

$$\frac{dK(t)}{dt}\approx\frac{1}{2h}K(t-h), \qquad (5)$$

边界条件 K(0)=1。我们寻找 K(t) = exp(ct/h)的解，将该等式代入(5),得到

$$\frac{c}{h}\exp\left(c\frac{t}{h}\right) \approx \frac{1}{2h}\exp\left(c\frac{t}{h} - c\right),$$

因此：

$$K(t) = \exp\left(W\left(\tfrac{1}{2}\right)\frac{t}{h}\right) \approx \exp\left(0.352\frac{t}{h}\right) \qquad (6)$$

这是一个近似解，其中 w(•)是所谓的 lambert w 函数(也称为 ω 函数或积对数,对于 x∈[0，+∞)，x = W(x) exp(W(x))。)。

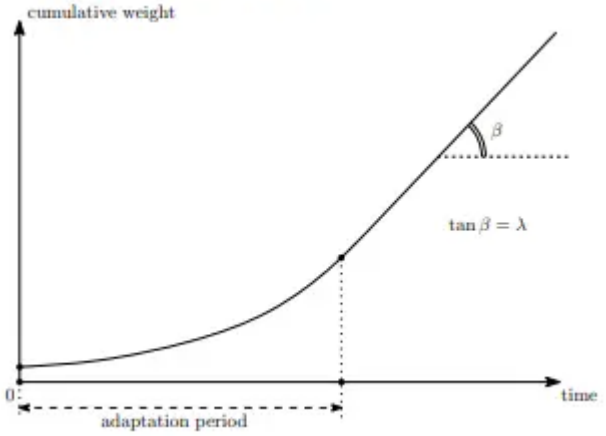

Figure4

取推导函数(6) 两边的对数，我们发现 K(t)达到 ε L0 的大致时间是

$$t_0 \approx \frac{h}{W\left(\frac{1}{2}\right)} \times \left(\ln L_0 - \ln \varepsilon^{-1}\right) \lesssim 2.84 \cdot h \ln L_0. \qquad (7)$$

回到方程(3)，我们继续移除方程中最右边的式子，在“adaptation period” 范围内(t<= t0,t0 为方程(7)中所求值)我们可以得到

$$\begin{aligned}\frac{dH(t)}{dt} &\approx \frac{2\lambda}{L_0}K(t-h)\\ &\approx \frac{1}{h\exp\left(W\left(\frac{1}{2}\right)\right)}\exp\left(W\left(\tfrac{1}{2}\right)\frac{t}{h}\right)\\ &= \frac{2W\left(\frac{1}{2}\right)}{h}\exp\left(W\left(\tfrac{1}{2}\right)\frac{t}{h}\right)\end{aligned}$$

所以

$$H(t) \approx 2\exp\left(W\left(\tfrac{1}{2}\right)\frac{t}{h}\right) \approx 2\exp\left(0.352\frac{t}{h}\right). \qquad (8)$$

另外，根据 Figure6，在“adaptation period(适应其)”过后，累积权重时间函数 H(t) 随 λ 线性

增长。
我们强调方程(8)中的“指数增长”并不意味着在“adaptation period(适应其)”期间，累积权重会迅速增长，相反，它的行为如图 Figure 4。

总结

1.在低负荷的情况下，交易被多次选定认证后，其累积权重的增长速度为 λw，而 w 为一般交易的平均权重。
2.在高负载的情况下，这里有两个增长阶段。首先，在阶段“adaptation period”范围内，累积权重的增长速度为上述推导的方程(8),通过“adaptation period”后，它的增长速度又回到 λw。事实上,对于任何合理的 tip 选择策略,交易的累计权重增长速度将在“adaptation period”结束后，都会回到 λw，因为所有新到交易都会间接地认证交旧的交易。
3.我们可以将交易的适应期 的结束看作 被当前大部分交易间接认证的开始。具体的适应期时长为方程(7)所推导的值。

4、可能的攻击场景

我们首先讨论一个攻击场景，攻击者试图单独“超过”WXCHAIN 网络。
1.攻击者向商户发起一笔支付交易，当该笔交易在累积足够大的权重后，商户在货物发给攻击者。
2.攻击者发起一笔“双花”交易。
3.攻击者利用自己的计算能力发起许多小额交易，这些小额交易选定并认证该笔“双花”交易，但不直接或间接认证发给商家的原始交易。
4.攻击者拥有大量 不需要认证 tip 的 sybil 身份。
5.与第 3 项相类似的攻击式是使用他们所有的计算能力来发布一个更“大”的“双花”交易。这个“大”是指该笔交易自身的权重非常大，这样会导致在原交易被确认前，该笔“双花”交易被提前确认。

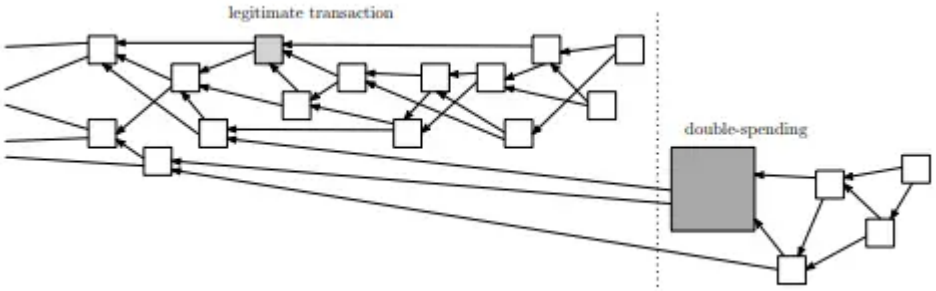

Figure5

6.攻击者希望他们不诚实的 子 tangle 会超过 诚实的 子 tangle。如果这种情况发生了，tangle 主网将继续在“双花”交易 的分支上发展，而原先合理的交易 所在的分支就会成为孤儿(Figure5)。事实上，一个大权重的“双花”交易策略增加了攻击者的成功几率。而且，在这个“理想”的数学模型下，这种攻击总是成功的。

假设 是获得一个 nonce 的所需时间，该 nonce 使得双花交易至少具有 3n 的权重；并且是一个具有参数 μ/3^n 的指数分布随机变量，其中 μ 表示攻击者的算力。

我们继续作出以下假设，在原交易发起的 t0 个单位时间后，商户在原交易的累积权重至少为 w0 时，承认该笔交易。因此，在正常情况，可以可以合理预估，原交易的累积权重其增长速率为 λw，而 λ 则是网络上诚实节点发起交易的总单位时间交易到达率，w 为一般交易的自身权重。所在，该交易在 t0 时刻，其累积权重为 s w1 = λwt0。

我们定义 $\lceil x \rceil$ 为大于 x 的最小正整数，定义 $n_0 = \lceil \frac{lnw_1}{ln3} \rceil$，所以，3^n0 >= w1(如果 w1 非常大，两者将约等)。如果攻击者试图在 t0 时间间隔内获得一个 将赋予双花交易至少为 3n0 权重的 nonce 成功，那么该次攻击会成功。而该事件发生的概率为：

$$\mathbb{P}[W^{(n_0)} < t_0] = 1 - \exp(-t_0\mu 3^{-n_0}) \approx 1 - \exp(-t_0\mu w_1^{-1}) \approx \frac{t_0\mu}{w_1}.$$

上述方程中的约等时合理的，当 t0μ/w1 较小时。如果这种“直接”攻击没有成功，攻击者将继续寻找可以赋予 权重 3^n (n>n0)的 nonce，如果寻找成功，合理分支的总权重将小于 3^n。而这种事件发生的概率为

$$\mathbb{P}[\lambda w W^{(n)} < 3^n] = 1 - \exp\left(-\mu 3^{-n_0} \times (3^{n_0}/\lambda w)\right) = 1 - \exp(-\mu/\lambda w) \approx \frac{\mu}{\lambda w}.$$

也就是说，虽然 μ/λw 通常是一个很小的数，但是，对于每一个“级别”n，攻击成功的概率时恒定的。而成功的时间大约是 3^(λw/μ)。虽然，攻击所需时间非常大，但“第一次”攻击的成功概率是不能忽略的。因此，我们需要对策。其中一种对策是将交易自身的权重加以限制，甚至将其设置为常量。可正如第 3 节中所提到的，后者可能不是最佳解决方案，因为它没有提供足够的保护机制来防止垃圾邮件攻击。

现在，让我们来讨论交易自身权重设置为常量 1 的情况
，并估计攻击成功的概率。我们假设某个给定的交易自发起到 t0 个时间单位后，其累积权重为 w0，并且该交易已经过了适应期。在这种情况下，交易的累积权重增长速率为 λ。现在，假设攻击者相要在该交易发起双花攻击。为此，攻击者秘密的准备该双花交易，并开始生成无意义的交易，在商户准备认可原实交易前，这些无意义的交易开始认证双花交易。如果双花交易累积权重所超过原交易的累积权重，则攻击成功。如果累积权重没有超过，那么双花交易将得不到别的交易选定认证，因为在原交易累积权重更大的情况下，基本所有新到来的交易都会间接选定认证原交易。这种情况下，双花交易将被孤立起来。

和以前一样，μ 代表攻击者的算力。同时，我们还简化了一个假设，即交易可以立即传播。假设 G1 G2 G3…表示独立同分布。V1 V2 V3…也是独立同分布。指数随机变量为 μ，定义 ，其中 k≥1。

假设在 t0 时，商家决定认定累积权重为 w0 的交易。我们继续估计双花交易攻击成功的可能性。我们定义 M(θ) = 1/(1-θ) 为参数为 1 的指数分布的矩母函数(详细参见引用[14] 的 7.7 节)。对于 α ∈ (0, 1) 的情况下，可以得到：

$$\mathbb{P}\left[\sum_{k=1}^{n} V_k \le \alpha n\right] \approx \exp\left(-n\varphi(\alpha)\right), \qquad (9)$$

ln M(θ) 的 勒让德 转变 为 ϕ(α) = − ln α + α − 1。作为一般事实，当 α ∈ (0, 1) 时，ϕ(α) > 0 。回想一下，参数为 1 的指数随机变量的期望值也等于 1。

假定 μt0/w0 < 1，否则，攻击者的 subtangle 最终超过 合法 subtangle 的概率将接近 1。现在，为了在时间 t0 时，其指定交易的累计权重超过 w0，攻击者需要在 t0 期间，发出至少 w0 笔 自

身权重为最大值 m 的交易。因此，使用方程(9),我们发现双花交易在 t0 时刻可以获得更大累积权重的概率大致为

$$\begin{aligned}\mathbb{P}\Big[\sum_{k=1}^{w_0/m} G_k < t_0\Big] &= \mathbb{P}\Big[\sum_{k=1}^{w_0} V_k < \mu t_0\Big] \\ &= \mathbb{P}\Big[\sum_{k=1}^{w_0} V_k < w_0 \times \frac{\mu t_0}{w_0}\Big] \\ &\approx \exp\big(-w_0\varphi\big(\tfrac{\mu t_0}{w_0}\big)\big). \qquad (10)\end{aligned}$$

因为上述方程所计算的概率值很小， w0/m
的值需要非常大，而 ϕ(μt0/w0) 的值又不能太小。

需要注意的是，在时间 t>=t0 时，合法交易的累积权重大约为 w0 +λ(t-t0)，因为我们认证适应期已经结束，所以累积权重的增长速率为 λ。

通过代入方程(10),我们可以得到双花交易在 t>=t0 时，其累积权重超过原交易的概率大致为

$$\exp\big(-(w_0+\lambda(t-t_0))\varphi\big(\tfrac{\mu t}{w_0+\lambda(t-t_0)}\big)\big). \qquad (11)$$

然后，如果累积权重的增长速率小于 μt0/w0≥ μ/λ，则可以推导出 μt0/w0≥ μ/λ。可以看出，双花攻击成功的概率时有规律的

$$\exp\big(-w_0\varphi\big(\max(\tfrac{\mu t_0}{w_0}, \tfrac{\mu}{\lambda})\big)\big). \qquad (12)$$

例如，在 μ=2, λ=3 的情况下，攻击者的算力相比于整个网络时非常小的。假设在时间 12 之前，交易的累积权重为 32。可以计算出 max(μt0/w0,μt0/w0) = 3/4,ϕ(3/4)≈ 0.03768, 方程(12)给出的上界约为 0.29。如果假设 μ=1 并保持所有其他参数不变，然后可以计算出 max(μt0/w0,μt0/w0) = 3/8,ϕ(3/8)≈ 0.3558, 方程(12)的大约值为 0.00001135,可见变化相当大。

根据上面的讨论，我们可以得到不等式 λ>μ 成立的情况下，对于整个网络来说是安全的。换言之，诚实的交易总量需要大于攻击者的算力，否则，方程(12) 的预估量是无效的。因此出于安全判断，系统需要通过检查 checkpoint 来到达目的。

当我们使用累计权重作为决定两个冲突交易交易哪个是有效的度量决策时，必须十分小心。这是因为通过累计权重的方式，系统可能收到于 4.1 节所描述的类似攻击，即攻击者提前通过一个双花交易，并通过大量的交易来选定该双花交易，使得双花交易的累积权重超过原交易，然后在商家接受原交易后，使得双花交易生效并广播。我们将在下一节提出一个能更好解决冲突交易的方法。

4.1 寄生链攻击 和新的 tip 选择算法

我们继续回到 Figure6 中的攻击模式：攻击者秘密的构建一个子 tangle，偶尔引用主 tangle 从而使自己获得更高的 score。这里再说明一下，score 代表着由该笔交易直接或间接认证的所有交易的权重之 以及自身权重之和。由于网络延迟对于攻击者构建子 tangle 来说不是问题(这是因为攻击者总是可以随便选定并认证自己的事务，而不依赖于来自网络其余部分的任何信息。)，因此，如果他们使用足够强大的计算机，他们可能能够为子 tangle 的 tip 提供更高的 score。此外，攻击者通过广播大量新交易来人为地增加 tip 数量，这些新 tip 会选定认证他们先前在寄生链发布的交易(如用

例 Figure 6)。因此，在诚实节点使用简单的 tip 选择算法的情况下，攻击者则更容易成功。

为了抵御这类攻击，我们打算使用一个已有的条件，即主 tangle 应该比 攻击者有更多的有效 hash 算力，因此，相对于攻击者的更过交易，主 tangle 能够产生更大的累积权重。核心是使用 mcmc 作为 tip 选择算法。

定义 Hx 为一笔交易的累积权重,在交易自身权重为 1 的情况下,tip 交易的累积权重为 1，非 tip 交易的累积权重至少为 2。

而 MCMC 主要是一种 random walker tip 选择算法(这里的随机数由节点自身提供的伪随机生成器生成)。然后，通过该算法选出的 tip 将被 candidates 选定认证。该算法的工作流程大致如下：

1.考虑区间[w, 2w] 上所有的交易,其中 w 相当大(主要的考虑是,walker 所在交易位置必须足够深 ，防止 walker 一下就移动到 tip,当然,又不能太深,考虑到要在合理的时间内选定 tip。而区间[w,2w]是武断的，当然也可以是[w,5w]等，甚至直接将 walker 放在创世交易上。当然，可以通过时间区间的方式来确定 walker 的位置，例如，可以将 walker 置于在时间区间范围[t0, 2t0]到达的交易。）

2.单独将 N 个 walker 放在 1 中的 的区间交易所在位置上。(首先，这个 N 的选择也是武断的，但主要考虑到如果选择一两个 walker ，那么他们有可能会跑到攻击者做构建的 tangle 上。）

3.让这些 walker 独立随机地向 tip 移动。

4.向 walker 移动到 tip 时，表明该 tip 被选定认证。然而，我们需要添加一个规则，即那些太快移动到 tip 的 walker 需要被抛弃，因为他们可能移动到“lazy tip”上。

5.walker 移动到下一个位置的概率定义如下：如果 y 直接认证 x (y ~>x),则 Pxy 代表 x 移动到 y 的概率，计算方式为

$$\exp\left(-\alpha(\mathcal{H}_x-\mathcal{H}_y)\right), \text{ that is}$$

$$P_{xy}=\exp\left(-\alpha(\mathcal{H}_x-\mathcal{H}_y)\right)\Big(\sum_{z:z\rightsquigarrow x}\exp\left(-\alpha(\mathcal{H}_x-\mathcal{H}_z)\right)\Big)^{-1}, \qquad (13)$$

上述方程中， α >0,并且需要选定一个值。

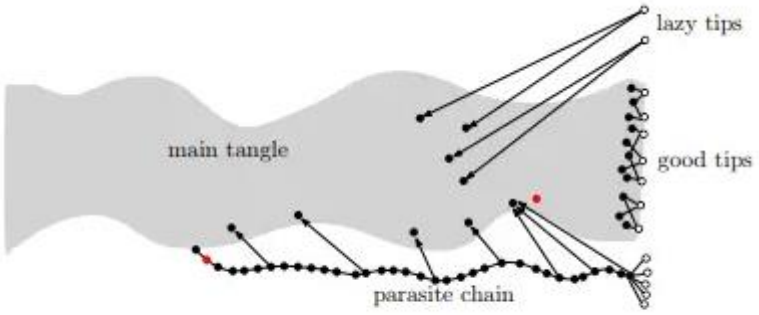

Figure6，该用例为主tangle以及寄生链共存的例子，红点为双花交易

需要注意的是，该算法是“局部的”，换言之，整个 walker 游历的过程无需从 Genesis 开始，也不需要计算整个 Tangle 的累积权重，只需计算从 walker 的起始点到终点所历经交易的权重之和。

为了检查算法是否按照预期工作，首先需要考虑的是“lazy tip”。这类 tip 可能有意选定认证一些旧得交易，以避免校验工作。即使 walker 最终在 lazy tip 上，该 lazy tip 也基本不可能被新到来得交易选定，因为它得累积权重相对来说太小。

接下来，我们来考虑一种交替得攻击模式：攻击者将秘密地构建一条包含将某个用户的余额

全部转移到指定用户这一笔交易的链，如 Figure 6 中，parasite chain 的红点交易。然后，攻击者在主 tangle 上发起一笔带商家接受的正常交易(Figure 6,main tangle 中的红色交易)，并且，寄生链上的交易偶尔会引用主 tangle 上的交易。然而，寄生链的累积权重并不是很大。需要注意的是，寄生链上的交易不在引用主 tangle 中红色交易之后的交易。此外，攻击者在发起正真的攻击前，会人为地增加寄生链上的 tip 交易。而攻击者的目的很简单，让寄生链成为主 tangle，而原来的主 tangle 则被抛弃。

可以明显地观察出 mcmc 选择算法选择攻击者发起的 tip 的概率不高，推论于 lazy tip 的场景相同:总的来说,寄生链上的交易累积权重会笔主 tangle 上交易累积权重要小的多。因此,walker 在移动的过程中,基本不可能会走到寄生链上,除非 walker 一开始就在寄生链上,但这也不太可能,因为主 tangle 中包含更多的位点。

作为一种额外的保护措施，我们可以首先使用一个较大的 α （这样一来，它实际上是“几乎确定性的”)来运行 random walk 算法来选择 tip;然后在使用一个较小的 α 来运行 random walk ，以从来验证两者的所选的 tip 是否一致。

同时观察到，对于总是向 tip 移动的 random walk，使用简单的递归计算出口概率分布是非常简单和快速的：这是我们不希望节点做的事情。然而，我们可以通过以下方式修改我们的方法:在每一步中，random walk 可能会往回走,并且我们可以假设往回走的概率例如为 1/3 等。无论如何，步行会很快到达终点(因为它会向终点漂移)，但计算出口尺度并不容易。

让我们讨论一下为什么节点会遵循这个算法。从第 1 节开始回忆，可以合理地假设至少有“良好”比例的节点将遵循引用算法。此外，由于计算和网络延，tip 选择算法更偏向处理交易发起时刻的 tangle 快照。出于我们在后续部分中详细解析的原因，将快照时间点移动到一个更早的时刻可能是一个好主意(首先，random walker 找到了快照的一个 tip，然后它继续向当前 tangle 的“实际”tip 走去。)。设想一个“自私”的节点，它只想最大化其交易 被快速选定认证的机会。本节的 MCMC 算法定义了一组 tips 的概率分布，该算法被相当比例的节点所采用。显然，自私的节点自然会首选获得最大分布值的 tip。然而，一个合理的假设是，如果其它自私的节点也按照该方式运行，并且使用相同的策略,那么他们都将失败。许多新交易将在大致相同的时间取选定认证 两个相同的 tip，因此，这些交易在随后被选定中，会产生许多竞争。另外，由于节点使用的是过去的快照，因此，大量交易选定相同的交易所导致的累积权重增加并不会被立即“感觉到”。因此，即使是一个自私的节点，也必须使用一些随机的具有概率分布的 tip 选择算法，并且其概率分布需要接近于 mcmc 所产生的概率分布。我们并不声称这种“聚合”的概率分布将等于自私节点存在时的默认概率分布。然而，上面的论点表明它应该接近它。这意味着许多节点试图验证相同的坏 tip 的怪率任然很小。无论如何，节点自私的动机并不大，因为获取的收益不大。这与其他分散化结构(如比特币)本质上是不同的。重要的事实是，节点没有理由放弃 MCMC tip 选择算法。

我们想指出，方程(13)中给出的转移概率的定义并不是一成不变的。我们可以用另一个函数来代替指数函数，这个函数可以快速减小，比如 f(s) = 1/s^3。W 和 N 的选择是随意的。在这一点上，目前还不清楚是否有任何理论论据确切地表明这些参数应该以何种方式选择。综上所述，我们认为本节的主要内容是使用 MCMC 进行 tip 选择。

4.2 分裂攻击

Aviv Zohar 针对所提出的 MCMC 算法提出了以下攻击方案。当 tangle 出于高负载的情况下，一个攻击者可以将 tangle 分裂成两个分支，并维持他们间的平衡。这种情况会导致两个分支都会继续发展。攻击者必须在拆分的开始处放置至少两个会引起冲突的交易，以防止一个可靠的节点同时引用这两个事务，从而有效地达到分裂目的。然后，攻击者希望网络的算力大约均分给每个分支，这样他们就能够通过自身补偿因随机引起的算力波动，即使自身算力相对较小。如果该技术有效，攻击者将能够在两个分支上花费相同的资金。

为了防御该类攻击，我们需要使用“锐阈值”规则，这规则使得很难在两个分支之间保持平衡。该规则的一个例子是选择比特币网络上最长的链。让我们将这种概念为 tangle 的所用。假设第一个分支的总权重为 537，第二个分支的总权重为 528。如果一个诚实的节点选择第一个分支的概率

非常接近 1/2，那么攻击者可能能够保持分支之间的平衡。然而，如果一个诚实的节点选择第一个分支的概率远远大于 1/2，那么攻击者可能无法保持平衡。在后一种情况下，由于网络在不可避免的随机波动之后，会快速地选择其中一个分支而放弃另一个分支，所以无法保持两个分支之间的平衡。为了使 MCMC 算法具有这样的行为，必须选择一个衰减非常快速的函数 f，并在一个深度较大的节点上开始随机游走，这样游动很可能在分支分叉之前就开始了。在这种情况下，即使竞争分支之间的累积权重差很小，walker 也会选择概率较高的 大权重的分支。值得注意的是，由于网络同步问题，攻击者的任务非常困难:他们可能不知道最近发布的大量交易(实际的累积权重与他们认为的会有很大的不同)。防范分裂攻击的另一种有效方法是让一个足够强大的节点在一个分支上立即发布大量交易，从而迅速改变平衡算力平衡，使攻击者难以处理这种变化。当然，如果攻击者设法维护分裂的分支，那么最近的交易大约只有 50%的确认信度，这样一来，交易边不会被确认，分支也不会增长。在此场景中，“诚实”节点可能决定开始有选择地选择认证发生在分支之前的交易，从而绕过了分叉分支上的交易。

可以考虑其他版本的 tip 选择算法。例如，例如，如果一个节点看到两个累积权重较大的子 tangle，那么在执行 MCMC tip 选择算法之前，它将选择权重总和较大的子 tangle。

对于进一步的实现，下面的想法可能值得考虑。我们可以使方程(13) 中定义的转移概率同时依赖于 hx-hy 和 hx，这样一来，当 walker 在 tangle 内部时，马尔可夫链的下一步移动几乎是确定的。而当 walker 靠近 tip 时，下一步的移动则变得更加随机。这将有助于避免进入较弱的分支，同时确保在选择 tip 时，具有足够的随机性。

总结

1.我们考虑了攻击者试图通过 超过 整个 WXCHAIN 网络的 双花攻击。
2.“大权重”攻击意味着，为了使双花交易成功，攻击者视图给双花交易一个非常大权重，以便它超过原交易的累积权重。换言之，如果允许交易自身的权重不受限制，这种策略将对网络构成威胁。所以，我们可以通过限制事务自身的权重或将其设置为常量作为解决方案。
3.在自身权重最大为 m 的情况下，最佳的攻击策略是生成自身权重为 m 的双花交易，然后新交易都直接选定并认证该双花交易。当诚实的交易分支笔攻击者的算力大是，攻击者可以使用公式(12) 来增大“双花”攻击被选定的概率。
4.构建“寄生链”的攻击方式使得 基于 height 或 score 的 tip 选择策略变得过时，因为与合法的 tangle 相比，攻击者所构建的链具有更高的度量值。另一方面，4.1 节所描述的 mcmc tip 选择算法似乎为这种攻击提供了保护机制。
5.mcmc 还额外的提供了正对懒惰节点的保护。

5.抵抗量子计算

众所周知，一个足够大的量子计算机(至今仍是一个假设性的构想)可以非常有效地处理 需要反复试验才能找到解决方案的问题。如比特币中，为了生成一个区块而找到一个 nonce 的过程就是一个很好的例子。对于比特币来说，到目前为止，必须平均检查 2^{68} 个 nonces，才能找到允许生成新块的合适散列。众所周知(见引用[15]),量子计算机需要 Θ(√N)(等价于 10√N)操作来解决一个类似于上述比特币难题的问题。而该问题对于经典计算机来说,需要 Θ(N)操作。因此,一台量子计算机挖掘比特币区块链的效率将是传统计算机的 $\sqrt{(2^{68})} = 234 \approx 170$ 亿倍左右。另外，值得注意的是，如果区块链不通过增加难度来响应 hash 算力的提高，那么孤立区块的速率将会增加。出于同样的原因，“大权重”攻击在量子计算机上也会更有效。然而，正如第 4 节所建议的那样去限制权重，也将有效地防止量子计算机攻击。这在 WXCHAIN 中很明显，因为为了找到合适的散列来发布交易，需要检查的 nonces 的数量并不大，平均的，大约为 3^8。因此，“理想”量子计算机的效率增益为 81，这已经是可以接受的了。更重要的是，WXCHAINHAIN 实现中使用的算法的结构是这样的:查找 nonce 的时间并不比发起交易所需的其他任务的时间长多少。后者对量子计算的抵抗力要大得多，因此与(比特币)区块链相比，它为 tangle 提供了更大的保护，使其免受量子计算机对手的攻击。

6.经济模型：

WXC 做为 WXChain 的主币，在币安链上映射一个 Token 代币

6.1 WXC

由于 WXChain 正处于开发阶段，暂无经济模式，敬请期待

6.2 WTOT

WTOT 是基于币安智能链（BINANCE SMART CHAIN）发布的 WXChain 的映射币
代币名称： WTOT
合约地址：0xf74db2c1a35a7f239Caa9D99B7941B329c7268BD
发行总量： 210,000
发行公链： 币安智能链 （BINANCE SMART CHAIN）
交易平台： PANCAKESWAP
分配机制:

1. 100%进入底池
2. 通过沃鑫贸易商城(https://www.woxintrade.com)购物获得
3. 商城分批次限量赠送

7.免责声明：

本白皮书仅作为项目阐述，任何行为均视为个人自愿承担一切后果之行为。除本白皮书所明确载明的之外，WXChain 不作任何陈述或保证。任何人参与 WTOT 交易的行为均基于其自己本身对 WXChain 的相关知识、 法律法规以及本白皮书的信息综合考虑，本文档不构成任何投资建议，在做出任何投资决定前，投资者应考虑产品的特点、其本身的投资目标、可承受的风险程度及其他因素，并适当的寻求独立的财务及专业意见。

WXChain 在此明确不予承认并拒绝承担下述责任：

•任何人在 WTOT 交易时违反了任何国家的反洗钱、反恐怖主义融资或其他监管要求；

•任何人在购买 WTOT 时违反了本白皮书规定的任何陈述、保证、义务、 承诺或其他要求，以及由此导致的无法使用或无法提取 WTOT；

•由于任何原因，WTOT 的交易计划被放弃；

•WXChain 的开发失败或被放弃，以及因此导致的无法交付或无法使用 WTOT；

•WXChain 开发的推迟或延期，以及因此导致的无法达成事先披露的日程；

•WXChain 源代码的错误、瑕疵、缺陷或其他问题；

•WXChain 平台的故障、崩溃、瘫痪、回滚或硬分叉；

•WXChain 未能实现任何特定功能或不适合任何特定用途；

•未能及时且完整的披露关于 WXChain 开发的信息。

引用

[1] bitcoinj. Working with micropayment channels.
https://bitcoinj.github.io/working-with-micropayments
[2] people on nxtforum.org (2014) DAG, a generalized blockchain.
https://nxtforum.org/proof-of-stake-algorithm/dag-a-generalized-blockchain/
(registration at nxtforum.org required)
[3] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, Aviv Zohar (2012) On Bitcoin and red balloons. Proc. 13th ACM Conf. Electronic Commerce, 56 - 73.
[4] Richard Durrett (2004) Probability - Theory and Examples. Duxbury advanced series.
[5] Sergio Demian Lerner (2015) DagCoin: a cryptocurrency without blocks.
https://bitslog.wordpress.com/2015/09/11/dagcoin/
[6] Yonatan Sompolinsky, Aviv Zohar (2013) Accelerating Bitcoin's Transaction Processing. Fast Money Grows on Trees, Not Chains.
https://eprint.iacr.org/2013/881.pdf
[7] Yonatan Sompolinsky, Yoad Lewenberg, Aviv Zohar (2016) SPECTRE: Serialization of Proof-of-work Events: Confirming Transactions via Recursive Elections.
https://eprint.iacr.org/2016/1159.pdf
[8] Yoad Lewenberg, Yonatan Sompolinsky, Aviv Zohar (2015) Inclusive Block Chain Protocols.
http://www.cs.huji.ac.il/~avivz/pubs/15/inclusive btc.pdf
[9] Joseph Poon, Thaddeus Dryja (2016) The Bitcoin Lightning Network:Scalable Off-Chain Instant Payments.
https://lightning.network/lightning-network-paper.pdf
[10] Sheldon M. Ross (2012) Introduction to Probability Models. 10th ed.
[11] David Vorick (2015) Getting rid of http://slides.com/davidvorick/braids
[12] Amir Dembo, Ofer Zeitouni (2010) Large Deviations Techniques and Applications. Springer.
[13] Sheldon M. Ross (2009) A First Course in Probability. 8th ed.
[14] Gilles Brassard, Peter Hyer, Alain Tapp (1998) Quantum cryptanalysis of hash and claw-free functions. Lecture Notes in Computer Science 1380, 163 - 169.